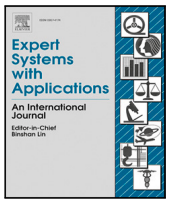
Expert Systems With Applications 277 (2025) 127242

Contents lists available at ScienceDirect

Expert Systems With Applications

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



SCHG: Spectral Clustering-guided Hypergraph Neural Networks for Multi-view Semi-supervised Learning

Yuze Wu [a], Shiyang Lan [a], Zhiling Cai [b], Mingjian Fu [a,*], Jinbo Li [c], Shiping Wang [a,d]

[a] College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
[b] College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou 350002, China
[c] China Unicom Research Institute, Beijing 100176, China
[d] Fujian Key Laboratory of Big Data Application and Intellectualization for Tea Industry, Wuyi University, Wuyishan 354300, China

ARTICLE INFO

Keywords:
Multi-view learning
Hypergraph neural network
Hypergraph construction
Global graph structure

ABSTRACT

Multi-view semi-supervised learning enables to efficiently leverage multi-view information as well as labeled and unlabeled data to solve practical problems. With graph neural networks, multi-view semi-supervised learning can be smooth and robust to the label propagation process. Hypergraph learning is an approach to hypergraph topology that aims to identify and exploit high-order relations on hypergraphs to uncover data beyond one-to-one in real-world applications. However, traditional hypergraph construction methods usually consider only local correlations between samples and may ignore dependencies that exist in the wider context of the dataset. In this paper, we propose a novel multi-view high-order correlation modeling method, where the connectivity of hyperedges is determined through clustering, and complementary information from each view is integrated via a hypergraph neural network. Inspired by the divisibility of graphs revealed by spectral graph theory, the proposed method works well to capture global high-order correlations within data and uncover potential manifolds. To assess the effectiveness of hypergraph modeling, we conduct a comprehensive evaluation of a multi-view semi-supervised node classification task. The experiments illustrate that the proposed approach achieves superior performance compared to current state-of-the-art algorithms and general hypergraph learning across eight datasets.

1. Introduction

In multi-view data, each instance is described from various perspectives or angles, constituting multiple views of the entity (Dong et al., 2018). For instance, webpages can be described by the document text and the anchor text of hyperlinks pointing to that page, while personal social media possesses textual, social network, and timeline views. Numerous studies indicate that properly integrating multi-view features helps to leverage their complementary nature to mitigate ambiguity and enhance the ultimate performance of learning tasks (Jiang et al., 2022; Li, Song et al., 2023; Zhou et al., 2024). To achieve better learning performance, many methods necessitate training classifiers on extensive annotated datasets. However, with the booming of data mining, multi-view learning is challenged by the scarcity of labeled data and the imbalance of unlabeled data (Cai et al., 2024; Luo et al., 2012; Tao et al., 2017). Semi-supervised learning effectively harnesses the potential information contained within a limited amount of labeled data and a substantial volume of unlabeled data, enhancing generalization capabilities compared to traditional supervision methods.

Graph neural networks, with their powerful information extraction ability, can effectively handle non-Euclidean data, injecting new vitality into semi-supervised learning. These methods have focused on devising an effective convolution operation that is capable of extracting features within the graph domain while considering information from various leap points of neighboring nodes. Technically, this advancement is typically articulated through neighborhood aggregation or message-passing schemes (Gilmer et al., 2017). Given that multi-view datasets lack a natural topology, much research (Chen, Fu et al., 2023; Xie et al., 2020) endeavors to jointly optimize feature and graph fusion within an end-to-end framework to uncover hidden supervision between samples. The information aggregation process of these methods is based on the assumption of one-to-one association of samples. In scenarios where nodes exhibit high similarity, this aggregation aids in preserving the consistency and accuracy of label spreading. Nevertheless, real-world data correlations go beyond pairwise associations, and samples within the same class in the feature space are not always highly similar which may surpass the modeling capacity of a simple graph

* Corresponding author at: College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China.
E-mail addresses: wuyuze28@163.com (Y. Wu), llanshiyang@163.com (S. Lan), zhilingcai@126.com (Z. Cai), sinceway@fzu.edu.cn (M. Fu), lijb7213@chinaunicom.cn (J. Li), shipingwangphd@163.com (S. Wang).

https://doi.org/10.1016/j.eswa.2025.127242
Received 18 October 2024; Received in revised form 24 February 2025; Accepted 10 March 2025
Available online 18 March 2025

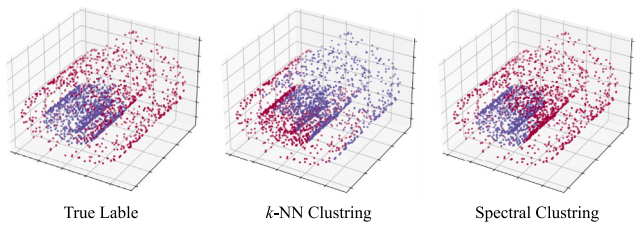


Fig. 1. Comparison of k-NN clustering and spectral clustering in dealing with manifolds. k-NN clustering often struggles to accurately group nodes at category boundaries, whereas spectral clustering better captures the underlying manifolds.

structure. For instance, in the task of categorizing a collection of papers into various topics, the relationships among these papers may exhibit a group-like pattern, where multiple papers share a common author. This association implies that papers authored by the same individual are more inclined to be related to the same topic, posing a challenge in representing such high-order correlations within a simple graph structure.

Hypergraph learning (Karypis et al., 1997; Li et al., 2024) uses hyperedges, which can connect multiple nodes at once, to model high-order correlations. The quality of the constructed hypergraph structure significantly impacts the efficacy of hypergraph learning. The widely used kNN-hypergraph construction involves identifying the closest neighbors for each node (Li, Yang et al., 2023). Essentially, for a given node in the feature space, a hyperedge can establish connections with its nearest neighbors. Thus, the network can capture underlying structures by modeling high-order neighborhoods of nodes. However, the kNN-hypergraph based solely on distance metrics only considers the local correlation of nodes, assuming that nodes within a specific neighborhood are more likely to be similar. This approach may struggle with handling "class-overlapping" and "class-imbalance" datasets (Vicente García & Sánchez, 2008).

The proposed method constructs a hypergraph based on clustering, where similar nodes are grouped to form higher-order relationships. This process generates diverse hyperedge structures across various metrics and scales, leveraging the scalability of hypergraph adjacency (Feng et al., 2019) to enhance complementary interactions between views. Acknowledging the constraints of traditional hypergraph learning in capturing solely local high-order correlations, we utilize spectral graph theory (Zhong et al., 2022) to incorporate the global structure. Therefore, our approach involves modeling the clusters identified by spectral clustering as hyperedges, enabling hypergraph learning to holistically consider the contextual information of each node. The hypergraph topology facilitates the modeling of high-order implicit manifolds, enabling efficient information transfer between nodes that share the same hyperedge. In Fig. 1, it is evident that spectral clustering has an advantage over k-NN in delineating category boundaries by mining manifolds. To promote message exchange across nodes on different hyperedges, we explore the scalability of hypergraph adjacency in a multi-view setting by concatenating hypergraphs from each view which allows for closer interactions between nodes with the same category. Finally, we introduce a hypergraph neural network to map the higher-order correlations across views into a shared feature space, enabling the extraction of complementary information between views. This view collaboration-based message passing mechanism not only captures local neighborhood features but also integrates global information from different views. The framework diagram illustrating this paper is depicted in Fig. 2.

The main contributions of the proposed work can be highlighted as follows:

- Introduce a novel and straightforward hypergraph modeling approach on clustering to facilitate the capture of the global latent structure in hypergraph learning using spectral graph theory.
- Adaptable to multi-view learning scenarios, the proposed method integrates complementary high-order properties of each view with the help of hyperedges at varied scales.
- Through experimental comparisons with state-of-the-art methods and traditional hypergraph learning, the effectiveness of the proposed method is validated.

The structure of the paper is as follows: Section 2 introduces the related work. Section 3 elaborates on the model proposed in this study. Section 4 showcases and analyzes the experimental outcomes of semi-supervised node classification. Lastly, Section 5 provides a summary of the paper.

2. Related work

In this section, we present the existing research in the areas of multi-view learning and hypergraph learning. By understanding the challenges and advancements in these areas, we can appreciate the significance of the proposed method in addressing the limitations of current approaches.

2.1. Multi-view learning

Multi-view learning involves modeling each perspective and collectively optimizing these models to improve tasks such as dimensionality reduction (Jiang et al., 2025; Sun et al., 2023; Zhang et al., 2021), clustering (Huang et al., 2019; Wang et al., 2019; Zhang et al., 2024), and classification (Cai et al., 2020; Li et al., 2019; Tao et al., 2018). A basic approach to the multi-view challenge is to merge data from all perspectives into a single view. However, this single-view strategy overlooks the complementary and consistent nature of multi-view data. To enhance the mutual information among different perspectives of the same entity, researchers have introduced a range of algorithms rooted in multi-view learning. Wang et al. (2020) employed graph convolutional networks to extract a shared embedding of node features and topology while leveraging the attention mechanism to learn adaptive weights for the embedding, but their approach faced limitations with unseen samples. In contrast, Jiang et al. (2023) made the classification of unseen samples not only rely on the labels of their direct neighbors, but also comprehensively consider the relationships with similar samples by exploiting the higher-order relationships of similar samples for feature propagation. To achieve a comprehensive discriminative representation with limited labels, Wang et al. (2024) proposed a knowledge distillation-driven multi-view classification algorithm that effectively mitigates non-informative features by employing a gating strategy and sparse regularization to dynamically assess the feature informativeness of samples. Zheng et al. (2021) achieved a shared multi-view subspace representation by incorporating first- and second-order graphs, mining local and global graph information, and combining auto-encoder networks and multi-view graph information. The global structure aids in understanding data relationships, inspiring our effort to integrate information from various perspectives.

2.2. Hypergraph learning

Due to the inherent advantages of hypergraph in effectively modeling complex node relationships, hypergraph learning (Agarwal et al., 2005) has emerged as a popular methodology for integrating diverse types of information to mitigate the impact of noise and outliers. The fundamental concept behind hypergraph learning revolves around the exploration of potential affinity relationships among nodes through the construction of hypergraphs, where hyperedges capture various affinities. This has been applied to many areas such as image matching (Zass & Shashua, 2008), multi-label classification (Wu et al., 2020), video object segmentation (Lv et al., 2018), and image retrieval (Huang et al.,

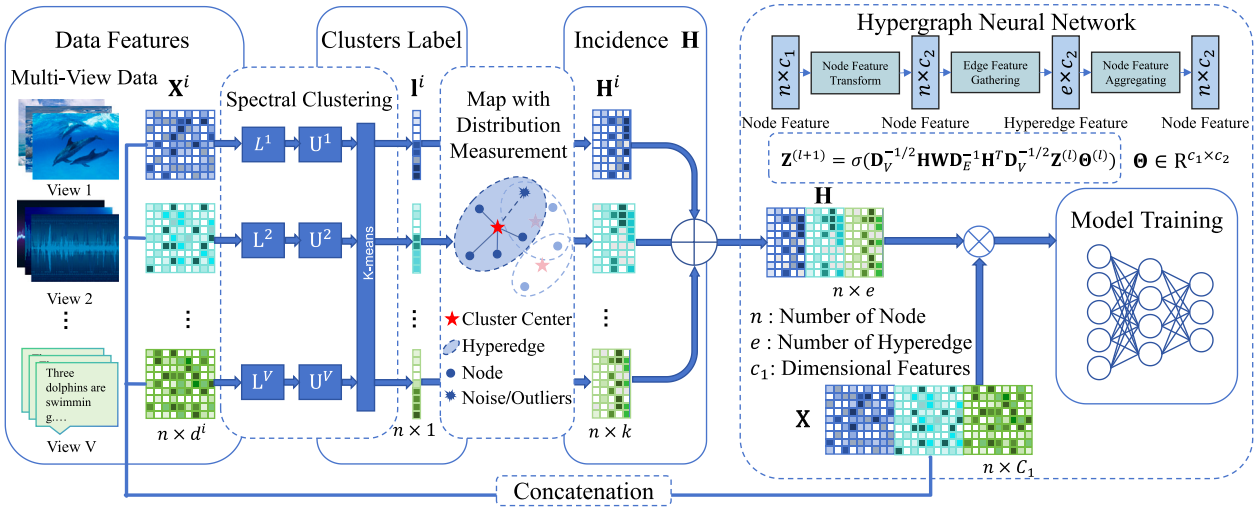


Fig. 2. An illustration of the proposed method. The hypergraph is constructed by spectral clustering and extended in the view direction, with adjacencies determined by the clusters (Fig. 3) and weights determined by the distance from the cluster centers (Eq. (7)). Moreover, node representation $\mathbf{Z}$ is learned by feeding $\mathbf{Z}^{(0)} = \mathbf{X}$ to the neural network, aiming to maximize high- and low-order correlations between nodes.

2010). For instance, Sun et al. (2008) accomplished hypergraph construction by progressively introducing new nodes into existing hyperedges through either cluster or star unfolding techniques. Meanwhile, the emergence of neural networks has injected new vigor into the study of hypergraph learning. Tu et al. (2018) highlighted the limitations of hypergraph-based embedding methods in sparse hypergraphs by using an autoencoder structure to preserve the second-order similarity, thus improving generalization in non-uniform heterogeneous hypergraphs. Zhang et al. (2019) utilized the classical self-attention mechanism to aggregate hypergraph information, constructing pairwise attention coefficients between nodes as dynamic features, and describing the nodes in conjunction with their static features. Drawing inspirations from graph convolutional networks, Gao et al. (2022) integrated the normalized hypergraph Laplacian into hypergraph neural networks, defining graph convolution on hypergraphs. And in multi-view data, there are problems of sparsity and feature space inconsistency among different views. To address these, Huang et al. (2023) proposed a non-negative matrix factorization (NMF) method combined with hypergraph regularization, which preserved higher-order geometric structures in the high-dimensional feature space and explicitly modelled view-specific features. Huang et al. (2024) proposed efficient hypergraph convolutional networks within an explainable optimization framework centered on a regularizer, while employing a low-rank approximation to transform the initial complex multi-view heterogeneous graph into hypergraphs. These approaches open up the possibility of the still immature hypergraph-based multi-view semi-supervised learning.

3. The proposed method

In this section, we propose the spectral clustering-guided multi-view hypergraph neural networks. For clarity, Table 1 summarizes the notations and definitions utilized in this paper, with bold uppercase letters representing matrices, bold lowercase letters representing vectors, and italicized text indicating constants.

3.1. Revisiting hypergraph neural network

We begin by reviewing the hypergraph neural network. A hypergraph extends the concept of a simple graph by allowing its edges to connect any number of nodes. Mathematically, a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with n nodes and m hyperedges consists of a set of nodes $\mathcal{V}$ with $v_i \in \mathcal{V}$, a set of hyperedges $\mathcal{E}$ with $e_i \in \mathcal{E}$, and a weight matrix $\mathbf{W}$ of hyperedges. The structure of $\mathcal{G}$ can be represented by an incidence

Table 1
Symbolic normalization notations with their descriptions.

Notations	Descriptions
V	The number of views.
C	The number of categories in the sample.
k	The number of clusters for clustering.
α	The factor for hyperedge distribution.
$\mathbf{X}^i \in \mathbb{R}^{n \times d^i}$	The given multi-view training data of the ith view.
$\mathbf{X} \in \mathbb{R}^{n \times \sum_{i=1}^{V} d^i}$	The multi-view training data for multi-view.
$\mathbf{A} \in \mathbb{R}^{n \times n}$	The adjacency matrix measured by similarity between nodes.
$\mathbf{L} \in \mathbb{R}^{n \times n}$	The regularized Laplacian matrix of $\mathbf{A}$.
$\mathbf{U} \in \mathbb{R}^{n \times k}$	The eigenvector matrix of the first k principal components.
$\mathbf{l} \in \mathbb{R}^{n \times 1}$	The category label vector obtained after clustering $\mathbf{U}$.
$\mathbf{H}^i \in \mathbb{R}^{n \times k}$	The incidence matrix of the ith view.
$\mathbf{H} \in \mathbb{R}^{n \times Vk}$	The incidence matrix for multi-view data.
$\mathbf{D}_v \in \mathbb{R}^{n \times n}$	The diagonal matrix of node degrees.
$\mathbf{D}_e \in \mathbb{R}^{Vk \times Vk}$	The diagonal matrix of hyperedge degrees.
$\mathbf{W} \in \mathbb{R}^{Vk \times Vk}$	The diagonal matrix of the edge weights.
$\mathbf{\Theta} \in \mathbb{R}^{c_1 \times c_2}$	The learnable parameter matrix in the network.

matrix $\mathbf{H} \in \mathbb{R}^{n \times m}$, where $h(v, e)$ indicates the presence of node v in hyperedge e and is defined as:

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

The degree of node v is defined as $d(v) = \sum_{e \in \mathcal{E}} w(e) h(v,e)$, and the degree of hyperedge e is defined as $d(e) = \sum_{v \in \mathcal{V}} h(v,e)$. For each node and hyperedge, their diagonal degree matrices $\mathbf{D}_v \in \mathbb{R}^{n \times n}$ and $\mathbf{D}_e \in \mathbb{R}^{m \times m}$ are defined as $[\mathbf{D}_v]_{ii} = \sum_j^m w(e_j)\mathbf{H}_{ij}$ and $[\mathbf{D}_e]_{jj} = \sum_i^n \mathbf{H}_{ij}$ respectively.

Utilizing second-order Chebyshev polynomials, the convolution operation on the signal $\mathbf{X}$ within a hypergraph can be succinctly characterized as follows:

$$\mathbf{Y} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{X}\mathbf{\Theta}, \tag{2}$$

where the filter $\mathbf{\Theta}$ is used to perform a filtering operation on the signal to extract features with specific frequencies and the convolved signal matrix $\mathbf{Y} \in \mathbb{R}^{n \times c_2}$ can be used for node-level tasks.

In the message passing on hypergraphs, hyperedges can be interpreted as factor nodes that connect multiple nodes, thereby establishing an equivalence between hypergraphs and factor graphs (Bianconi & Dorogovtsev, 2024). Analogous to Graph Neural Networks (GNNs), the hypergraph neural network model can be perceived as learning to

embed rooted subtrees into a lower-dimensional space. Rooted subtrees (Shervashidze et al., 2011) not only delineate the connectivity relationships between hypergraph nodes and factor nodes but also elucidate the paths through which messages propagate within the hypergraph. In the hypergraph convolution process outlined in Eq. (2), $\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{X}\boldsymbol{\Theta}$ denotes the aggregation of original node information to the hyperedge nodes, while $\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}$ involves reintegrating the information from the derived edge nodes back to the original nodes. Compared to traditional graph convolution, multi-layer hypergraph convolution enhances the prominence of the root node within the subtree path with more message interaction processes. This mechanism effectively captures both low-order and high-order correlations within the hypergraph.

3.2. Clustering-guided hypergraph construction

For a given dataset $\mathbf{X}^i$ under a single view, the n data vectors are considered as individual nodes on an undirected graph in the d^i-dimensional space. To quantify the similarity between each pair of nodes, a fully connected approach is employed to construct the adjacency matrix, while the Gaussian kernel is utilized for implementing the non-linear mapping:

$$\mathbf{A}_{i,j} = exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\mu^2}), \tag{3}$$

where $\|\cdot\|$ denotes the distance between samples i and j, generally taken as the Euclidean distance. Here, we simply consider the width of the Gaussian kernel as μ. The larger the value of $\mathbf{A}_{i,j}$ is, the more akin the two nodes are.

The clustering of the samples can be discerned based on the disparity between the least continuous eigenvalues of the normalized Laplacian matrix $\mathbf{L}$ of the connected graph. Therefore, through the eigen-decomposition of $\mathbf{L}$, we derive the eigenvector matrix corresponding to the k smallest eigenvalues:

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k] = eig(\mathbf{L}),\ \text{s.t. } \mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{4}$$

where $eig(\cdot)$ denotes the eigen-decomposition, $\mathbf{I}_n$ is a n-dimensional identity matrix and $\mathbf{D}$ is the degree matrix of $\mathbf{A}$. The eigenvector matrix can be treated as a new dataset with k-dimensional features, which can then be used for k-means clustering to obtain the label vector $\mathbf{l}$ corresponding to the categories of each sample.

After clustering, nodes within the same cluster are interconnected by one hyperedge, with each hyperedge corresponding to a specific cluster. Specifically, we map $\mathbf{l}$ to an $n \times k$ dimensional hypergraph incidence matrix $\mathbf{H}$:

$$h(v_i, e_j) = \begin{cases} 1, & \text{if } \mathbf{l}(i) = j, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where i refers to the index of node and j signifies the index of the cluster. In order to preserve the higher-order correlations across different perspectives and promote the information expressiveness of representations through view interactions, we take advantage of the scalability of hypergraph adjacency to integrate potential correlations between different views and construct a comprehensive hypergraph incidence matrix:

$$\mathbf{H} = concat([\mathbf{H}^1, \mathbf{H}^2, \dots, \mathbf{H}^V]). \tag{6}$$

For clarity, Fig. 3 exemplifies the process of mapping clusters to hypergraphs.

For the purpose of ensuring that the hyperedge accurately aggregates node information, we maintain the number of clusters to be greater than the number of sample categories, i.e., $k > C$, so that the true labels of nodes in the same hyperedge are as identical as possible. From a single-view perspective, message delivery is centralized within a single hyperedge, with barriers separating different hyperedges. This

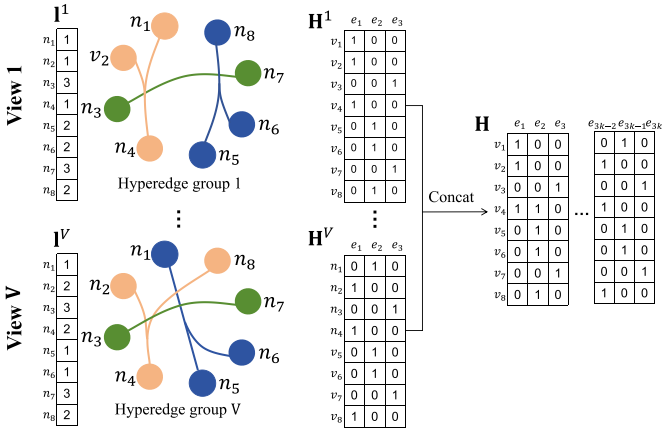


Fig. 3. Construction of a multi-view incidence matrix $\mathbf{H}$. The nodes were partitioned into 3 classes and represented by the label vector $\mathbf{l}$. Nodes sharing the same label are grouped within the same hyperedge, resulting in the construction of 3 hyperedges in each view. Subsequently, the proposed method is horizontally concatenated as $\{\mathbf{H}^v\}_{v=1}^V$.

structure helps to reduce the risk of nodes gathering incorrect information. However, the effectiveness of such propagation heavily relies on the quality of the initial clustering. Therefore, we can leverage complementary information from multiple views to ensure that nodes with the same label are more likely to be grouped within the same hyperedge in each view. Conversely, nodes with different labels that are erroneously clustered together may occur in a few views, a discrepancy that can be mitigated through representation learning. Thus, the proposed method ensures efficient message interaction between nodes of the same class, while communication between nodes of different classes is frustrated.

Considering that nodes situated at the periphery of a cluster exhibit weak similarity to nodes within the cluster, with a higher probability of being noisy or outliers, particularly in spectral clustering. Hence, we introduce the degree of node-to-hyperedge distribution to capture this distinction:

$$h(v_{t+i}, e_t) = exp(-\frac{\|\mathbf{x}_{t+i} - \mathbf{c}_t\|_2^2}{(\alpha \sum_j^{n_t} \|\mathbf{x}_{t+j} - \mathbf{c}_t\|/n_t)^2}), \tag{7}$$

where v_{t+i} denotes the ith sample point in the tth cluster, $\mathbf{c}_t$ denotes the center of the cluster to which v_{t+i} belongs, and n_t denotes the number of nodes in such a cluster. The function $h(v, e)$ is low enough to indicate that the sample point is more likely to be noise or an outlier, enabling the network to identify such sample and reduce their influence within the hyperedge. A smaller factor $\alpha > 0$ sharpens this distinction. The validity of this calculation is confirmed through the subsequent parameter sensitivity analysis and ablation experiments.

3.3. Hypergraph network architecture and model training

For the node classification problem with multi-view data $\{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^V\}$, a deep neural network is considered for training. So as to preserve the local information of the low convolutional layers and allow the model to consider feature details from multiple views simultaneously, we concatenate the embedding representation groups and hyperedge groups of all views. Subsequently, we obtain the feature matrix $\mathbf{X}$ and the incidence matrix $\mathbf{H}$. Based on the spectral convolution in the hypergraph, a hyperedge convolution layer $f(\mathbf{Z}, \mathbf{H}, \boldsymbol{\Theta})$ can be constructed:

$$\mathbf{Z}^{(l+1)} = \sigma(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{Z}^{(l)}\boldsymbol{\Theta}^{(l)}), \tag{8}$$

where σ is a nonlinear activation function, $\mathbf{Z}^{(l)} \in \mathbb{R}^{n \times d_l}$ denotes the latent feature space representation of the lth layer with $\mathbf{Z}^{(0)} = \mathbf{X}$, and $\boldsymbol{\Theta}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ is the parameter of the lth layer to be learned during the training process.

After several layers of forward propagation, we obtain the embedding representation of each node and map it to a probability distribution via softmax. The softmax activation function is defined as:

$$\mathbf{Z}_{ij} = softmax(\mathbf{Z}_{ij}^{(l)}) = \frac{exp(\mathbf{Z}_{ij}^{(l)})}{\sum_{k=1}^{C} exp(\mathbf{Z}_{ik}^{(l)})}, \tag{9}$$

For semi-supervised multiclass classification, we utilize the cross-entropy error over all labeled samples:

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{C} \mathbf{Y}_{lf} \ln \mathbf{Z}_{lf}, \tag{10}$$

where $\mathcal{Y}_L$ represents the set of indices corresponding to nodes that have labels, $\mathbf{Y}$ denotes the category of all samples and $\mathbf{Z}$ is the node embedding obtained after softmax in the output layer.

Model performs one step of forward propagation and backward propagation in each iteration to update its network parameters according to the loss function. The time complexities of computation of the similarity matrix $\mathbf{A}$ and the first k principal components $\mathbf{U}$, k-means clustering of $\mathbf{U}$, hypergraph construction, and neural network training are: $\mathcal{O}(n^2), \mathcal{O}(kn^2), \mathcal{O}(knt), \mathcal{O}(nk), \mathcal{O}(Lnd)$. Accordingly, we can compute the simplified time complexity of the proposed method:

$$\mathcal{O} = \mathcal{O}(kn^2 + knt + Lnd). \tag{11}$$

where t is the iterations in k-means and L is the number of layers in the neural network. The algorithmic flow of the proposed method is shown in Algorithm 1.

Algorithm 1 Spectral Clustering-guided HyperGraph neural networks for multi-view semi-supervised learning (SCHG)

Input: Multi-view data $\{\mathbf{X}^v\}_{v=1}^V$, number of clusters k, factor α, training epoch number t;
Output: The embedding representation matrix for each node, $\mathbf{Z} \in \mathbb{R}^{n \times C}$;

```
Initialize learnable parameter Θ;
Obtain Laplacian matrices {L^v}_{v=1}^V;
Generate the label vector {l^v}_{v=1}^V by the spectral clustering;
Calculate initial {H^v}_{v=1}^V by Eq. (5);
Update {H^v}_{v=1}^V that considers the degree of distribution by Eq. (7);
Calculate H, X by concatenating {H^v}_{v=1}^V, {X^v}_{v=1}^V respectively;
for i = 1 to t do
    Update Z by Eq. (9);
    Calculate the classification loss by Eq. (10);
    Update Θ by back propagation;
end for
return representation Z.
```

4. Experiments

In this section, we assess the performance of the proposed technique in node classification across eight authentic datasets under semi-supervised conditions. We juxtapose its results against those of eight cutting-edge comparison algorithms. Subsequently, we analyze parameter sensitivity and explore the clustering-guided global structure and distribution measurement module.

4.1. Dataset description

The proposed method undertakes a semi-supervised node classification assignment using eight authentic multi-view datasets: ALOI, MFeat, Caltech20, MNIST, Youtube, ORL, MSRC, and NoisyMNIST. The statistical details of these datasets are consolidated in Table 2 and briefly outlined below:

Table 2
A brief description of the homogeneous datasets.

Datasets	# Samples	# Views	# Feature dimensions	# Classes
ALOI	1079	4	64/64/77/13	10
MFeat	2000	6	216/76/64/6/240/47	10
Caltech20	2386	6	48/40/254/1,984/512/928	20
MNIST	10,000	3	30/9/9	10
Youtube	2000	6	2000/1024/64/512/64/647	10
ORL	400	4	512/59/864/254	40
MSRC	210	5	24/576/512/256/254	7
NoisyMNIST	30,000	2	784/784	10

ALOI[1] contains 1079 images of objects, including tools, furniture, animals, etc., taken under different rotation angles and lighting conditions. Four commonly used feature representations are available, including 64-D RGB color histogram, 64-D HSV color histogram, 77-D color similarity and 13-D Haralick features.

MFeat[2] is a collection of binary images that depict handwritten Arabic numerals, with each category containing 200 images. The dataset is represented in six views, including 216-D profile correlations, 76-D Fourier coefficients, 64-D Karhunen-Love coefficients, 6-D morphological features, 240-D pixel averages within 2 × 3 windows and 47-D Zernike moments.

Caltech20[3] is a dataset featuring real-world scenes characterized by diversity and complexity, encompassing 20 classes of images. It employs six distinct sets of extracted features: 48-D Gabor features, 40-D Wavelet Moments features, 254-D CENTRIST features, 1984-D HOG features, 512-D GIST features and 928-D LBP features.

MNIST[4] is a renowned dataset consisting of handwritten digits. Each sample is represented by three types of features, including 30-D IsoProjection, 9-D Linear Discriminant Analysis and 9-D Neighborhood Preserving Embedding.

Youtube[5] is a dataset of video games described by six audio and visual features: 2000-D Mel-frequency cepstral coefficients, 1024-D Spectrogram Stream, 64-D Histogram Motion Estimate, 512-D Cuboids Histogram, 64-D Volume Stream and 647-D Histogram of Oriented Gradients.

ORL[6] is a dataset comprising 40 distinct face images, with each subject having 10 images captured at different times with lighting, facial expressions and facial details.

MSRC[7] is a dataset consisting of 210 images from 7 different subjects. For each image, five features can be extracted: 24-D CM features, 576-D HOG features, 512-D GIST features, 256-D LBP features and 254-D CENTRIST features.

NoisyMNIST[8] is a handwritten dataset of 30,000 samples randomly selected from the MNIST image database, covering 10 classes. Each sample has two noisy views, with white Gaussian noise added at varying intensities.

4.2. Compared algorithms

In order to evaluate the strengths of the proposed algorithm and to verify the feasibility of the method, we compare it with nine mainstream algorithms in the multi-view semi-supervised node classification task. These algorithms include: Co-GCN, GCN-fusion, ERL-MVC, DSRL, JFGCN, GEGCN, ECMGD and ORLNet. The state-of-the-art algorithms are described as follows:

[1] https://elki-project.github.io/datasets/multiview
[2] http://archive.ics.uci.edu/ml/datasets/Multiple+Features
[3] http://www.vision.caltech.edu/ImageDatasets/Caltech20/
[4] http://yann.lecun.com/exdb/mnist/
[5] http://archive.ics.uci.edu/ml/machine-learning-databases/00269/
[6] http://cam-orl.co.uk/facedatabase.html
[7] http://riemenschneider.hayko.at/vision/dataset
[8] https://github.com/nineleven/NoisyMNISTDetection

Table 3
Node classification accuracy (mean% and standard deviation%) with optimal outcomes emphasized in bold and sub-optimal outcomes underscored. Out-of-memory errors are marked with "OOM".

Methods	ALOI	MFeat	Caltech20	MNIST	Youtube	ORL	MSRC	NoisyMNIST
Co-GCN	79.97 (1.98)	87.84 (1.84)	71.05 (2.11)	92.02 (0.53)	29.28 (0.27)	61.72 (1.08)	57.00 (1.78)	90.60 (1.71)
GCN-fusion	86.40 (1.51)	92.78 (1.01)	80.90 (1.73)	92.57 (0.18)	49.01 (0.50)	65.39 (1.76)	84.83 (1.51)	86.24 (1.98)
ERL-MVC	87.88 (1.30)	48.40 (0.82)	82.64 (0.88)	91.73 (0.12)	45.22 (0.96)	44.87 (0.43)	66.67 (4.49)	90.39 (0.02)
DSRL	90.91 (0.85)	95.74 (0.46)	83.76 (1.28)	94.00 (0.21)	44.74 (0.80)	55.06 (2.71)	77.04 (7.10)	OOM
JFGCN	95.32 (1.94)	96.98 (0.07)	84.65 (0.04)	93.68 (0.20)	59.41 (0.64)	72.94 (3.43)	84.81 (0.34)	OOM
GEGCN	88.96 (1.18)	96.78 (0.29)	83.00 (0.87)	93.30 (0.14)	54.67 (0.31)	61.39 (0.81)	83.60 (0.89)	OOM
ECMGD	94.36 (1.20)	94.63 (0.39)	87.72 (0.49)	90.29 (0.33)	59.83 (0.40)	64.03 (2.02)	80.53 (0.85)	OOM
ORLNet	83.63 (2.27)	88.97 (2.0)	75.51 (1.95)	89.3 (0.44)	53.36 (2.19)	41.89 (7.38)	73.39 (4.84)	80.09 (2.69)
SCHG	**95.58 (0.27)**	**97.74 (0.04)**	**87.73 (0.14)**	**94.06 (0.05)**	**61.39 (0.21)**	**80.36 (0.18)**	**87.51 (0.26)**	**93.07 (0.11)**

Table 4
Node classification F1-score (mean% and standard deviation%) with optimal outcomes emphasized in bold and sub-optimal outcomes underscored. Out-of-memory errors are marked with "OOM".

Methods	ALOI	MFeat	Caltech20	MNIST	Youtube	ORL	MSRC	NoisyMNIST
Co-GCN	79.13 (2.44)	87.81 (1.89)	45.55 (1.71)	91.86 (0.45)	22.45 (1.20)	62.99 (1.00)	57.47 (1.55)	90.12 (1.93)
GCN-fusion	86.51 (1.54)	92.75 (1.10)	62.31 (2.51)	92.46 (0.18)	48.54 (0.79)	63.01 (2.67)	84.07 (1.46)	86.07 (2.27)
ERL-MVC	88.42 (1.06)	47.62 (1.11)	66.01 (1.54)	91.63 (0.10)	47.88 (0.89)	44.39 (0.39)	73.00 (4.35)	90.39 (0.02)
DSRL	91.00 (0.79)	95.75 (0.46)	65.52 (3.88)	93.91 (0.21)	44.00 (0.94)	47.70 (3.15)	74.86 (8.31)	OOM
JFGCN	95.46 (1.86)	97.07 (0.07)	70.65 (0.16)	93.78 (0.18)	59.74 (0.70)	73.20 (1.96)	86.15 (0.27)	OOM
GEGCN	89.80 (1.24)	96.83 (0.30)	65.60 (0.96)	93.20 (0.18)	55.86 (0.34)	54.76 (0.83)	83.29 (0.97)	OOM
ECMGD	94.39 (1.20)	94.65 (0.38)	71.27 (1.14)	88.96 (0.39)	59.48 (0.39)	61.74 (2.17)	79.85 (0.88)	OOM
ORLNet	83.26 (2.44)	88.96 (1.98)	44.63 (4.36)	89.11 (0.44)	53.36 (2.19)	41.89 (7.38)	73.39 (4.84)	80.09 (2.69)
SCHG	**95.60 (0.26)**	**97.74 (0.04)**	**71.98 (0.28)**	**93.98 (0.05)**	**61.10 (0.22)**	**79.63 (0.15)**	**86.95 (0.30)**	**93.28 (0.16)**

Co-GCN (Li et al., 2020): This algorithm introduces GCN to multi-view learning by adaptively utilizing graph information from multiple views and a combined Laplacian matrix.

GCN-fusion: In this method, each view can use its own graph to run GCN, and then aggregate these GCN models for prediction. This approach allows each view to adaptively learn its own structure information and integrate this information into the final prediction.

ERL-MVC (Huang et al., 2021): This algorithm accomplishes semi-supervised node classification by integrating diverse multi-view data and leveraging an embedding regularizer learning scheme to exploit the sparse local structure and enhance consensus, thereby enabling accurate classification with limited labeled data.

DSRL (Wang et al., 2022): This algorithm introduces a deep sparse regularizer learning model for adaptive learning of deep sparse regularizers. Specifically, the model constructs a neural network with multiple differentiable and reusable blocks to extract discriminative features.

JFGCN (Chen, Wu et al., 2023): This algorithm proposes an end-to-end federated fusion framework, aiming at both consistent feature integration and adaptive topology adjustment to capture the consistency and complementarity of multi-view data.

GEGCN (Lu, Wu, Zhong et al., 2024): The algorithm enhances the purity of graph embeddings by extracting shared representations, constructing coherent graphs, and introducing a denoising GCN with a learnable threshold shrinkage function.

ECMGD (Lu, Wu, Chen et al., 2024): The algorithm proposes an energy-constrained multi-view graph diffusion framework, and designs a feature propagation process with inter-view perception by considering inter- and intra-view feature flows throughout the system.

ORLNet (Fang et al., 2024): This algorithm integrates optimization-derived components into a deep neural network, ensuring interpretability and flexibility, while converting convex optimization solutions into feed-forward layers and incorporating view-wise weights for multi-view learning.

4.3. Experimental setup

Following the setup in prior works, default parameter settings are used for all comparison algorithms. A consistent 10% of randomly labeled data is employed for all algorithms. The number of training iterations for the hypergraph convolutional neural network within the proposed approach is uniformly set to 1000. The Adam optimizer is utilized to update the learnable parameters of each layer. The learning rate of the model is set to {0.001, 0.01, 0.03}. Network weight decay is implemented through the l_2 regularization term to promote the use of smaller weights and mitigate overfitting risks. The network architecture is comprised of two hidden layers along with one output layer. The dimension of first layer is determined by the feature characteristics of the input data, while the second layer is consistently set to 128. Softmax activation function is applied in the output layer, with ReLU activation utilized in all preceding layers. The number of clusters k is typically set to 5%–25% of the total number of samples, depending on dataset size and the number of true classes. In large datasets with sparse sample similarity, a smaller k helps to capture the global structure and prevent over-segmentation. As the number of true classes increases, the hypergraph requires a larger k to ensure proper connections among nodes, reflecting intra-class similarities and inter-class differences. The hyperedge factor α is set to 1–5, in accordance with the state-of-the-art works settings. The implementation of the framework is implemented on the Pytorch platform on a system equipped with an R9-5900X CPU, a Nvidia RTX 3060 GPU, and 60 GB of RAM.

4.4. Multi-view semi-supervised classification

Comparison of classification results: We conduct an evaluation of the proposed SCHG algorithm on 8 multi-view node classification datasets. Tables 3 and 4 present the classification accuracies and F1 scores for the state-of-the-art algorithms and SCHG, respectively. Our observations indicate that SCHG achieves optimal classification performance across all datasets. In particular, its accuracy on Youtube, ORL, and MSRC improves by 2.38%, 7.42%, and 2.70%, respectively. Notably, SCHG exhibits lower variance (0.16 on average) for semi-supervised classification compared to other algorithms (1.56 on average), underscoring the robustness of the approach.

Furthermore, in Fig. 4, we compare the classification accuracy of each algorithm alongside SCHG, adjusting the supervised ratio from 5% to 50%. Our analysis reveals that SCHG outperforms other algorithms

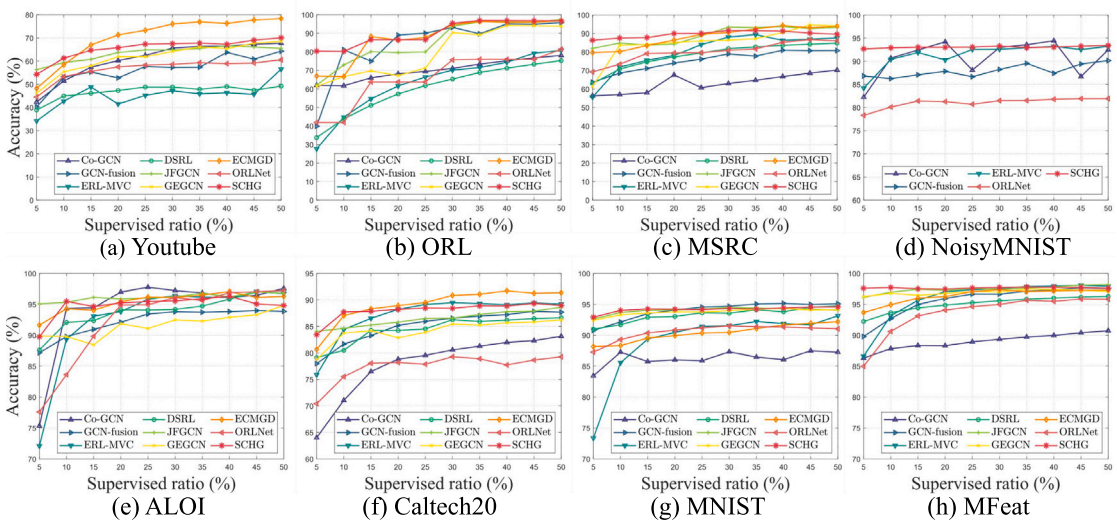


Fig. 4. The accuracy of the proposed method as the ratio of labeled data ranges in {0.05, 0.10, …, 0.5} on the eight datasets.

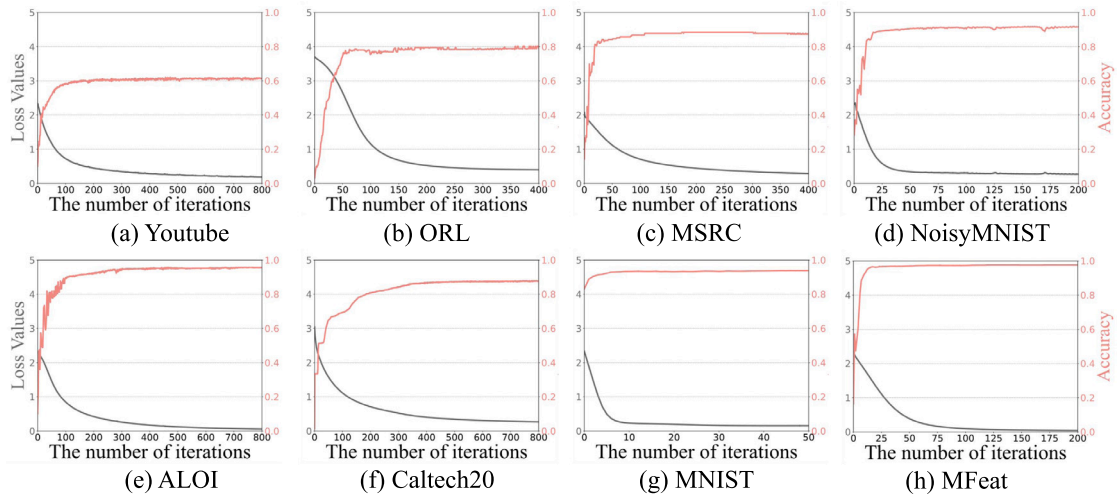


Fig. 5. Curves of loss value (black) and accuracy (red) of the proposed method with 10% labeled samples on eight datasets.

at lower ratios of labeled samples, while other methods necessitate significantly higher supervision rates to attain comparable accuracy levels. In summary, SCHG demonstrates superior performance in the semi-supervised node classification task, showcasing its effectiveness and robustness.

Learning process evaluation: To further evaluate the effectiveness of the proposed SCHG learning process, Fig. 5 illustrates the variation curves of accuracy and loss at each iteration across eight datasets. The results demonstrate a rapid convergence of both loss value and accuracy metrics, which eventually stabilize post-convergence. Particularly on the MNIST dataset, stability is achieved around the 30th iteration. This outcome suggests that the model efficiently learns the input-target relationship without succumbing to issues like local optima or gradient disappearance/explosion.

Parameter sensitivity analysis: We further explore the impact of the parameters α and k on constructing the hypergraph and its subsequent effect on classification accuracy. Here, α controls the smoothing degree, influencing how nodes are distributed to hyperedges, while k denotes both the number of classes in spectral clustering and the number of hyperedges. A larger α leads to more similar distributions among nodes, with distributions reaching 1 at infinity. To standardize the metric, we employ the ratio k/n of the number of samples instead of k and Fig. 6 visually depicts the data distribution pattern. As shown in Fig. 6(b), when the number of hyperedges k is too small, even smaller than that of sample categories, the hyperedges tend to aggregate samples from different clusters, resulting in the propagation of incorrect labeling information directly on the hyperedges. Conversely, an excessive number of hyperedges partitions the dataset excessively, dispersing similar samples across different hyperedges, making it challenging for the hypergraph neural network to capture intraclass correlations, thereby affecting classification accuracy. The impact of the factor α is minimal in large datasets like MNIST and NoisyMNIST, with fluctuations of only 0.5%. However, it exhibits a more significant effect on relatively smaller datasets, where the performance boost from Eq. (7) can reach up to 7%.

4.5. Ablation analysis

To validate the effectiveness of the proposed hypergraph modeling technique in capturing high-order correlations globally, we compare the semi-supervised classification outcomes of kNN-Hypergraph and SC-Hypergraph, with optimized network parameters and neighbor numbers. In addition, we investigate the impact of the distribution measurement by setting the distribution of all nodes to 1 and denoting it as w/o DM. The compared results are depicted in Fig. 7. Compared with the kNN-hypergraph, the proposed method is more capable of

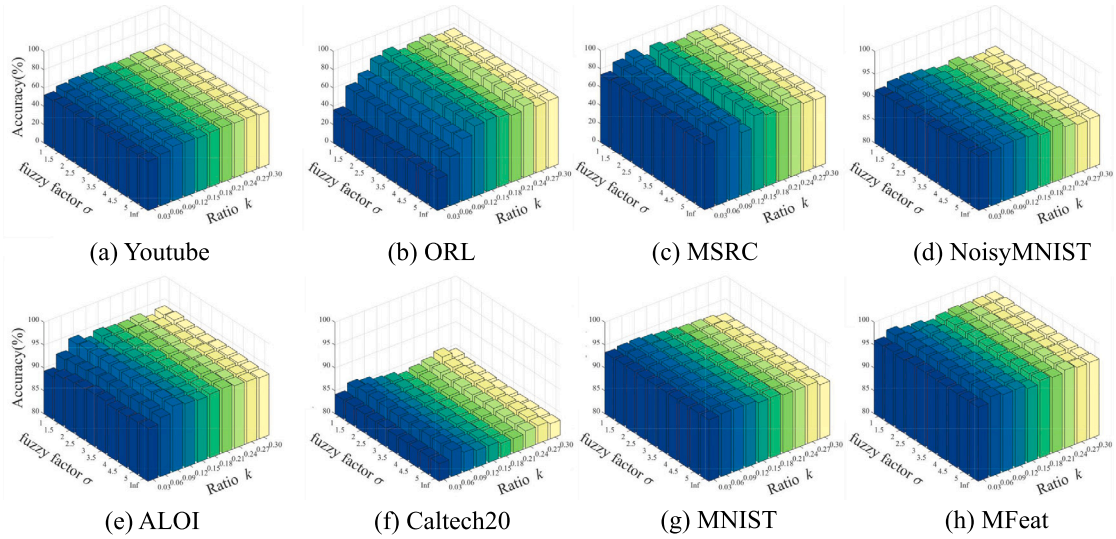


Fig. 6. The parameter sensitivity analysis (accuracy %) of the proposed method on the tested datasets with respect to ratio k and factor α for node-to-hyperedge distribution degrees.

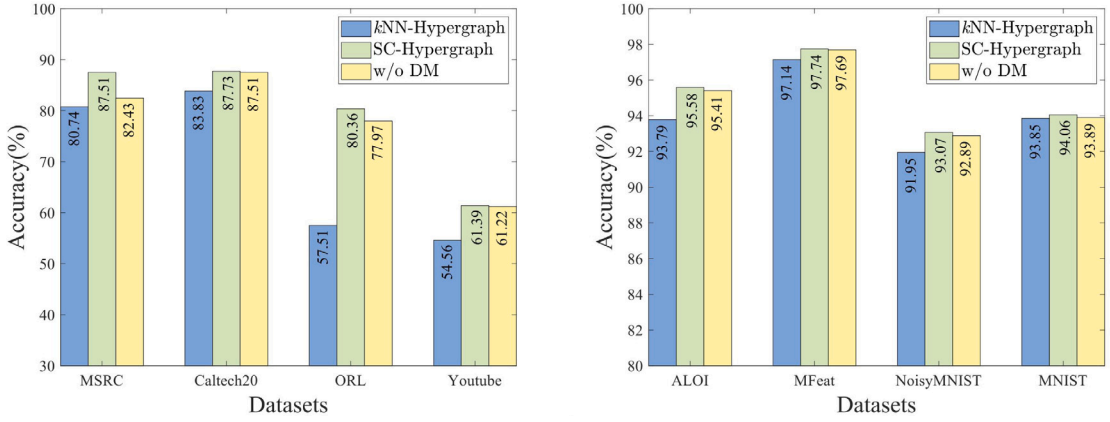


Fig. 7. Comparison of semi-supervised classification performance (accuracy(%)) of kNN-Hypergraph without considering global structure, SCHG, and SCHG without distribution measurement.

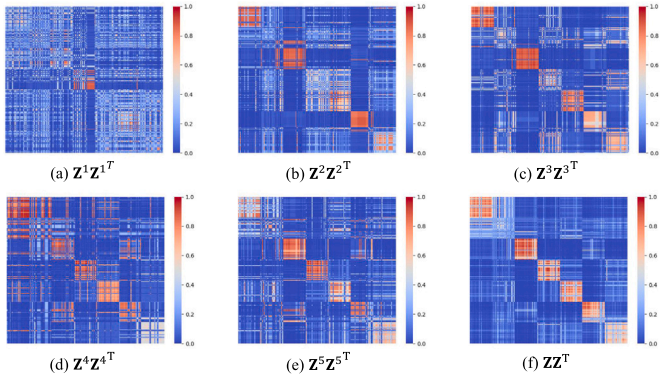


Fig. 8. Visualization of node embedding $\mathbf{Z} \in \mathbb{R}^{N\times F}$ of the learning process on MSRC, where hypergraphs are constructed by the single view (a)–(e) and by mulitple views (f).

identifying potential groups in the data by considering the global structure, and capturing the sample relationships in the scarce classes, while reducing the influence of noise and outliers through the collaboration of other views (e.g., 3.9% increase on the class-imbalanced dataset Caltech20 and 22.75% increase on ORL). Meanwhile, the proposed distribution measurement module effectively improves classification performance (e.g., 5.08% increase on MSRC and 2.39% increase on ORL). By modeling the degrees of distribution, this module allows the hypergraph neural network to take into account the relative positions and significance of nodes. This enhances the diversity of higher-order relationships and allows the model to more effectively identify hidden supervised signals.

4.6. Multi-view integration analysis

As the number of views increases, SCHG efficiently extends the high-order interactions of the hypergraph through simple concatenation, gradually enhancing classification performance. In this experiment, we investigate whether SCHG effectively integrates information from individual views by inputting the **H** matrix of each view into the network individually for supervision to assess the contribution of each view. Fig. 8 illustrates the visualized layer embeddings on the dataset MSRC. It is evident that when SCHG is constructed using a single view, the label information struggles to propagate to similar nodes on different hyperedges, thereby limiting the expressive power of the hypergraph. In contrast, the multi-view learning approach ensures a robust propagation mechanism.

5. Conclusion

In this paper, we introduce a framework for hypergraph modeling in multi-view scenarios. Our approach focuses on exploring manifold structures to construct high-order correlations among nodes and extends the hypergraph along the view direction to facilitate message coupling between hyperedges. We utilize spectral clustering for each view, translating the clustering outcomes into the hypergraph adjacency matrix, and generate node embeddings through training with a hypergraph convolutional neural network. Compared to distance-based hypergraphs, the proposed method demonstrates superior learning performance and effectively leverages the complementary information from multiple views. Through experiments on a semi-supervised node classification task, we assess the efficacy of the proposed approach. By capturing complementary high-order correlations across individual views, the proposed method showcases potential for diverse applications in multi-view learning. However, the time complexity is relatively high, which makes it less suitable for large datasets. Additionally, the number of hyperedges must be set manually. Future research will focus on optimization algorithms to improve model efficiency and explore adaptive training strategies for automatically determining the optimal hypergraph construction.

CRediT authorship contribution statement

Yuze Wu: Methodology, Writing – original draft, Software. **Shiyang Lan:** Writing – original draft, Investigation, Software. **Zhiling Cai:** Methodology, Writing – review & editing. **Mingjian Fu:** Supervision, Writing – review & editing. **Jinbo Li:** Supervision. **Shiping Wang:** Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Shiping Wang reports financial support was provided by National Natural Science Foundation of China.

Acknowledgments

This work is in part supported by the National Natural Science Foundation of China under Grants U21A20472 and 62276065, the Fujian Provincial Natural Science Foundation of China under Grants 2024J01510026, 2022J01117 and JZ230011, and the Open Research Fund from the Fujian Provincial Key Laboratory of Big Data Application and Intellectualization for Tea Industry, Wuyi University, under Grant FKLBDAITI202302.

Data availability

No data was used for the research described in the article.

References

Agarwal, Sameer, Lim, Jongwoo, Zelnik-Manor, Lihi, Perona, Pietro, Kriegman, David, & Belongie, Serge (2005). Beyond Pairwise Clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 838–845).

Bianconi, Ginestra, & Dorogovtsev, Sergey N. (2024). Theory of Percolation on Hypergraphs. *Physical Review E*, *109*(1), Article 014306.

Cai, Ronggang, Chen, Hongmei, Mi, Yong, Luo, Chuan, Horng, Shi-Jinn, & Li, Tianrui (2024). Feature-guided Multi-view Clustering by Jointing Local Subspace Label learning and Global Label learning. *Expert Systems with Applications*, *252*, Article 124191.

Cai, Hao, Liu, Bo, Xiao, Yanshan, & Lin, LuYue (2020). Semi-supervised Multi-view Clustering Based on Orthonormality-constrained Nonnegative Matrix Factorization. *Information Sciences*, *536*, 171–184.

Chen, Zhaoliang, Fu, Lele, Yao, Jie, Guo, Wenzhong, Plant, Claudia, & Wang, Shiping (2023). Learnable Graph Convolutional Network and Feature Fusion for Multi-view Learning. *Information Fusion*, *95*, 109–119.

Chen, Yuhong, Wu, Zhihao, Chen, Zhaoliang, Dong, Mianxiong, & Wang, Shiping (2023). Joint Learning of Feature and Topology for Multi-view Graph convolutional Network. *Neural Networks*, *168*, 161–170.

Dong, Xiao, Zhu, Lei, Song, Xuemeng, Li, Jingjing, & Cheng, Zhiyong (2018). Adaptive Collaborative Similarity Learning for Unsupervised Multi-view Feature Selection. In *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 2064–2070).

Fang, Zihan, Du, Shide, Cai, Zhiling, Lan, Shiyang, Wu, Chunming, Tan, Yanchao, & Wang, Shiping (2024). Representation Learning Meets Optimization-derived Networks: From Single-view to Multi-view. *IEEE Transactions on Multimedia*, *26*, 8889–8901.

Feng, Yifan, You, Haoxuan, Zhang, Zizhao, Ji, Rongrong, & Gao, Yue (2019). Hypergraph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 3558–3565).

Gao, Yue, Feng, Yifan, Ji, Shuyi, & Ji, Rongrong (2022). HGNN+: General Hypergraph Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(3), 3181–3199.

Gilmer, Justin, Schoenholz, Samuel S, Riley, Patrick F, Vinyals, Oriol, & Dahl, George E (2017). Neural Message Passing for Quantum Chemistry. In *Proceedings of the International Conference on Machine Learning* (pp. 1263–1272).

Huang, Aiping, Fang, Zihan, Wu, Zhihao, Tan, Yanchao, Han, Peng, Wang, Shiping, & Zhang, Le (2024). Multi-view Heterogeneous Graph Learning with Compressed Hypergraph Neural Networks. *Neural Networks*, *179*, Article 106562.

Huang, Shudong, Kang, Zhao, Tsang, Ivor W., & Xu, Zenglin (2019). Auto-weighted Multi-view Clustering via Kernelized Graph Learning. *Pattern Recognition*, *88*, 174–184.

Huang, Yuchi, Liu, Qingshan, Zhang, Shaoting, & Metaxas, Dimitris N (2010). Image Retrieval via Probabilistic Hypergraph Ranking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 3376–3383).

Huang, Aiping, Wang, Zheng, Zheng, Yannan, Zhao, Tiesong, & Lin, Chia-Wen (2021). Embedding Regularizer Learning for Multi-view Semi-supervised Classification. *IEEE Transactions on Image Processing*, *30*, 6997–7011.

Huang, Haonan, Zhou, Guoxu, Liang, Naiyao, Zhao, Qibin, & Xie, Shengli (2023). Diverse Deep Matrix Factorization with Hypergraph Regularization for Multi-view Data Representation. *IEEE/CAA Journal of Automatica Sinica*, *10*(11), 2154–2167.

Jiang, Bingbing, Liu, Jun, Wang, Zidong, Zhang, Chenglong, Yang, Jie, Wang, Yadi, Sheng, Weiguo, & Ding, Weiping (2025). Semi-supervised Multi-view Feature Selection with Adaptive Similarity Fusion and Learning. *Pattern Recognition*, *159*, Article 111159.

Jiang, Bingbing, Wu, Xingyu, Zhou, Xiren, Liu, Yi, Cohn, Anthony G, Sheng, Weiguo, & Chen, Huanhuan (2022). Semi-supervised Multiview Feature Selection with Adaptive Graph Learning. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(3), 3615–3629.

Jiang, Bingbing, Zhang, Chenglong, Zhong, Yan, Liu, Yi, Zhang, Yingwei, Wu, Xingyu, & Sheng, Weiguo (2023). Adaptive Collaborative Fusion for Multi-view Semi-supervised Classification. *Information Fusion*, *96*, 37–50.

Karypis, George, Aggarwal, Rajat, Kumar, Vipin, & Shekhar, Shashi (1997). Multilevel Hypergraph Partitioning: Application in Vlsi Domain. In *Proceedings of the Annual Design Automation Conference* (pp. 526–529).

Li, Shu, Li, WenTao, & Wang, Wei (2020). Co-gcn for Multi-view Semi-supervised Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 4691–4698).

Li, Qingfeng, Ma, Huifang, Jin, Wangyu, Ji, Yugang, & Li, Zhixin (2024). Hypergraph-enhanced Multi-interest Learning for Multi-behavior Sequential Recommendation. *Expert Systems with Applications*, Article 124497.

Li, Guopeng, Song, Dan, Bai, Wei, Han, Kun, & Tharmarasa, Ratnasingham (2023). Consensus and Complementary Regularized Non-negative Matrix Factorization for Multi-view Image Clustering. *Information Sciences*, *623*, 524–538.

Li, Yiran, Yang, Renchi, & Shi, Jieming (2023). Efficient and Effective Attributed Hypergraph Clustering via K-nearest Neighbor Augmentation. *Proceedings of the ACM on Management of Data*, *1*(2), 1–23.

Li, Jinxing, Zhang, Bob, Lu, Guangming, & Zhang, David (2019). Generative Multi-view and Multi-feature Learning for Classification. *Information Fusion*, *45*, 215–226.

Lu, Jielong, Wu, Zhihao, Chen, Zhaoliang, Cai, Zhiling, & Wang, Shiping (2024). Towards Multi-view Consistent Graph Diffusion. In *Proceedings of the 32nd ACM International Conference on Multimedia* (pp. 186–195).

Lu, Jielong, Wu, Zhihao, Zhong, Luying, Chen, Zhaoliang, Zhao, Hong, & Wang, Shiping (2024). Generative Essential Graph Convolutional Network for Multi-view Semi-supervised Classification. *IEEE Transactions on Multimedia*, 1–13.

Luo, Yong, Tao, Dacheng, Geng, Bo, Xu, Chao, & Maybank, Stephen J (2012). Manifold Regularized Multitask Learning for Semi-supervised Multilabel Image Classification. *IEEE Transactions on Image Processing*, *22*(2), 523–536.

Lv, Xin, Wang, Le, Zhang, Qilin, Zheng, Nanning, & Hua, Gang (2018). Video Object Co-segmentation from Noisy Videos bya Multi-level Hypergraph Model. In *Proceedings of the IEEE International Conference on Image Processing* (pp. 2207–2211).

Shervashidze, Nino, Schweitzer, Pascal, van Leeuwen, Erik Jan, Mehlhorn, Kurt, & Borgwardt, Karsten M (2011). Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, *12*, 2539–2561.

Sun, Yinghui, Gao, Xizhan, Niu, Sijie, Wei, Dong, & Cui, Zhen (2023). Two-directional Two-dimensional Fractional-order Embedding Canonical Correlation Analysis for Multi-view Dimensionality Reduction and Set-based Video Recognition. *Expert Systems with Applications*, *214*, Article 119062.

Sun, Liang, Ji, Shuiwang, & Ye, Jieping (2008). Hypergraph Spectral Learning for Multi-label Classification. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 668–676).

Tao, Hong, Hou, Chenping, Nie, Feiping, Zhu, Jubo, & Yi, Dongyun (2017). Scalable Multi-view Semi-supervised Classification via Adaptive Regression. *IEEE Transactions on Image Processing*, *26*(9), 4283–4296.

Tao, Hong, Hou, Chenping, Yi, Dongyun, & Zhu, Jubo (2018). Multiview Classification with Cohesion and Diversity. *IEEE Transactions on Cybernetics*, *50*(5), 2124–2137.

Tu, Ke, Cui, Peng, Wang, Xiao, Wang, Fei, & Zhu, Wenwu (2018). Structural Deep Embedding for Hyper-networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 426–433).

Vicente García, Ramón Alberto Mollineda, & Sánchez, José Salvador (2008). On the K-NN Performance in a Challenging Scenario of Imbalance and Overlapping. *Pattern Analysis and Applications*, *11*(3–4), 269–280.

Wang, Shiping, Chen, Zhaoliang, Du, Shide, & Lin, Zhouchen (2022). Learning Deep Sparse Regularizers with Applications to Multi-view Clustering and Semi-supervised Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5042–5055.

Wang, Xiaoli, Wang, Yongli, Ke, Guanzhou, Wang, Yupeng, & Hong, Xiaobin (2024). Knowledge Distillation-driven Semi-supervised Multi-view Classification. *Information Fusion*, *103*, Article 102098.

Wang, Hao, Yang, Yan, & Liu, Bing (2019). GMC: Graph-based Multi-view Clustering. *IEEE Transactions on Knowledge and Data Engineering*, *32*(6), 1116–1129.

Wang, Xiao, Zhu, Meiqi, Bo, Deyu, Cui, Peng, Shi, Chuan, & Pei, Jian (2020). AM-GCN: Adaptive Multi-channel Graph Convolutional Networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1243–1253).

Wu, Xiangping, Chen, Qingcai, Li, Wei, Xiao, Yulun, & Hu, Baotian (2020). AdaHGNN: Adaptive Hypergraph Neural Networks for Multi-label Image Classification. In *Proceedings of the ACM International Conference on Multimedia* (pp. 284–293).

Xie, Yu, Zhang, Yuanqiao, Gong, Maoguo, Tang, Zedong, & Han, Chao (2020). MGAT: Multi-view Graph Attention Networks. *Neural Networks*, *132*, 180–189.

Zass, Ron, & Shashua, Amnon (2008). Probabilistic Graph and Hypergraph Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–8).

Zhang, Kaiwu, Du, Shiqiang, Wang, Yaoying, & Deng, Tao (2024). Deep Incomplete Multi-view Clustering via Attention-based Direct Contrastive Learning. *Expert Systems with Applications*, *255*, Article 124745.

Zhang, Bin, Qiang, Qianyao, Wang, Fei, & Nie, Feiping (2021). Flexible Multi-view Unsupervised Graph Embedding. *IEEE Transactions on Image Processing*, *30*, 4143–4156.

Zhang, Ruochi, Zou, Yuesong, & Ma, Jian (2019). Hyper-SAGNN: A Self-attention Based Graph Neural Network for Hypergraphs. In *Proceedings of the International Conference on Learning Representations* (pp. 1–7).

Zheng, Qinghai, Zhu, Jihua, Ma, Yuanyuan, Li, Zhongyu, & Tian, Zhiqiang (2021). Multi-view Subspace Clustering Networks with Local and Global Graph Information. *Neurocomputing*, *449*, 15–23.

Zhong, Guo, Shu, Ting, Huang, Guoheng, & Yan, Xueming (2022). Multi-view Spectral Clustering by Simultaneous Consensus Graph Learning and Discretization. *Knowledge-Based Systems*, *235*, Article 107632.

Zhou, Yiyang, Zheng, Qinghai, Wang, Yifei, Yan, Wenbiao, Shi, Pengcheng, & Zhu, Jihua (2024). MCoCo: Multi-level Consistency Collaborative Multi-view Clustering. *Expert Systems with Applications*, *238*, Article 121976.