Complex & Intelligent Systems (2023) 9:4389–4401
https://doi.org/10.1007/s40747-022-00964-7

ORIGINAL ARTICLE

Multi-order hypergraph convolutional networks integrated with self-supervised learning

Jiahao Huang[1,2] · Fangyuan Lei[1,2] · Jianjian Jiang[1,2] · Xi Zeng[1,2] · Ruijun Ma[1,2,4] · Qingyun Dai[1,2,3]

Received: 23 June 2022 / Accepted: 26 December 2022 / Published online: 9 January 2023

Abstract

Hypergraphs, as a powerful representation of information, effectively and naturally depict complex and non-pair-wise relationships in the real world. Hypergraph representation learning is useful for exploring complex relationships implicit in hypergraphs. However, most methods focus on the 1-order neighborhoods and ignore the higher order neighborhood relationships among data on the hypergraph structure. These often result in underutilization of hypergraph structure. In this paper, we exploit the potential of higher order neighborhoods in hypergraphs for representation and propose a Multi-Order Hypergraph Convolutional Network Integrated with Self-supervised Learning. We first encode the multi-channel network of the hypergraph by a high-order spectral convolution operator that captures the multi-order representation of nodes. Then, we introduce an inter-order attention mechanism to preserve the low-order neighborhood information. Finally, to extract valid embedding in the higher order neighborhoods, we incorporate a self-supervised learning strategy based on maximizing mutual information in the multi-order hypergraph convolutional network. Experiments on several hypergraph datasets show that the proposed model is competitive with state-of-the-art baselines, and ablation studies show the effectiveness of higher order neighborhood development, the inter-order attention mechanism, and the self-supervised learning strategy.

Keywords Hypergraph · Multi-order · Hypergraph convolutional networks · Mutual information maximization · Contrastive learning

Introduction

Hypergraphs [7] provide a natural way to model complex patterns of component connectivity in the real world. In comparison to graphs, hypergraphs can connect non-pair-wise relations, a pattern that contains more information. With the development of deep network-based learning methods, hypergraphs have been widely applied in many domains, including pose estimation [16,29] and brain state classification [6,32].

Recently, researchers propose several hypergraph-based neural network frameworks [2,12,19,41,45,46]. Most of these methods focus on hypergraph expansion or on extending different network structures. For the Hypergraph Neural Network [12] (HGNN), message propagates by a hypergraph Laplacian operator on a clique expansion hypergraph, and follows a node-hyperedge-node propagation strategy. HyperGCN [41] approximates hyperedges as pair-wise edges, and thus, the hypergraph learning problem is converted to a graph learning problem. Moreover, unified framework [19,46] of hypergraphs and graphs is emerger as a trend in recent times.

✉ Fangyuan Lei
leify@126.com

Jiahao Huang
huangjaha@foxmail.com

Jianjian Jiang
371290422@qq.com

Xi Zeng
386832741@qq.com

Ruijun Ma
ruijunma@scau.edu.cn

Qingyun Dai
dyq@gpnu.edu.cn

[1] Guangdong Provincial Key Laboratory of Intellectual Property and Big Data, Guangdong Polytechnic Normal University, Guangzhou 510665, China

[2] School of Electronic and Information, Guangdong Polytechnic Normal University, Guangzhou 510665, China

[3] Pazhou Laboratory, Guangzhou 510320, China

[4] College of Engineering, South China Agricultural University, Guangzhou 510642, China

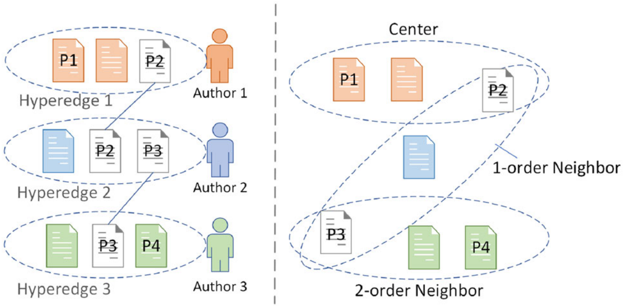


Fig. 1 Representation of higher order neighborhoods in hypergraphs. The papers of an author from a hyperedge. Author 2 co-authored paper P2 with author 1, meaning that hyperedge 2 constitutes a 1-order neighbor of hyperedge 1. Similarly, by the connection of paper P3, hyperedge 3 is a 2-order neighbor of hyperedge 1

Generally, these methods are designed by message passing process, making nodes and hyperedges confine to 1-order neighborhood in single propagation. However, nodes and hyperedges with the same attributes do not exist only in the 1-order neighborhood. For example, multiple authors co-authored a paper, which can be considered as a node, and multiple papers containing the same author are connected to a hyperedge. As shown in Fig. 1, Paper P2 in hyperedge 1 has two common author, and hyperedge 2 is a 1-order neighbor of hyperedge 1. Paper P3 in hyperedge 2 has two common author, and hyperedge 3 is a 2-order neighbor of hyperedge 2. Such a connection provides a way to reveal patterns of cross-domain collaboration. Therefore, hypergraphs serve as a powerful representation method for retaining information through deeper and more complex connectivity relationships. Furthermore, some works [1,31] on graph learning focus on neighborhood expansion of adjacency matrix. Method [20] uses powers of the incidence matrix to obtain higher order relationships, but it cannot be adapted to hypergraphs with arbitrary hyperedge sizes. Specifically, a larger receptive field means that nodes may receive more performance-degrading noises. Although the higher order neighborhood encapsulates a rich representation, it also brings more challenges, and it remains an open problem to effectively extract valuable information from the complex higher order neighborhood of objects while maintaining the lower order neighborhood information.

To address the above challenges, we propose Multi-Order Hypergraph Convolutional Networks Integrated with Self-Supervised Learning (MO-HGCN), where the multi-order representation maintains in the manner of a multi-channel network. We first perform k-th expansions of Chebyshev polynomials for spectral convolution to obtain spectral 2-order and spectral 3-order hypergraph convolution operators. Specifically, the operators are constructed as independent hypergraph convolution layers and modeled as a 2-order channel and a 3-order channel, respectively. In addition, we adaptly adjust the nodes' feature on the 1-order hypergraph convolution and utilize it as an enhanced information channel. Then, we propose an inter-order attention mechanism to learn the contrastive information among the different order neighborhoods. By assigning the attention scores to the node embedding of the higher order channel, the low-order neighborhood information is brought into focus. To extract valuable information of the higher order channels, we learn distinct representations in an self-supervised manner by incorporating maximizing mutual information-based contrastive learning. Finally, we fuse the node embedding learned from the 2-order and 3-order channels to represent the completed multi-order embeddings and optimize the weights of the network by joint learning. Compared with existing methods, MO-HGCN is a semi-supervised node classification model that combines self-supervised learning to obtain a multi-order neighborhood representation of nodes. The main contributions of this paper are as follows:

- We propose a Multi-Order Hypergraph Convolutional Network Integrated with Self-Supervised Learning (MO-HGCN) to explicitly capture the complex relationships of higher order neighborhood by spectral high-order hypergraph convolution operators, and obtain a multi-order representation through a multi-channel network.
- We propose an inter-order attention mechanism to maintain the information of low-order neighborhoods and learn the distinct representation of higher order neighborhoods by a mutual information maximization strategy in a self-supervised learning manner.
- We conduct extensive experiments on several hypergraph datasets, and the results show the effectiveness of MO-HGCN compared with the state-of-the-art.

Related works

Hypergraph neural networks

In recent years, hypergraphs have gained attention among researchers, and representation learning methods based on hypergraph have been greatly developed. Feng et al. [12] propose the Hypergraph Neural Network (HGNN), a general framework which implements the message passing strategy on the hypergraph with a hyperedge convolutional layer. To avoid the limitations of inherent hypergraph structure, Jiang et al. [22] propose a dynamic hypergraph neural network that updates the hypergraph structure. Bai et al. [4] propose two trainable operators, namely Hypergraph Convolution and Hypergraph Attention, that can be extended and migrated in neural networks. Besides, some studies propose new hypergraph representation learning frameworks, such as HNHN

[10], Hyper-SAGNN [45], HyperSAGE [2], and HGC-RNN [43].

In the exploration of hypergraph structure, HyperGCN [41] makes hypergraphs to be trained on graph convolutional networks by approximating hyperedges as pair-wise edges. Bandyopadhyay et al. [5] apply graph convolution on the line graph of the hypergraph to adapt variable-sized hyperedges. Yang et al. [42] treat the vertices and hyperedges equally to solve the symmetric information loss problem of data co-occurrence. Various types of practices [13] based on hypergraphs are also evolving, such as pose estimation [16,29], link prediction [11], recommendation [23,38,39,44], and brain state classification [6,32].

A recent trend combining hypergraphs with graph network methods has emerged as a result of the advantages of data modeling brought by non-pair-wise relations in hypergraphs. Huang et al. [19] propose a framework for modeling the message passing process in graph and hypergraph neural networks. Zhang et al. [46] consider a hypergraph with edge-related vertex weight, propose the generic hypergraph spectral convolution networks (GHSC), and present various variants of hypergraph neural networks.

Self-supervised learning

Self-supervised learning [21,24,30] is currently receiving considerable attention in deep learning, serving downstream tasks by learning useful information in unlabeled data. Self-supervised learning has a wide range of applications in computer vision [3,18], natural language processing [28], and graph learning [17,25,34,37].

One popular approach on graph learning is mutual information maximization, i.e., global–local contrast. Hjelm et al. [18] introduce the application of mutual information maximization strategies on images by proposing Deep InfoMax (DMI). DMI is adapted to different downstream tasks by global–local contrast, e.g., local features are suitable for classification tasks. Veličković et al. [37] extend this paradigm to graph learning and propose the Deep Graph Infomax (DGI). DGI performs global and local neighborhood comparisons on graphs, enabling nodes to learn global and local structural information. InfoGraph [34] maximizes the mutual information between graph-level representations and substructured representations at different scales to learn the global graph representation. Rich representations are also learned from labeled and unlabeled datasets by semi-supervised learning.

The mutual information maximization strategy is extended to certain tasks on hypergraphs. Xia et al. [39] propose a dual channel hypergraph convolutional network, which employ self-supervised learning as an auxiliary task to enhance the performance of session recommendation. Yu et al. [44] use the higher order relations of hypergraphs to obtain complex relationships between users and compensates for the information loss due to multi-channel networks with multi-layer mutual information maximization. These works investigate the impact of mutual information maximization for different types of information, while our work explores the implications of mutual information maximization strategy in higher order neighborhood.

Method

In this section, we describe in detail the proposed Multi-Order Hypergraph Convolutional Network Integrated with Self-Supervised Learning (MO-HGCN). As shown in Fig. 2, MO-HGCN consists of a 2-order channel, a 3-order channel, and an enhanced information channel, with 2-order and 3-order channels as the outputs. Specifically, we design the spectral 2-order and 3-order hypergraph convolution operators to obtain the higher order information. Considering the importance of node features and the preservation of 1-order neighborhoods, we propose an inter-order attention mechanism in the multi-order hypergraph convolution network. Our goal is to fuse the multi-order information to obtain a multi-level representation of the nodes. We further introduce self-supervised learning on the hypergraph, i.e., mutual information maximization between different order to capture the distinctive higher order information of the nodes.

Preliminaries

Given a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, $\mathcal{V}$ is a vertex set $\{v_1, v_2, \ldots, v_n\}$ with n nodes, and $\mathcal{E}$ is a hyperedge set $\{e_1, e_2, \ldots, e_m\}$ with m hyperedges. The hyperedge weight $\mathbf{W}$ represents a diagonal matrix which the hyperedge weight set $\{W_1, W_2, \ldots, W_m\}$ is the diagonal. Thus, hypergraph $\mathcal{G}$ can be represented as an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, and the entries of $\mathbf{H}$ denote as

$$h(v, e) = \begin{cases} 1, & \text{if } v \in e, \\ 0, & \text{if } v \notin e. \end{cases} \tag{1}$$

The Laplacian [47] $\mathbf{\Delta}$ of a hypergraph $\mathcal{G}$ denotes as

$$\mathbf{\Delta} = \mathbf{I} - \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}, \tag{2}$$

where the $\mathbf{D}_v$ is a diagonal matrix of vertex degree and the $\mathbf{D}_e$ is a diagonal matrix of hyperedge degree.

Alternatively, given a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W}, \mathbf{\Delta})$ with n nodes and m hyperedges, the hypergraph Laplacian $\mathbf{\Delta} \in \mathbb{R}^{n\times n}$ can be decomposed into an orthonormal eigen vectors Φ and a non-negative eigenvalue diagonal matrix Λ. For a hypergraph $\mathcal{G}$ and a signal $\boldsymbol{x}$, the spectral convolution of a

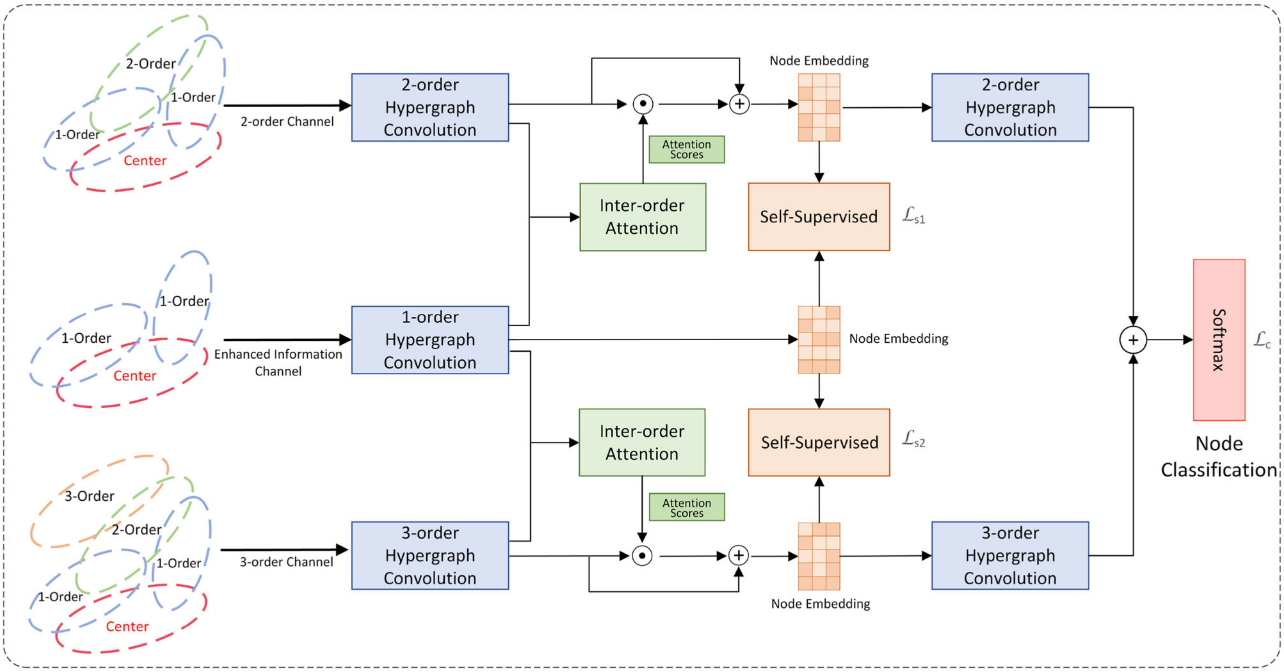


Fig. 2 The framework of MO-HGCN

signal $\boldsymbol{x}$ on a filter $\boldsymbol{g}$ represents as

$$\boldsymbol{g}\star\boldsymbol{x} = \Phi\boldsymbol{g}(\Lambda)\Phi^T\boldsymbol{x}, \tag{3}$$

where the symbol $\star$ represents the convolution operator, and $\boldsymbol{g}(\Lambda)$ indicates the Fourier coefficients. The function $\boldsymbol{g}(\Lambda)$ is further parameterized as K order polynomials which express as the truncated Chebyshev expansion. The Chebyshev polynomial is expanded as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, and $T_0(x) = 1$, $T_1(x) = x$. With the truncated Chebyshev expansion, the spectral convolution approximatively represents as

$$\boldsymbol{g}\star\boldsymbol{x} \approx \sum_{k=0}^{K}\theta_k T_k(\tilde{\boldsymbol{\Delta}})\boldsymbol{x}, \tag{4}$$

where $T_k(\tilde{\boldsymbol{\Delta}})$ indicates the K order Chebyshev polynomial, and $\tilde{\boldsymbol{\Delta}} = \frac{2}{\lambda_{\max}}\boldsymbol{\Delta} - \mathbf{I}$ is a scaled Laplacian. Moreover, $\lambda_{\max} \approx 2$ according to works [12,26]. Therefore, the spectral 1-order hypergraph convolution can be defined as

$$\mathbf{X}^{(l+1)} = \mathbf{D}_v^{-1/2}\mathbf{HWD}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{X}^{(l)}\theta^{(l)}. \tag{5}$$

Multi-order hypergraph convolutional network

On the basis of the spectral convolution theory of hypergraphs, we further develop an operator that facilitates nodes and hyperedges to interact in higher order neighborhoods.

For higher order hypergraph convolution, we extend the order of the neighborhoods from the perspective of spectral hypergraph convolution. According to the Chebyshev polynomial in Eq. (4), k is set to 2 to obtain the spectral 2-order hypergraph convolution operator as follows:

$$\begin{aligned}\boldsymbol{g}\star\boldsymbol{x} &\approx \sum_{k=0}^{2}\theta_k T_k(\tilde{\boldsymbol{\Delta}})\boldsymbol{x}\\ &= \theta_0 T_0(\tilde{\boldsymbol{\Delta}})\mathbf{x} + \theta_1 T_1(\tilde{\boldsymbol{\Delta}})\mathbf{x} + \theta_2 T_2(\tilde{\boldsymbol{\Delta}})\mathbf{x}\\ &= \theta_0\mathbf{x} - \theta_1^{(2)}\boldsymbol{\Theta}\mathbf{x} + 2\theta_2\boldsymbol{\Theta}^2\mathbf{x} - \theta_2\mathbf{Ix},\end{aligned} \tag{6}$$

where θ_0, θ_1, and θ_2 denote the parameters of filters $\boldsymbol{g}$, and $\boldsymbol{\Theta} = \mathbf{D}_v^{-1/2}\mathbf{HWD}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}$. Following the way of works [12,26] to avoid over-parameterization, we reduce multiple parameters to a single parameter which is assumed as

$$\begin{cases}\theta_2 = -\frac{1}{2}\theta,\\ \theta_1 = -\boldsymbol{\Theta}\theta,\\ \theta_0 = \boldsymbol{\Theta}^2\theta.\end{cases} \tag{7}$$

Thus, the spectral 2-order hypergraph convolution operator can be simplified as follows:

$$\boldsymbol{g}\star\boldsymbol{x} \approx \theta\left(\boldsymbol{\Theta}^2 + \frac{1}{2}\mathbf{I}\right)\mathbf{x}. \tag{8}$$

With the spectral 2-order hypergraph convolution operator, the 2-order hypergraph convolution of signal $\mathbf{X}$ can be defined as

$$\mathbf{X}^{(l+1)} = \theta^{(l)}\left(\boldsymbol{\Theta}^2 + \frac{1}{2}\mathbf{I}\right)\mathbf{X}^{(l)}, \tag{9}$$

where $\theta \in \mathbb{R}^{C_{\text{in}} \times C_{\text{out}}}$ represents the learnable parameter.

Similarly, the spectral 3-order hypergraph convolution operator represents as

$$\begin{aligned} \boldsymbol{g} \star \boldsymbol{x} &\approx \sum_{k=0}^{3} \hat{\theta}_k T_k(\tilde{\boldsymbol{\Delta}})\boldsymbol{x} \\ &= \hat{\theta}_0 T_0(\tilde{\boldsymbol{\Delta}})\mathbf{x} + \hat{\theta}_1 T_1(\tilde{\boldsymbol{\Delta}})\mathbf{x} + \hat{\theta}_2 T_2(\tilde{\boldsymbol{\Delta}})\mathbf{x} + \hat{\theta}_3 T_3(\tilde{\boldsymbol{\Delta}})\mathbf{x} \\ &= \hat{\theta}_0 \mathbf{x} - \hat{\theta}_1 \boldsymbol{\Theta}\mathbf{x} + 2\hat{\theta}_2 \boldsymbol{\Theta}^2\mathbf{x} - \hat{\theta}_2 \mathbf{I}\mathbf{x} - 4\hat{\theta}_3 \boldsymbol{\Theta}^3\mathbf{x} + 3\hat{\theta}_3 \boldsymbol{\Theta}\mathbf{x}, \end{aligned} \tag{10}$$

where $\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2$, and $\hat{\theta}_3$ denote the parameters of filters $\boldsymbol{g}$. We also uses a single parameter $\hat{\theta}$ to avoid over-parameterization as follows:

$$\begin{cases} \hat{\theta}_3 = \frac{1}{2}\hat{\theta}, \\ \hat{\theta}_2 = \frac{1}{2}\boldsymbol{\Theta}\hat{\theta}, \\ \hat{\theta}_1 = \frac{1}{2}\boldsymbol{\Theta}^2\hat{\theta}, \\ \hat{\theta}_0 = -\frac{1}{2}\boldsymbol{\Theta}^3\hat{\theta}. \end{cases} \tag{11}$$

With the single parameter, the spectral 3-order hypergraph convolution operator can be simplified as

$$\begin{aligned} \boldsymbol{g} \star \boldsymbol{x} &\approx -2\hat{\theta}\left(\boldsymbol{\Theta}^3 - \frac{1}{2}\boldsymbol{\Theta}\right)\mathbf{x} \\ &= \theta\left(\boldsymbol{\Theta}^3 - \frac{1}{2}\boldsymbol{\Theta}\right)\mathbf{x}. \end{aligned} \tag{12}$$

Thus, the 3-order hypergraph convolution of signal $\mathbf{X}$ can be defined as

$$\mathbf{X}^{(l+1)} = \theta^{(l)}\left(\boldsymbol{\Theta}^3 - \frac{1}{2}\boldsymbol{\Theta}\right)\mathbf{X}^{(l)}, \tag{13}$$

where $\theta \in \mathbb{R}^{C_{\text{in}} \times C_{\text{out}}}$ represents the learnable parameter.

In MO-HGCN, l in Eqs. (9) and (13) is set to 2. Therefore, the backbone network of MO-HGCN is a multi-channel network with two layers. This approach preserves the information of different order neighborhoods and facilitates the nodes to learn multi-order representations.

Inter-order attention

The spectral high-order hypergraph convolution operator allows each node to aggregate information from distal nodes and hyperedges. Such information may not be applicable to learn directly, which presents a challenge in regulating the involvement of low-order information. Therefore, we propose an inter-order attention mechanism to indicate the similarity between higher order neighborhoods and low-order neighborhoods. Unlike previous attentions, we focus on the comparison of attention between different orders of the same node, rather than among neighboring nodes. Particularly, we design an enhanced information channel based on 1-order hypergraph convolution that augments the nodes' own information. The convolution process of this channel is as follows:

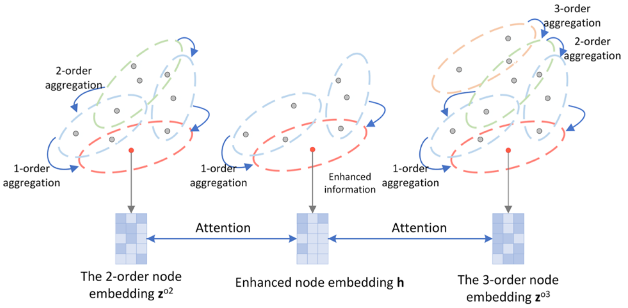


Fig. 3 The inter-order attention mechanism of MO-HGCN

$$\boldsymbol{h} = \theta \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{x} + \theta\beta\mathbf{I}\mathbf{x}, \tag{14}$$

where $\boldsymbol{h}$ represents the enhanced node embedding, and the $\beta \in \mathbb{R}^N$ denotes a learnable parameter that assigns different self-loop weights to each node.

As shown in Fig. 3, we obtain the node embeddings z^{o2} and z^{o3} of the 2-order and 3-order channels after the first layer of convolution. Then, the attention mechanism is applied as between z^{o2} and $\boldsymbol{h}$, and between z^{o3} and $\boldsymbol{h}$, respectively.

The attention scores of the nodes embeddings $z_i^l \in \mathbb{R}^K$ and $z_i^h \in \mathbb{R}^K$ in the j-th dimensional feature for the low-order channel and the higher order channel are calculated by Eq. 15

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(MLP(z_{ij}^l \| z_{ij}^h)\right)\right)}{\sum_{k \in i}\exp\left(\text{LeakyReLU}\left(MLP(z_{ik}^l \| z_{ik}^h)\right)\right)}, \tag{15}$$

where z_{ij}^l and z_{ij}^h represent the j-th dimensional feature of node i, $MLP(\bullet) : \mathbb{R}^{2 \times K} \to \mathbb{R}^K$ is the feature mapping function which set as a multi-layer perceptron, and $\|$ denotes the concatenation operation. Therefore, the attention score α_{o2} between 2-order channel and enhanced information channel calculated as

$$\alpha^{o2} = \frac{\exp\left(\text{LeakyReLU}\left(MLP(\boldsymbol{h}_{ij} \| z_{ij}^{o2})\right)\right)}{\sum_{k \in i}\exp\left(\text{LeakyReLU}\left(MLP(\boldsymbol{h}_{ik} \| z_{ik}^{o2})\right)\right)}. \tag{16}$$

The attention score α_{o3} between 3-order channel and enhanced information channel calculated as

$$\alpha^{o3} = \frac{\exp\left(\text{LeakyReLU}\left(MLP(\boldsymbol{h}_{ij}\|z^{o3}_{ij})\right)\right)}{\sum_{k\in i}\exp\left(\text{LeakyReLU}\left(MLP(\boldsymbol{h}_{ik}\|z^{o3}_{ik})\right)\right)}. \tag{17}$$

The attention scores α_{o2} and α_{o3} are further assigned to the higher order node embedding z^{o2} and z^{o3} as a way to enhance the most relevant node representation between channels, and the processes are represented as

$$\hat{z}^{o2} = \alpha^{o2} z^{o2}, \tag{18}$$
$$\hat{z}^{o3} = \alpha^{o3} z^{o3}. \tag{19}$$

To preserve the original higher order message of node embedding, we fuse node embeddings $\hat{z}^{o2}$ and $\hat{z}^{o3}$ with node embeddings z^{o2} and z^{o3}, respectively. The final obtained higher order channel node embedding $\hat{z}^{o2}$ and $\hat{z}^{o3}$ are as follows:

$$\tilde{z}^{o2} = \lambda z^{o2} + (1-\lambda)\hat{z}^{o2}, \tag{20}$$
$$\tilde{z}^{o3} = \mu z^{o3} + (1-\mu)\hat{z}^{o3}, \tag{21}$$

where the $\lambda \in \mathbb{R}^1$ and $\mu \in \mathbb{R}^1$ denote the learnable parameter restricted to [0,1], controlling the involvement of two node embeddings.

Algorithm 1 Algorithm of MO-HGCN.

```
Input:
Incidence matrix $\mathbf{H}$, feature matrix $\mathbf{X}$.
Output:
Predicted sample $\tilde{Y}$.
Initialize:
Compute spectral 2-order hypergraph convolution operator via Eq. (8);
Compute spectral 3-order hypergraph convolution operator via Eq. (12);
while not converged do
    Learn the node embeddings $\boldsymbol{h}$, $z^{o2}$, and $z^{o3}$ of first layer via Eqs. (14), (9), and (13) respectively;
    Calculate the inter-order attention scores $\alpha_{o2}$ and $\alpha_{o3}$ via Eqs. (16) and (17), respectively;
    Embed the enhanced node embeddings $\hat{z}^{o2}$ and $\hat{z}^{o3}$ via Eqs. (18) and (19), respectively;
    Establish the contrastive learning of $\boldsymbol{h}$, $\tilde{z}^{o2}$, and $\tilde{z}^{o3}$ via Eqs. (20) and (21);
    Learn the node embeddings $\boldsymbol{X}^{(2)}$ and $\boldsymbol{X}^{(3)}$ of second layer via Eqs. (9) and (13), respectively;
    Fuse the node embeddings $\boldsymbol{X}^{(2)}$ and $\boldsymbol{X}^{(3)}$ into $\hat{\boldsymbol{X}}$ via Eq. (24);
    Minimize $\mathcal{L}$ with gradient descent optimization algorithm via Eq. (26);
end while
```

Self-supervised learning auxiliary task

Multi-order hypergraph convolutional networks enable nodes to learn multiple levels of representations, further improving model performance. However, the multi-channel structure is independent of each other, and the higher order information usually contains varying degrees of redundant information. Therefore, it is worth considering how to extract the distinctive message from the multi-order hypergraph convolutional network. Inspired by mutual information maximization could improve the Deep Graph Infomax (DGI) [37] performance. We extend the mutual information maximization to the inter-order to guide the model to reduce feature redundancy. Specifically, we construct a contrastive learning between the enhanced information channel and the higher order channel, respectively. For the 2-order channel, the positive sample pair is $\left(\boldsymbol{h}_{ij}, \tilde{z}^{o2}_{ij}\right)$ and the negative sample pair is $\left(\hat{\boldsymbol{h}}_{ij}, \tilde{z}^{o2}_{ij}\right)$, where $\hat{\boldsymbol{h}}_{ij}$ denotes the negative sample with row-wise shuffling. We utilize InfoNCE [18] as the loss function for contrastive learning, as follows:

$$\mathcal{L}_{s1} = -\sum_{i\in\mathcal{V}}\left(\sum_{j\in k}\log\sigma\left(\mathcal{S}\left(\boldsymbol{h}_{ij}, \tilde{z}^{o2}_{ij}\right)\right) + \sum_{j\in k}\log\sigma\left(1-\mathcal{S}\left(\hat{\boldsymbol{h}}_{ij}, \tilde{z}^{o2}_{ij}\right)\right)\right), \tag{22}$$

where $\mathcal{S}(\bullet \odot \bullet)$ denotes the discriminator function as the dot product.

For the 3-order channel, the positive sample pair is $\left(\boldsymbol{h}_{ij}, \tilde{z}^{o3}_{ij}\right)$ and the negative sample pair is $\left(\hat{\boldsymbol{h}}_{ij}, \tilde{z}^{o3}_{ij}\right)$, where $\hat{\boldsymbol{h}}_{ij}$ denotes the negative sample with row-wise shuffling. The InfoNCE loss function $\mathcal{L}_{s2}$ is defined as

$$\mathcal{L}_{s2} = -\sum_{i\in\mathcal{V}}\left(\sum_{j\in k}\log\sigma\left(\mathcal{S}\left(\boldsymbol{h}_{ij}, \tilde{z}^{o3}_{ij}\right)\right) + \sum_{j\in k}\log\sigma\left(1-\mathcal{S}\left(\hat{\boldsymbol{h}}_{ij}, \tilde{z}^{o3}_{ij}\right)\right)\right). \tag{23}$$

Model learning

The node embedding $\tilde{z}^{o2}$ and $\tilde{z}^{o3}$ input to the second layer of multi-order hypergraph neural network. The outputs of second layer are denoted as $\boldsymbol{X}^{(2)} \in \mathbb{R}^{N\times q}$, and $\boldsymbol{X}^{(3)} \in \mathbb{R}^{N\times q}$, where q is the number of classes. To conduct the node clas-

Table 1 Summary of the hypergraph datasets used

Dataset	Co-authorship data		Co-citation data		
	DBLP	Cora	PubMed	Citeseer	Cora
#Number of nodes	43,413	2708	19,717	3312	2708
#Number of hyperedges	22,535	1072	7963	1079	1579
#Average hyperedge size	4.7 ±6.1	4.2 ± 4.1	4.3 ± 5.7	3.2 ± 2.0	3.0 ± 1.1
#Number of features	1452	1433	500	3703	1433
#Number of classes	6	7	3	6	7
#Ratio of training set	0.040	0.052	0.008	0.042	0.052

sification, we adopt the summation strategy to achieve the fusion of multi-channel information by Eq. (24)

$$\hat{\boldsymbol{X}} = \boldsymbol{X}^{(2)} + \boldsymbol{X}^{(3)}. \tag{24}$$

Then, we adopt the softmax function to predict the label $\hat{Y}$ by $\hat{\boldsymbol{X}}$. Thus, the cross-entropy loss function for node classification is defined as follows:

$$\mathcal{L}_c = -\sum_{i \in \mathcal{V}_L} \sum_{j=1}^{q} Y_{ij} \ln \hat{Y}_{ij}, \tag{25}$$

where the Y_{ij} denotes the true labels of $\mathcal{V}_L$.

Therefore, the joint learning loss function is as follows:

$$\mathcal{L} = \mathcal{L}_c + \eta_1 \mathcal{L}_{s1} + \eta_2 \mathcal{L}_{s2}, \tag{26}$$

where η_1 and η_2 are hyperparameters that control the participation of self-supervised learning. Algorithm1 reports the overall process of MO-HGCN.

Experiments

In this section, we conduct experiments and validate our model by answering the following questions.

- **Q1**: How does MO-HGCN perform on the node classification task?
- **Q2**: How does high-order spectral hypergraph convolution perform compared to 1-order spectral hypergraph convolution?
- **Q3**: How does the inter-order attention mechanism contribute to the performance of MO-HGCN?
- **Q4**: How sensitive is the performance of MO-HGCN to its parameter settings?
- **Q5**: How does the self-supervised learning component affect the effectiveness of MO-HGCN?

Datasets

For the semi-supervised node classification task on hypergraphs, we use the five hypergraph datasets provided by HyperGCN [41] for validation. These datasets include the co-citation network and the co-authorship network. Summary of the datasets is shown in Table 1 and details are as follows:

- **Co-citation datasets**: The original sources of the co-citation network hypergraph dataset are cora, citeseer, and PubMed. In the hypergraph construction, all documents are created as nodes, and documents cited by the same document are grouped as a hyperedge. Hyperedges containing only one node are removed and the node feature is the bag-of-words vector of documents.
- **Co-authorship datasets**: The original sources of the co-authorship network hypergraph dataset are DBLP and cora. In the hypergraph construction, all papers are considered as nodes and papers authored by a author are grouped as a hyperedge. The nodes are characterized by the bag-of-words vector of papers.

Baselines

We compare the proposed method with state-of-the-art baselines that include a variety of hypergraph neural networks combined with different neural network models. Details of these approach are as follows:

- **MLP+HLR** [2]: The method is a multi-layer perceptron using explicit hypergraph Laplacian for regularization.
- **HyperGCN** [2]: HyperGCN approximates the hypergraph learning problem to a graph problem by pair-wise edges and provides a variant FastHyperGCN that reduces the training time.
- **HyperSAGE** [2]: HyperSAGE utilizes a two-level neural messaging strategy to propagate information in the hypergraph and combines different neighborhood aggregation approaches of GraphSAGE [15].
- **HGNN** [12]: HGNN introduces the symmetric normalized hypergraph Laplacian [47] operator by means of

spectral hypergraph theory and provides a general framework for hypergraph neural networks.

- **UniGCN** [19]: UniGNN unifies the message passing process of graphs and hypergraphs into a framework, extending the Graph Neural Networks design naturally to hypergraphs.
- **UniGAT** [19]: The method extends the aggregation process of Graph Attention Networks [36] to hypergraphs, so that nodes learn the attention weights of neighboring hyperedges.
- **UniGIN** [19]: The method uses the mechanism of Graph Isomorphic Networks [40] to enhance the expressiveness by aggregating the information of neighboring hyperedges by nodes.
- **UniSAGE** [19]: The method is a variant of GraphSAGE [15], which adapts to different tasks by means of different aggregation functions.
- **H-ChebNet** [46]: Combined with ChebNet [9], a variant derived on the general hypergraph spectral convolution framework.
- **H-APPNP** [46]: H-APPNP is a hypergraph convolutional network with APPNP [27] as the backbone network.
- **H-SSGC** [46]: The method extends the SSGC [48] to a general hypergraph spectral convolution framework.
- **H-GCN** [46]: H-GCN is a general hypergraph spectral convolution framework with Graph Convolutional Networks [26] as the backbone network.
- **H-GCNII** [46]: H-GCNII extends the GCNII [8] to a general hypergraph spectral convolution framework, which is a deep network structures.

Experiments settings

For the semi-supervised node classification task, we use ACC (Accuracy) to evaluate the performance of the model. In the experimental setup, we utilize the Adma algorithm for training and set it as 2000 epochs. For the Cora (including Co-citation and Co-authorship) and Citeseer datasets, the learning rate is 0.005 and L2 regularization is 0.05. For the DBLP and Pubmed datasets, the learning rate is 0.05 and L2 regularization is 0.002. For the hyperparameters η_1 and η_2, the Cora (including Co-citation and Co-authorship) are set to 0.005 and 0.005, respectively, while other datasets used are set to 0.001. Each dataset has ten different split training-test sets with consistent training-test ratios. We follow the way of work [19] to test the datasets. For the baselines, we cite the experimental results reported in the original paper, since the compared datasets and evaluation metric are consistent.

Experimental results

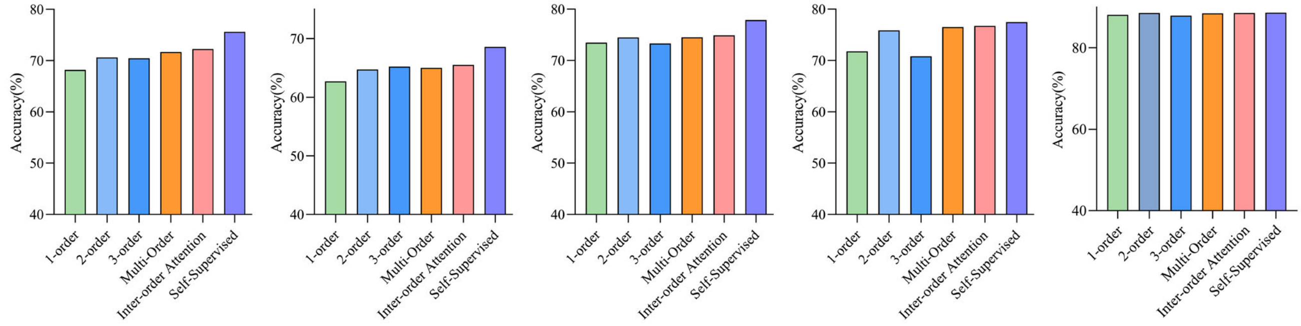
Performance analysis

We report the mean accuracy and standard deviation of the experimental results in Table 2, with the best results in bold and the second best results underlined. The experimental results show the advantage of our model in terms of its accuracy compared with the state-of-the-art, with improvements of 2.0%, 4.0%, 3.1%, and 1.0% on the co-citation cora, citeseer, PubMed, and co-authorship cora datasets, respectively. The best-performing method on the co-authorship DBLP dataset is H-GCNII. The experimental results in Table 2

Table 2 Test accuracy (%) of node classification on hypergraph datasets

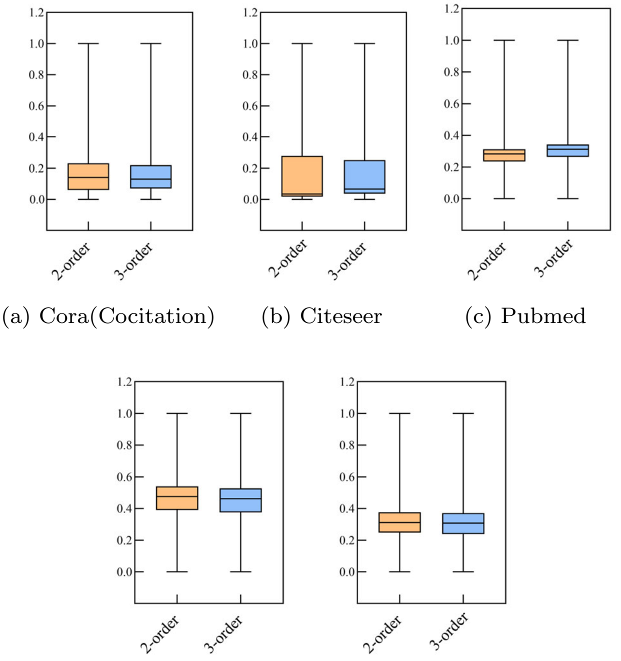
Dataset	Co-authorship data		Co-citation data		
	DBLP	Cora	PubMed	Citeseer	Cora
MLP + HLR	63.6 ± 4.7	59.8 ± 4.7	64.7 ± 3.1	56.1 ± 2.6	61.0 ± 4.1
FastHyperGCN	68.1 ± 9.6	61.1 ± 8.2	65.7 ± 11.1	56.2 ± 8.1	61.3 ± 10.3
HyperGCN	70.9 ± 8.3	63.9 ± 7.3	68.3 ± 9.5	57.3 ± 7.3	62.5 ± 9.7
HyperSAGE	77.4 ± 3.8	72.4 ± 1.6	72.9 ± 1.3	61.8 ± 2.3	69.3 ± 2.7
UniGAT	88.7 ± 0.2	75.0 ± 1.1	74.7 ± 1.2	63.8 ± 1.6	69.2 ± 2.9
UniGCN	88.8 ± 0.2	75.3 ± 1.2	74.4 ± 1.0	63.6 ± 1.3	70.1 ± 1.4
UniGIN	88.6 ± 0.3	74.8 ± 1.3	74.4 ± 1.1	63.3 ± 1.2	69.2 ± 1.5
UniSAGE	88.5 ± 0.2	75.1 ± 1.2	74.3 ± 1.0	63.8 ± 1.3	70.2 ± 1.5
H-ChebNet	87.9 ± 0.24	70.6 ± 2.1	74.3 ± 1.5	63.5 ± 1.3	69.7 ± 2.0
H-APPNP	89.4 ± 0.18	76.4 ± 0.8	75.3 ± 1.1	64.5 ± 1.4	70.9 ± 0.7
H-SSGC	88.6 ± 0.16	72.0 ± 1.2	74.5 ± 1.3	60.5 ± 1.7	68.8 ± 2.1
H-GCN	89.0 ± 0.19	74.8 ± 0.9	75.4 ± 1.2	62.7 ± 1.2	69.5 ± 2.0
H-GCNII	**89.8 ± 0.20**	76.2 ± 1.0	75.8 ± 1.2	64.5 ± 1.0	72.5 ± 1.2
HGNN	88.09 ± 0.15	71.77 ± 1.58	73.47 ± 1.30	62.71 ± 1.16	68.17 ± 1.12
MO-HGCN	88.65 ± 0.19	**77.47 ± 0.66**	**77.88 ± 1.18**	**68.56 ± 1.30**	**75.60 ± 0.74**

The best results are marked in bold, while the second best results are underlined



(a) Cora(Cocitation) (b) Citeseer (c) Pubmed (d) Cora(Coauthor) (e) DBLP

Fig. 4 Performance of different channels and components in MO-HGCN. The vertical axis denotes the accuracy and the horizontal axis represents the different channels and components. For the horizontal axis, the 1-order represents the 1-order approximation of the hypergraph convolution, meaning HGNN [12]. The 2-order and 3-order represent the 2-order and 3-order channels in MO-HGCN, respectively. The Multi-order denotes the MO-HGCN only consisting of 2-order and 3-order channels. The inter-order attention represents the MO-HGCN involving the inter-order attention component, and the Self-Supervised denotes the MO-HGCN with all channels and components



(a) Cora(Cocitation) (b) Citeseer (c) Pubmed

(d) Cora(Coauthor) (e) DBLP

Fig. 5 Distribution of inter-order attention scores

answer the question **Q1**: The proposed model outperforms the state-of-the-art baselines.

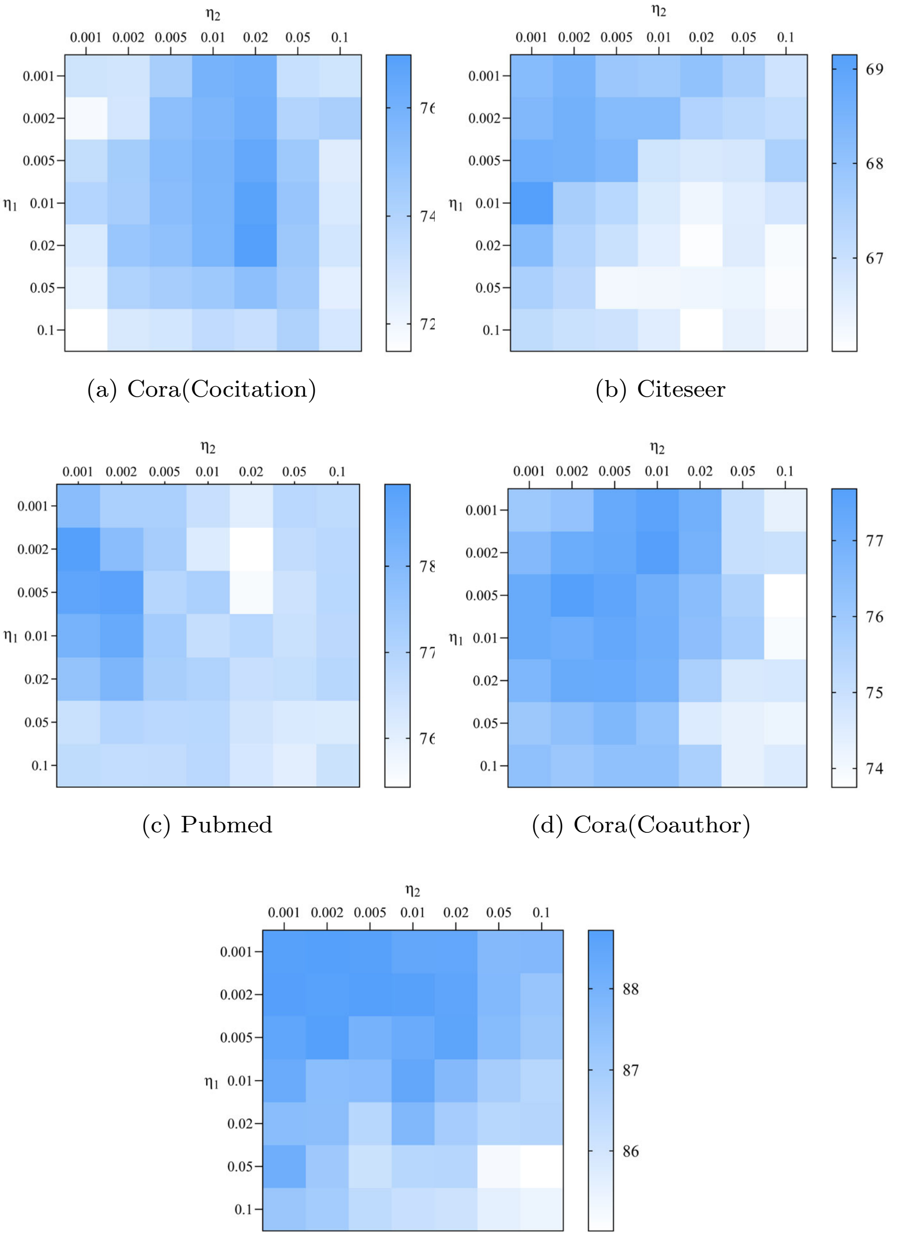
Comparatively to the pair-wise edge approximation of HyperGCN, MO-HGCN utilizes clique expansion to approximate the hypergraph structure as HGNN, and the multi-order approximation neighborhood further enlarges the receptive field of nodes and hyperedges. It is for this reason that MO-HGCN performs better than HyperGCN and HGNN. Compared to models combining the hypergraph with other GNN methods, although these models absorb the advantages of different GNN structures and perform well, the results show that multi-order hypergraph convolutional networks combining inter-order attention mechanisms and self-supervised learning can fully exploit the structural information of the hypergraph. The standard deviation results on multiple split sets also demonstrate that MO-HGCN achieves similar stability as the model that incorporates hypergraph and GNN methods.

Ablation study

We report in Fig. 4 the performance of different channels and different components of MO-HGCN as a way to investigate the contribution. As shown in Fig. 4, the **1-order** of horizontal axis is a 1-order approximation of the hypergraph convolution, which is compared with the higher order channels. The **2-order** and **3-order** denote hypergraph convolutional networks using only the 2-order channel and the 3-order channel, respectively. The **Multi-order** represents the MO-HGCN only consisting of 2-order and 3-order channels. The **Inter-order attention** denotes a multi-order hypergraph convolutional network that involves inter-order attention. The **Self-supervised** represents a multi-order hypergraph network consisting of inter-order attention and self-supervised learning.

As can be observed from Fig. 4, the 2-order channel always performs better than the 3-order channel, while the 2-order channel outperforms the 1-order channel in most cases but is inferior to the 2-order channel. This also indicates that the long-range information in higher order neighborhoods is not always directly applicable. The results of the **Multi-order** show that the fusion of multi-order information allows models to learn multiple levels of representation,

Fig. 6 Node classification results for MO-HGCN with different assignments on the η_1 and η_2 . The horizontal axis represents η_2 . The vertical axis represents η_1



(a) Cora(Cocitation)

(b) Citeseer

(c) Pubmed

(d) Cora(Coauthor)

(e) DBLP

thus further improving the performance. The self-supervised learning component delivers a significant boost compared to inter-order attention, suggesting a more prominent role for extracting the distinctive information of higher order information. The results in Fig. 4 answer question **Q2**: Channels and components in the model contribute differently, with inter-order attention and self-supervised learning taking full advantage of the natural information brought by multi-order neighborhoods.

Effectiveness of inter-order attention

We use box plots in Fig. 5 to report the distribution of attention scores in the inter-order attention mechanism as a way to investigate the question **Q3**. The 2-order and 3-order in Fig. 5 represent the attention scores distributions between the 2-order channel and the enhanced information channel, and between the 3-order channel and the enhanced information channel, respectively.

As shown in Fig. 5, the distribution of attention scores generated by the inter-order attention mechanism is concentrated in the lower score region, which indicates a large discrep-

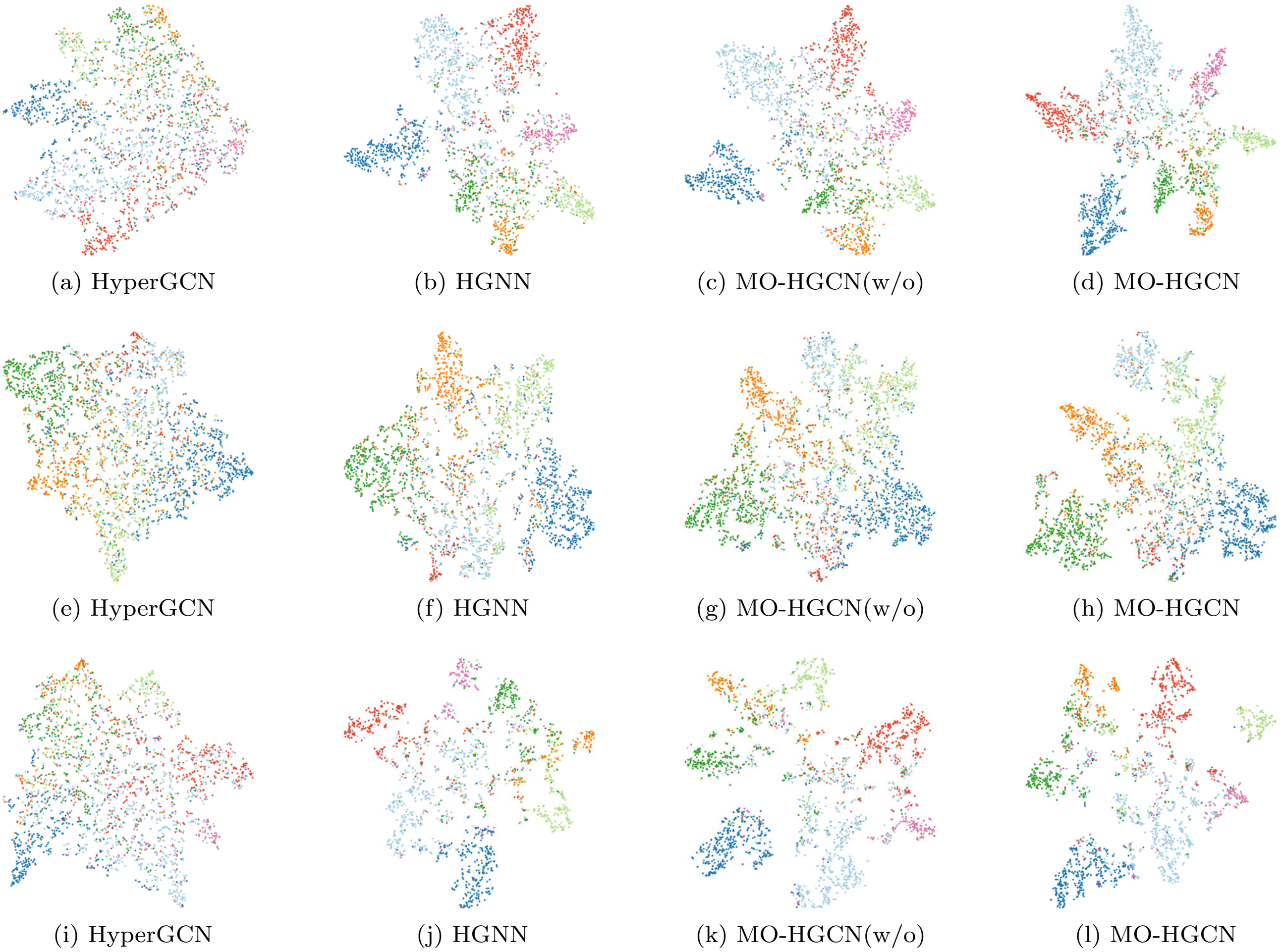


Fig. 7 2D visualization of T-SNE for node embedding on co-citation and co-authorship datasets. The first row represents the co-citation cora dataset, the second row represents the co-citation citeseer dataset, and the third row represents the co-authorship cora dataset. Note that we named the MO-HGCN without self-supervised learning as MO-HGCN (w/o)

ancy between the node embedding generated by the higher order channel and the enhanced information channel. This also verifies that the nodes are able to receive information from higher order neighborhoods. Higher order channel node embeddings that are more similar to those of the enhanced information channel are assigned higher scores. As a result, information with low similarity to 1-order neighbors (containing the node's information) has a lower weight in the fusion of embeddings.

Effectiveness of self-supervised learning

We conduct parameter sensitivity experiments on the hyperparameters of the self-supervised learning, i.e., problem **Q4**. For η_1 and η_2, which were chosen in the range. Figure 5 reports the node classification accuracy of MO-HGCN for different η_1 and η_2 ranges. Figure 5 shows the stable performance of the MO-HGCN when η_1 and η_2 are chosen in a suitable range. Moreover, since self-supervised learning as an auxiliary task, η_1 and η_2 contribute more to the performance of the MO-HGCN at smaller valuesb (Fig. 6).

To investigate the impact of self-supervised learning on the multi-order hypergraph convolutional network, i.e., question **Q5**, we visualize the node embeddings generated by HyperGCN, HGNN, MO-HGCN without self-supervised learning, and MO-HGCN, respectively. As shown in Fig. 7, we use T-SNE [35] to reduce the dimension of the node embeddings and perform projection of 2D coordinates to draw clusters, with each color representing a different class of nodes, respectively. To produce clear distributions, we test on the Cora and Citesseer datasets with a small number of nodes and report the clustering coefficients in Table 3.

In Fig. 7, the MO-HGCN produces clear clusters as a result of the node embeddings, and it also produces higher contour coefficients in Table 3 than the MO-HGCN without self-supervised learning. This shows that self-supervised

Table 3 Clustering coefficients on co-citation and co-authorship datasets

Dataset	Co-citation Cora	Co-citation Citeseer	Co-authorship Cora
HyperGCN	– 0.064	0.005	– 0.054
HGNN	0.112	0.108	0.135
MO-HGCN (w/o)	0.151	0.083	0.161
MO-HGCN	0.185	0.112	0.158

learning helps the node embeddings to learn distinctive information, which enables the separation of node embeddings to be improved.

Conclusions

In this paper, we propose a Multi-Order Hypergraph Convolutional Network incorporating self-supervised learning (MO-HGCN) to explore the potential of hypergraphs on higher order neighborhoods. MO-HGCN consists of a multi-channel network structure, where the higher order channels are composed of spectral 2-order and spectral 3-order hypergraph convolution operators, respectively. Through inter-order attention, we design an enhanced information channel that preserves low-order neighborhood information. To mine distinctive information in the higher order channels, we introduce self-supervised learning as an auxiliary task to enhance the performance of MO-HGCN. Experiments show that MO-HGCN is competitive with state-of-the-art baselines, and MO-HGCN develops the potential of higher order neighborhoods through inter-order attention and self-supervised learning components. In future work, we would like to explore hypergraphs with heterogeneous nodes to investigate higher order neighborhood problems on heterogeneous hypergraphs.

Acknowledgements This work was partly supported by the National Natural Science Foundation of China (U1701266), the Guangdong Provincial Key Laboratory Project of Intellectual Property and Big Data (2018B030322016), Special Projects for Key Fields in Higher Education of Guangdong, China (2020ZDZX3077, 2022ZDJS013), the Natural Science Foundation of Guangdong Province, China (2022A1515011146), Key Field R&D Plan Project of Guanzhou (202206070003), and the Youth Innovation Project of the Department of Education of Guangdong Province, China (File No. 2020KQNCX040).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abu-El-Haija S, Perozzi B, Kapoor A, Alipourfard N, Lerman K, Harutyunyan H, Ver Steeg G, Galstyan A (2019) Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In: International conference on machine learning. PMLR, pp 21–29
2. Arya D, Gupta DK, Rudinac S, Worring M (2020) Hypersage: generalizing inductive representation learning on hypergraphs. arXiv preprint arXiv:2010.04558
3. Bachman P, Hjelm RD, Buchwalter W (2019) Learning representations by maximizing mutual information across views. Adv Neural Inf Process Syst 32
4. Bai S, Zhang F, Torr PH (2021) Hypergraph convolution and hypergraph attention. Pattern Recogn 110:107637
5. Bandyopadhyay S, Das K, Murty MN (2020) Line hypergraph convolution network: Applying graph convolution for hypergraphs. arXiv preprint arXiv:2002.03392
6. Banka A, Buzi I, Rekik I (2020) Multi-view brain hyperconnectome autoencoder for brain state classification. In: International workshop on predictive intelligence in medicine. Springer, pp 101–110
7. Bretto A (2013) Hypergraph theory. An introduction. Mathematical engineering. Springer, Cham
8. Chen M, Wei Z, Huang Z, Ding B, Li Y (2020) Simple and deep graph convolutional networks. In: International conference on machine learning. PMLR, pp 1725–1735
9. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. Adv Neural Inf Process Syst 29
10. Dong Y, Sawin W, Bengio Y (2020) Hnhn: hypergraph networks with hyperedge neurons. arXiv preprint arXiv:2006.12278
11. Fan H, Zhang F, Wei Y, Li Z, Zou C, Gao Y, Dai Q (2021) Heterogeneous hypergraph variational autoencoder for link prediction. IEEE Trans Pattern Anal Mach Intell
12. Feng Y, You H, Zhang Z, Ji R, Gao Y (2019) Hypergraph neural networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 3558–3565
13. Gao Y, Zhang Z, Lin H, Zhao X, Du S, Zou C (2020) Hypergraph learning: methods and practices. IEEE Trans Pattern Anal Mach Intell
14. Georgiev D, Brockschmidt M, Allamanis M (2022) Heat: hyperedge attention networks. arXiv preprint arXiv:2201.12113
15. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. Adv Neural Inf Process Syst 30
16. Hao X, Li J, Guo Y, Jiang T, Yu M (2021) Hypergraph neural network for skeleton-based action recognition. IEEE Trans Image Process 30:2263–2275
17. Hassani K, Khasahmadi AH (2020) Contrastive multi-view representation learning on graphs. In: International conference on machine learning. PMLR, pp 4116–4126
18. Hjelm R.D, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A, Bengio Y (2018) Learning deep representations by

mutual information estimation and maximization. arXiv preprint arXiv:1808.06670
19. Huang J, Yang J (2021) Unignn: a unified framework for graph and hypergraph neural networks. arXiv preprint arXiv:2105.00956
20. Huang J, Lei F, Wang S, Wang S, Dai Q (2021) Hypergraph convolutional network with hybrid higher-order neighbors. In: Chinese conference on pattern recognition and computer vision (PRCV). Springer, pp 103–114
21. Jaiswal A, Babu AR, Zadeh MZ, Banerjee D, Makedon F (2020) A survey on contrastive self-supervised learning. Technologies 9(1):2
22. Jiang J, Wei Y, Feng Y, Cao J, Gao Y (2019) Dynamic hypergraph neural networks. In: IJCAI, pp 2635–2641
23. Jia R, Zhou X, Dong L, Pan S (2021) Hypergraph convolutional network for group recommendation. In: 2021 IEEE international conference on data mining (ICDM). IEEE, pp 260–269
24. Jing L, Tian Y (2020) Self-supervised visual feature learning with deep neural networks: a survey. IEEE Trans Pattern Anal Mach Intell 43(11):4037–4058
25. Jing B, Park C, Tong H (2021) Hdmi: high-order deep multiplex infomax. In: Proceedings of the web conference 2021, pp 2414–2424
26. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
27. Klicpera J, Bojchevski A, Günnemann S (2018) Predict then propagate: graph neural networks meet personalized pagerank. arXiv preprint arXiv:1810.05997
28. Kong L, d'Autume C.d.M, Ling W, Yu L, Dai Z, Yogatama D (2019) A mutual information maximization perspective of language representation learning. arXiv preprint arXiv:1910.08350
29. Liu S, Lv P, Zhang Y, Fu J, Cheng J, Li W, Zhou B, Xu M (2020) Semi-dynamic hypergraph neural network for 3d pose estimation. In: IJCAI, pp 782–788
30. Liu X, Zhang F, Hou Z, Mian L, Wang Z, Zhang J, Tang J (2021) Self-supervised learning: generative or contrastive. IEEE Trans Knowl Data Eng
31. Liu X, Lei F, Xia G, Zhang Y, Wei W (2022) Adjmix: simplifying and attending graph convolutional networks. Complex Intell Syst 8(2):1005–1014
32. Lostar M, Rekik I (2020) Deep hypergraph u-net for brain graph embedding and classification. arXiv preprint arXiv:2008.13118
33. Payne J (2019) Deep hyperedges: a framework for transductive and inductive learning on hypergraphs. arXiv preprint arXiv:1910.02633
34. Sun FY, Hoffmann J, Verma V, Tang J (2019) Infograph: unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv preprint arXiv:1908.01000
35. Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9(11)
36. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. arXiv preprint arXiv:1710.10903
37. Veličković P, Fedus W, Hamilton W.L, Liò P, Bengio Y, Hjelm RD (2018) Deep graph infomax. arXiv preprint arXiv:1809.10341
38. Wang J, Ding K, Zhu Z, Caverlee J (2021) Session-based recommendation with hypergraph attention networks. In: Proceedings of the 2021 SIAM international conference on data mining (SDM). SIAM, pp 82–90
39. Xia X, Yin H, Yu J, Wang Q, Cui L, Zhang X (2021) Self-supervised hypergraph convolutional networks for session-based recommendation. In: Proceedings of the AAAI conference on artificial intelligence, vol 35, pp 4503–4511
40. Xu K, Hu W, Leskovec J, Jegelka S (2018) How powerful are graph neural networks? arXiv preprint arXiv:1810.00826
41. Yadati N, Nimishakavi M, Yadav P, Nitin V, Louis A, Talukdar P (2019) Hypergcn: a new method for training graph convolutional networks on hypergraphs. Adv Neural Inf Process Systems 32
42. Yang C, Wang R, Yao S, Abdelzaher T (2020) Hypergraph learning with line expansion. arXiv preprint arXiv:2005.04843
43. Yi J, Park J (2020) Hypergraph convolutional recurrent neural network. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 3366–3376
44. Yu J, Yin H, Li J, Wang Q, Hung NQV, Zhang X (2021) Self-supervised multi-channel hypergraph convolutional network for social recommendation. In: Proceedings of the web conference 2021, pp 413–424
45. Zhang R, Zou Y, Ma J (2019) Hyper-sagnn: a self-attention based graph neural network for hypergraphs. arXiv preprint arXiv:1911.02613
46. Zhang J, Li F, Xiao X, Xu T, Rong Y, Huang J, Bian Y (2022) Hypergraph convolutional networks via equivalency between hypergraphs and undirected graphs. arXiv preprint arXiv:2203.16939
47. Zhou D, Huang J, Schölkopf B (2006) Learning with hypergraphs: clustering, classification, and embedding. Adv Neural Inf Process Syst 19
48. Zhu H, Koniusz P (2020) Simple spectral graph convolution. In: International conference on learning representations