

## Journal Pre-proof

Joint learning of feature and topology for multi-view graph convolutional network

Yuhong Chen, Zhihao Wu, Zhaoliang Chen, Mianxiong Dong,  
Shiping Wang



PII: S0893-6080(23)00498-7  
DOI: <https://doi.org/10.1016/j.neunet.2023.09.006>  
Reference: NN 5878

To appear in: *Neural Networks*

Received date: 18 May 2023  
Revised date: 15 July 2023  
Accepted date: 3 September 2023

Please cite this article as: Y. Chen, Z. Wu, Z. Chen et al., Joint learning of feature and topology for multi-view graph convolutional network. *Neural Networks* (2023), doi: <https://doi.org/10.1016/j.neunet.2023.09.006>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2023 Elsevier Ltd. All rights reserved.

# Joint Learning of Feature and Topology for Multi-view Graph Convolutional Network

Yuhong Chen<sup>a,b</sup>, Zhihao Wu<sup>a,b</sup>, Zhaoliang Chen<sup>a,b</sup>, Mianxiong Dong<sup>c</sup>, Shiping Wang<sup>a,b,\*</sup>

<sup>a</sup>College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China

<sup>b</sup>Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China

<sup>c</sup>Department of Sciences and Informatics, Muroran Institute of Technology, Muroran 050-8585, Japan

---

## Abstract

Graph convolutional network has been extensively employed in semi-supervised classification tasks. Although some studies have attempted to leverage graph convolutional networks to explore multi-view data, they mostly consider the fusion of feature and topology individually, leading to the underutilization of the consistency and complementarity of multi-view data. In this paper, we propose an end-to-end joint fusion framework that aims to simultaneously conduct a consistent feature integration and an adaptive topology adjustment. Specifically, to capture the feature consistency, we construct a deep matrix decomposition module, which maps data from different views onto a feature space obtaining a consistent feature representation. Moreover, we design a more flexible graph convolution that allows to adaptively learn a more robust topology. A dynamic topology can greatly reduce the influence of unreliable information, which acquires a more adaptive representation. As a result, our method jointly designs an effective feature fusion module and a topology adjustment module, and lets these two modules mutually enhance each other. It takes full advantage of the consistency and complementarity to better capture the more intrinsic information. The experimental results indicate that our method surpasses state-of-the-art semi-supervised classification methods.

**Keywords:** Multi-view learning, semi-supervised classification, graph convolution network, feature and topology fusion.

---

\*Corresponding author.

*Email addresses:* yhchen2320@163.com (Yuhong Chen), zhihaowu1999@gmail.com (Zhihao Wu), chenzl23@outlook.com (Zhaoliang Chen), mx.dong@csse.muroran-it.ac.jp (Mianxiong Dong), shipingwangphd@163.com (Shiping Wang)

## 1. Introduction

Recently, multi-view data has become an important data source in numerous fields due to the rich information it provides. The data from various perspectives not only complements the missing information of each other but also represents the consistency property of one object. As the information captured in multi-view data is more complete, multi-view learning has been widely applied to the fields of clustering [1, 2, 3, 4], discriminant analysis [5, 6, 7] and computer vision [8, 9, 10, 11].

As multiple connective relations between entities can be described well by graphs, multi-view data can be naturally represented as graph structures. By abstracting different types of data into graph structures, graph learning makes it possible to use the same machine learning framework to solve different types of problems. Graph learning has achieved remarkable results in numerous domains like node classification [12, 13, 14], graph clustering [15, 16, 17], link prediction [18, 19, 20] and fashion recommendation [21, 22]. Graph Convolutional Network (GCN), which has the ability to handle non-Euclidean data, is an effective approach to graph learning [23]. Due to its efficient layer-wise propagation rule, GCN can aggregate feature information from the topological structure.

There have been many attempts to utilize GCN for learning a consistent representation from multi-view data. Some approaches [24, 25, 26] try to learn independently from multiple perspectives to obtain view-specific representations, which inevitably carry redundant information. In other words, as the previously mentioned methods focus on the explicit fusion of multi-view data, they assume that each view carries equally important information, which does not consider the consistency of multi-view data well enough.

Since multi-view data lacks a natural graph structure, most methods determine the similarity of node geometric relations to construct a topology structure, which is probably unreliable. The topology is constructed with fixed connective relations built by each view data, and it is difficult to filter out the influence of unreliable information [25, 27, 28]. Therefore, some research [29, 30, 31] pointed out that it is necessary to further adjust the generated topology structure to obtain more comprehensive topological information. Regarding the differences among multi-view data descriptions, this solution which considers only the nearest neighbor search, is not comprehensive. It often happens in multi-view scenarios that two objects do not belong to the same class, but they have relatively similar features in one view. Filtering the undesirable connective relations provided by particular views can improve the performance of classification tasks. Therefore, how to make the best use of these connective relations is a point that the topology adjustment module

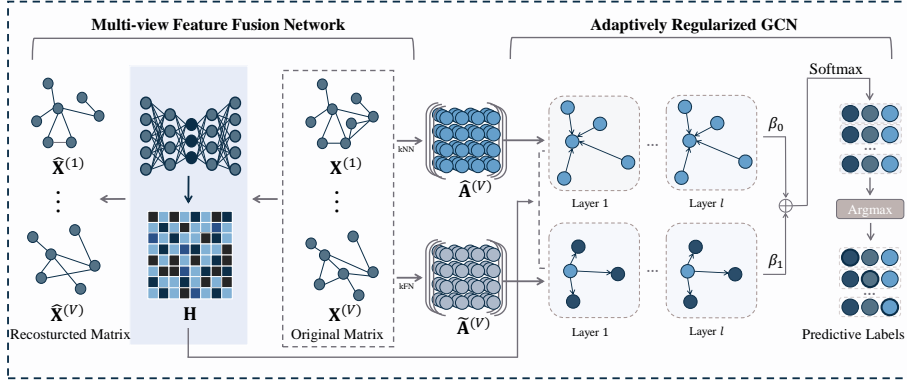
should be considered properly.

The methods mentioned above only focused on feature fusion and topology adjustment separately and did not consider the joint learning of feature fusion and topology adjustment, leading to inconsistent information that is transmitted. To solve the problems mentioned above, we propose an end-to-end framework named Joint Fusion Graph Convolutional Network (JFGCN). For feature fusion, we design a deep neural network to approximate matrix decomposition and finally gain a consistent feature representation. For topology adjustment, after analyzing the graph Laplacian regularization, we put forward an adaptive convolution operation to flexibly generate a comprehensive topology by extracting the connective relations from multiple views. Specifically, we construct an adaptive topological structure on  $k$ -Nearest Neighbor ( $k$ NN) and  $k$ -Farthest Neighbor ( $k$ FN) to keep similar nodes close and different nodes far away. JFGCN exploits the supervisory information between perspectives to dynamically generate a more accurate feature representation and a more adaptive topology structure, fully capturing the consistency and complementarity of multi-view data. The complete representations require a coordinated operation of feature fusion and topology adjustment. Thus, JFGCN is designed as a joint framework for deeply extracting the consistency and complementarity of multi-view data. Fig. 1 provides a description of the proposed JFGCN. The major contributions of this paper are summarized as listed below:

- Propose an end-to-end neural network for multi-view semi-supervised classification to jointly fuse feature information and adjust topology, allowing them mutually reinforce each other.
- Design an implicit feature fusion to mine the underlying consistent information and explore a more robust topology that adaptively fuses the adjacency matrices generated by  $k$ NN and  $k$ FN.
- The proposed framework is applied to conduct multi-view semi-supervised classification tasks, showcasing superior performance compared with other state-of-the-art graph-based algorithms.

The following segments are arranged as below. In Section 2, we present a brief overview of related work on graph convolutional networks and multi-view learning. The presented method is described in detail in Section 3. In Section 4, we evaluate the performance of the proposed method on six real-world datasets for semi-supervised classification. Finally, Section 5 is the conclusion of this paper.





**Figure 1:** An overview of the proposed framework JFGCN. We design a multi-view autoencoder to approximate matrix decomposition, which integrates the consistency of multi-view data. Simultaneously, the  $k$ -Nearest Neighbor ( $k$ NN) and  $k$ -Farthest Neighbor ( $k$ FN) strategies are utilized to calculate a more accurate set of topology matrices from two perspectives. Then JFGCN dynamically adjust it by using a flexible graph convolution to learn a robust connective pattern.

## 2. Related Work

In this section, we review some previous efforts that are pertinent to our investigation, containing graph convolutional network and multi-view learning.

### 2.1. Graph Convolutional Network

Kipf *et al.* [23] first presented Graph Convolutional Network (GCN), which could extract node features from non-Euclidean space and applied it to the task of classifying nodes in a network. GCN originated from the spectral convolution on the graph, which was performed by a signal  $x \in \mathbf{R}^m$  and the filter  $g_\theta = \text{diag}(\theta)$ , expressed as

$$g_\theta \star x = \mathbf{U}g_\theta\mathbf{U}^T x, \quad (1)$$

where  $\mathbf{U}$  is the eigenvector matrix of the normalized graph Laplacian  $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ , with a diagonal matrix constructed from its eigenvalues  $\mathbf{\Lambda}$  and  $\mathbf{U}^T x$  being the graph Fourier transform of  $x$ .

GCN performed a first-order approximation of the truncated Chebyshev polynomial and designed a propagation rule which can be formulated as

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l+1)}), \quad (2)$$

where  $\mathbf{H}^{(l)}$  and  $\mathbf{H}^{(l+1)}$  are the input and output of the  $l$ -th graph convolution layer. Here,  $\sigma(\cdot)$  is an activation function,  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$  is an adjacency matrix with self-loop, and  $\tilde{\mathbf{D}}$  is denoted as a degree matrix.

GCN has the ability to aggregate the neighbor features of nodes and learn node representations by weighted aggregation. Thus it has a significant advantage that encouraging performance can be obtained with a very small amount of labeled data, which allows it to be used in a broad spectrum of research.

Pang et al. [32] adopted an adaptive quadratic frequency response function and designed a new strategy to improve flexibility and interpretability through theoretical analysis in a spatial domain. Wang et al. [24] introduced an adjustable multi-channel graph convolutional network, extracting representations from node features, topology and their combinations using an attention mechanism to acquire the adaptive importance weights of the representation. Li et al. [25] presented an approach for the adaptive integration of multi-view feature information, which employed GCN to generate representations for each view and finally aggregated them. Feng et al. [33] designed a new graph convolution operator called cross-feature graph convolution, which has a complexity that is linear to the feature dimension. Chen et al. [34] proposed a method name LGCN, which utilized sparse autoencoders and fully connected networks to fuse features from different views. It aims to study the unique underlying representations from all view features, eventually fused them into one representation. Distinct from the aforementioned papers, we design a multi-view autoencoder to capture the consistency of multi-view data simultaneously. In addition, instead of designing multiple GCNs to learn multiple representations, and adaptively learning the weights of a single topology, we explore one representation of the weights of different topologies in the GCN.

In our explorations we have found that the majority of existing GCN-based methods focus on single-view data in an attempt to better tune the effect of feature learning. The remaining small number of approaches mostly focuses on explicit feature fusion and do not take full advantage of the consistency and complementarity of multi-view data. There is still a long portion of GCN-based multi-view learning that can be explored.

## 2.2. Multi-view Learning

Multi-view learning is trained in scenarios that are comprised of multiple data sources, extracting consistent and complementary information from multiple views while reducing the interference of redundant information. In practical applications, one object can be described from different views, and these various descriptions constitute multiple views of this object. Besides, different views often contain view-specific information that can complement each other to obtain a more complete description of the object. The intrinsic representation of

multi-view data can be better learned through the deep mining of consistent and complementary information.

Multi-view learning is desired to leverage data from different sources to improve the holistic effect and eventually achieves knowledge transfer among different views. Nie et al. [34] proposed a brand new method by reconstructing the standard spectral learning model, which was capable of adaptively learning the weight of each view and exploring a matrix of tokens by aggregating the loss functions of multiple views. Yang et al. [30] developed a topology fusion method that aims to obtain a complete topology by fusing the similarity relations learned from different views and designed a learnable weight parameter to aggregate the useful topological information. Jia et al. [35] proposed a multi-view deep discriminative representation learning method to reduce the redundancy of the learned representation by combining orthogonality and adversarial similarity constraints. Wu et al. [36] suggested an end-to-end model, which constructed a flexible graph filter and introduced orthogonal normalization to enhance the interpretability of neural networks. Yao et al. [37] posited a multi-view graph convolutional network with an attentional mechanism, by combining the topology of multiple views and an attention-based feature aggregation strategy. Methods on multi-view learning attempt to explore the consistency and complementarity of multi-view data, capturing a more comprehensive representation. The above methods attached more importance to consistent information embodied in multi-view data by designing various constraints to learn a representation. In contrast to the above mentioned papers, our method captures the consistency of multi-view data by designing a multi-view autoencoder, and exploits the complementarity of multi-view data by adaptively learning the weights between multiple topologies via  $k$ NN and  $k$ FN.

Existing GCN-based multi-view learning focuses on several modules such as explicit representation fusion and topology adjustment. Most of the methods attempt to mine the consistency information of multi-view data in various ways and also achieve encouraging performance. However, these methods neglect the utilization of supervised information between views. The proposed method attempts to explore an implicit feature fusion and topological adjustment to fully exploit the consistency and complementarity of multi-view data.

### 3. The Proposed Method

In this section, to jointly extract the feature and topology information from multi-view data, we present the JFGCN model for exploring the underlying representation. JFGCN consists of two main modules: A feature fusion network and an adaptively regularized GCN. The feature fusion module is designed to integrate features from multiple views into a consistent low-dimensional representation, so as to deeply explore the shared information. For the adaptively regularized GCN

**Table 1:** The symbolic notations and their descriptions.

Notations	Descriptions
$\mathbf{X}^{(v)} \in \mathbb{R}^{n \times d_v}$	The $v$ -th view feature matrix.
$\hat{\mathbf{X}}^{(v)} \in \mathbb{R}^{n \times d_v}$	The $v$ -th view reconstructed feature matrix.
$\mathbf{Z}^{(l)} \in \mathbb{R}^{n \times c}$	The latent consistent representation of the $l$ -th layer.
$\mathbf{H}_i \in \mathbb{R}^{n \times m}$	The feature representation matrix of the $i$ -th layer.
$\mathbf{A}_{pos}^{(v)} \in \mathbb{R}^{n \times n}$	The similarity matrix constructed from $\mathbf{X}^{(v)}$ by $k$ NN.
$\mathbf{A}_{neg}^{(v)} \in \mathbb{R}^{n \times n}$	The similarity matrix constructed from $\mathbf{X}^{(v)}$ by $k$ FN.

module, we revisit GCN to explore more accurate topological relations among views based on the graph Laplacian regularization. To be specific, we design a GCN with an adaptive topology structure, which employs  $k$ NN to learn the common features between similar nodes, and  $k$ FN to distinguish the difference among nodes far from each other. By simultaneously training these two modules, we take full advantage of the consistency and complementarity of multi-view data and finally achieve promising results.

To better understand the mathematical notation usages which are primarily used in this paper, we list the explanations of these notations in Table 1.

### 3.1. Multi-view Feature Fusion Network

Owing to the underutilization of consistency of multi-view data which results in incomprehensive feature representation, the proposed JFGCN attempts to design a feature fusion method, which can acquire more comprehensive representations. We regard multi-view data as a set of feature data obtained by different mappings of an essential feature representation, which contains the consistency and complementarity of the data. Considering that the consistent information among views needs to be emphasized, there is a strong desire to obtain high-quality feature extraction by exploiting mutual supervision between views. The deep matrix decomposition is presented with the goal of improving the feature extraction capabilities, which distills more vital features. The feature extraction operations approximate the decomposition of a matrix to the product of two matrices with lower rank. When performing matrix decomposition on the view-specific features, we can obtain filtered feature matrices, which express the information more obviously.

However, using matrix decomposition for each view separately cannot adequately utilize the strengths of the consistent information of multi-view data.

Thus, matrix decomposition of the multi-view feature matrix maps the samples in the feature space of different views onto a shared space and extracts the most important feature. For more flexible computation, we design a multi-view autoencoder network to approximate matrix decomposition, so that a unified shared representation of all views can be derived to offer better access to the underlying information. By implicitly fusing feature information, JFGCN filters some unreliable data, obtaining a more intrinsic representation.

An autoencoder can be formulated as  $\hat{\mathbf{X}} = g(f(\mathbf{X}))$ , where  $f(\cdot)$  is an encoder, and  $g(\cdot)$  is a decoder. We denote  $\mathbf{H}$  and  $\mathbf{P}$  as the weight matrices of the encoder and decoder, which can be represented as  $\hat{\mathbf{X}} = \mathbf{XHP}^T$ . If the input is replaced by an identity matrix, the expression can be written as  $\hat{\mathbf{X}} = \mathbf{IHP}^T = \mathbf{HP}^T$ , which can be considered as matrix decomposition. Furthermore, to solve the matrix decomposition of multiple views, we use a shared encoder and  $V$  decoders, where the output of the shared encoder is a consistent feature representation of the multi-view data, and the output of each decoder corresponds to a reconstructed matrix  $\hat{\mathbf{X}}$  of the view data.

Specifically, the output of the  $i$ -th layer in the encoder is

$$\mathbf{H}_{i-1} = \sigma(\mathbf{H}_i \mathbf{P}_i^T), \quad (3)$$

where  $\sigma(\cdot)$  is a specific activation function used to guarantee some constraint on matrix decomposition. We denote the last layer of the encoder as  $\mathbf{H}$ , which can be regarded as the consolidated representation of all views. By using multiple projection matrices, we map data from different original spaces onto a unified representation matrix, which contains a view-shared representation. In a nutshell, JFGCN acquires a low-dimensional shared representation  $\mathbf{H}$  by an implicit matrix multiplication operation to reduce the dimensionality in the decoder part, and maps it onto the next layer in the encoder, until a new view-specific  $\hat{\mathbf{X}}^{(v)}$  is obtained, which can be expressed as

$$\hat{\mathbf{X}}^{(v)} = \sigma(\sigma(\mathbf{HP}_l^{(v)}) \cdots \mathbf{P}_2^{(v)}) \mathbf{P}_1^{(v)}, \quad (4)$$

where  $\hat{\mathbf{X}}^{(v)}$  is the output of the  $v$ -th decoder, denoted as the reconstructed data matrix of the  $v$ -th view.  $\mathbf{H}$  is the final output of the encoder. Eventually, the loss is minimized to filter out the view-specific information to obtain a consistent low-dimensional representation that incorporates multi-view consensus features.

Because the constructed low-dimensional representation  $\mathbf{H}$  is expected to be extracted as accurate features of each view as possible, we adopt a loss function to evaluate the distance between original features and the reconstructed representations, i.e.,

$$\mathcal{L}_{RE} = \frac{1}{2} \sum_{v=1}^V \|\mathbf{X}^{(v)} - \hat{\mathbf{X}}^{(v)}\|_F^2. \quad (5)$$

### 3.2. Adaptively Regularized GCN

Graph regularization has been widely studied as a graph-based nonlinear dimensionality reduction method. We first introduce the multi-view graph regularization problem

$$\min_{\mathbf{z}_i} \sum_{i=1, \dots, n} \sum_{v=1}^V \alpha^{(v)} \mathbf{A}_{ij}^{(v)} \sum_{i,j=1}^n \left\| \frac{\mathbf{z}_i}{\sqrt{\mathbf{D}_{ii}^{(v)}}} - \frac{\mathbf{z}_j}{\sqrt{\mathbf{D}_{jj}^{(v)}}} \right\|_2^2, \quad (6)$$

where  $\mathbf{z}_i$  is the  $i$ -th row vector of the learned representation  $\mathbf{Z}$ . Here,  $\mathbf{A}_{ij}^{(v)}$  is the  $[i, j]$ -th element of the adjacency matrix  $\mathbf{A}^{(v)}$ , and  $\alpha^{(v)}$  is the weight of  $\mathbf{A}^{(v)}$ . Problem (6) aims to generate a low-dimensional representation that preserves the potential structure of the original data. To filter unnecessary connective relations, we jointly consider two kinds of connective relations and design a new topology structure, which applies  $k$ -Nearest Neighbor ( $k$ NN) and  $k$ -Farthest Neighbor ( $k$ FN) simultaneously.  $k$ NN gathers the  $k$  neighbors closest to the sample, while  $k$ FN collects the  $k$  neighbors farthest from the sample.

If two samples  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are connected, they should be close in the low-dimensional space. We utilize  $k$ NN to design positive connective relations named  $\mathbf{A}_{pos}^{(v)}$ , as follows

$$\mathbf{A}_{pos}^{(v)} = \begin{cases} 1, & \mathbf{x}_i^{(v)} \in k\text{NN}(\mathbf{x}_j^{(v)}) \text{ or } \mathbf{x}_j^{(v)} \in k\text{NN}(\mathbf{x}_i^{(v)}), \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

However, it is not perfectly reasonable as each edge in each view contributes equally to the objective function in the multi-view case. For example, two samples that do not belong to the same class have similar features in a certain view, and this misinformation may be exploited and propagated, which degrades the performance of the classification task. This information is inevitably carried in multi-view data, but the information carried by the nodes that are farther away at this point is more credible. This type of connection may encode the dissimilarity between samples, called negative edges, which means that the corresponding samples are expected to be far away in the latent space.

Therefore, we apply  $k$ FN to construct adjacency matrices with negative edges to widen the distance of dissimilar nodes and design negative connective relations named  $\mathbf{A}_{neg}^{(v)}$ , as follows

$$\mathbf{A}_{neg}^{(v)} = \begin{cases} -1, & \mathbf{x}_i^{(v)} \in k\text{FN}(\mathbf{x}_j^{(v)}) \text{ or } \mathbf{x}_j^{(v)} \in k\text{FN}(\mathbf{x}_i^{(v)}), \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Finally, we define an adaptive fusion rule of adjacency matrices as

$$\mathbf{A}^{(v)} = \beta_1 \mathbf{A}_{pos}^{(v)} + \beta_2 \mathbf{A}_{neg}^{(v)}, \quad (9)$$

where  $\beta_1 + \beta_2 = 1$ .  $\mathbf{A}_{pos}^{(v)}$  is the positive connective relations that can narrow the similar nodes, and  $\mathbf{A}_{neg}^{(v)}$  is the negative connective relations that are used to pull away the different nodes. Please note that the final generated  $\mathbf{A}^{(v)}$  is designed to be normalized. Therefore,  $\mathbf{D}^{(v)}$  is guaranteed to be positive.

For a brief formulation, we temporarily ignore the weighted sum, and only use the view-specific form

$$\min_{\mathbf{Z}} \sum_{i,j=1}^n \mathbf{A}_{ij}^{(v)} \left\| \frac{\mathbf{z}_i}{\sqrt{\mathbf{D}_{ii}^{(v)}}} - \frac{\mathbf{z}_j}{\sqrt{\mathbf{D}_{jj}^{(v)}}} \right\|_2^2, \quad (10)$$

Equation (10) can be transformed into the following form

$$\min_{\mathbf{Z}} \text{Tr}(\mathbf{Z}^\top \hat{\mathbf{L}}^{(v)} \mathbf{Z}), \quad (11)$$

where  $\hat{\mathbf{L}}^{(v)} = \mathbf{I} - (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \hat{\mathbf{A}}^{(v)} (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}}$  is the Laplacian matrix using the symmetric normalization with  $\hat{\mathbf{A}}^{(v)} = \mathbf{I} + \mathbf{A}^{(v)}$  and  $\hat{\mathbf{D}}^{(v)}$  is a degree matrix of  $v$ -th view data.

To tackle the problem (11), we denote  $\mathcal{L} = \text{Tr}(\mathbf{Z}^\top \hat{\mathbf{L}}^{(v)} \mathbf{Z})$  and then calculate its derivative w.r.t.  $\mathbf{Z}$ , that is,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 2 \left( \mathbf{I} - (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \hat{\mathbf{A}}^{(v)} (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \right) \mathbf{Z}. \quad (12)$$

Letting Equation (12) be equal to zero, it leads to

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 0 \Rightarrow \mathbf{Z}^* = (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \hat{\mathbf{A}}^{(v)} (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \mathbf{Z}. \quad (13)$$

The above Equation (13) can be explained as a limit distribution where  $\mathbf{Z}_{lim} = (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \hat{\mathbf{A}}^{(v)} (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \mathbf{Z}_{lim}$ . Then we have the following iterative form to approximate the limit of  $\mathbf{Z}$ :

$$\mathbf{z}_i^{(l+1)} = \frac{1}{\hat{\mathbf{D}}_{ii}^{(v)}} \mathbf{z}_i^{(l)} + \sum_{j \neq i} \frac{\mathbf{A}_{ij}^{(v)}}{\sqrt{\hat{\mathbf{D}}_{ii}^{(v)} \hat{\mathbf{D}}_{jj}^{(v)}}} \mathbf{z}_j^{(l)}, \quad (14)$$

which is the forward computation of the commonly used GCN. And we rewrite it as the Equation (9),

$$\begin{aligned} \mathbf{Z}^{(l+1)} &= (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} (\mathbf{I} + \mathbf{A}^{(v)}) (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \mathbf{Z}^{(l)} \\ &= (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} (\mathbf{I} + \beta_1 \mathbf{A}_{pos}^{(v)} + \beta_2 \mathbf{A}_{neg}^{(v)}) (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \mathbf{Z}^{(l)} \\ &= (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} [\beta_1 (\mathbf{I} + \mathbf{A}_{pos}^{(v)}) + \beta_2 (\mathbf{I} + \mathbf{A}_{neg}^{(v)})] (\hat{\mathbf{D}}^{(v)})^{-\frac{1}{2}} \mathbf{Z}^{(l)}. \end{aligned} \quad (15)$$

Thus, we define  $(\mathbf{I} + \mathbf{A}_{pos}^{(v)})$  as the positive graph-regularized convolution, while  $(\mathbf{I} + \mathbf{A}_{neg}^{(v)})$  is a negative graph-regularized convolution, which is an operation to stress the self features and promotes the connected samples to be dissimilar. In a weighted way, the two connective relations supervise each other to generate more accurate topology information.

Utilizing this graph convolution to construct the network, we obtain the following propagation rule

$$\mathbf{Z}^{(l+1)} = \sigma \left( \sum_{v=1}^V \alpha^{(v)} \mathbf{A}^{(v)} \mathbf{Z}^{(l)} \mathbf{W}^{(l+1)} \right), \quad (16)$$

where  $\mathbf{A}^{(v)} = \beta_1^{(v)} \mathbf{A}_{pos}^{(v)} + \beta_2^{(v)} \mathbf{A}_{neg}^{(v)}$  and the initial input  $\mathbf{Z}^{(0)} = \mathbf{H}$ . With this series of adaptive operations on edges, the relationship between pairs of nodes in graph topological space is adaptively adjusted to efficiently extract more accurate topological information. The loss of JFGCN is evaluated with the cross-entropy error over all labeled samples, as follows

$$\mathcal{L}_{CE} = - \sum_{i \in \Omega} \sum_{j=1}^c y_{ij} \ln \mathbf{Z}_{ij}^{(K)}, \quad (17)$$

where  $\Omega$  refers to the set of labeled samples,  $c$  is the amount of classes, and  $K$  is the layer in JFGCN.

We aggregate the above loss functions to train the proposed method. The loss function for the whole networks is shown below

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{RE}. \quad (18)$$

JFGCN continuously trains the loss while approaching better feature representations and a more accurate topology structure. We conclude the training procedure of the proposed method in Algorithm 1, and return the predictive representation.

#### 4. Experiments

In this section, we performed the experimental settings, detailing the selected data sets and comparison methods. All compared methods are tested on six mainstream datasets to evaluate the evaluation metrics. In addition, experiments such as ablation studies and parameter sensitivity analysis are performed to prove the effectiveness and feasibility of JFGCN for semi-supervised node classification tasks.



**Algorithm 1** Joint Fusion Graph Convolutional Network (JFGCN)

**Input:** Multi-view data  $\{\mathbf{X}_v\}_{v=1}^V$ , label matrix  $\mathbf{Y} \in \mathbb{R}^{|\Omega| \times c}$ , hyperparameter  $k$  in  $k$ NN and  $k$ FN.

**Output:** Predictive labels.

Initialize weights and biases of autoencoders;

Calculate  $\mathbf{A}_{pos}^{(v)}$  and  $\mathbf{A}_{neg}^{(v)}$  from  $\mathbf{X}_v$  for each view through  $k$ NN and  $k$ FN.

**while** not convergent **do**

**for**  $i = 1 \rightarrow l$  **do**

    Compute corresponding representation  $\mathbf{H}_i$  with Equation (3);

**end for**

  Calculate  $\mathbf{A}^{(v)}$  through  $\mathbf{A}_{pos}^{(v)}$  and  $\mathbf{A}_{neg}^{(v)}$  by Equation (9);

  Update  $\mathbf{Z}$  with forward propagation with Equation (16);

  Compute the loss  $\mathcal{L}$  by Equation (18);

  Optimize the trainable parameters from network by backward propagation;

**end while**

Obtain the predictive output  $\mathbf{Z}$ ;

Generate labels based on the predictive output  $\mathbf{Z}$ ;

**return** Predictive labels.

#### 4.1. Experimental Settings

In this subsection, the proposed method JFGCN is compared with eight methods on six benchmark datasets. We detail the sources of the six datasets and briefly describe the parameters used for each compared method.

##### 4.1.1. Datasets

Six commonly available multi-view datasets are used to evaluate the performance, as listed below. Table 2 presents some details of these datasets, including the number of samples, features, views, and data types.

**Table 2:** A brief description of the test multi-view datasets.

Datasets	# Samples	# Views	# Features	# Classes	# Data Types
100leaves	1,600	3	64/64/64	100	Object images
ALOI	1,079	4	64/64/77/13	10	Object images
Animals	10,158	2	4,096/4,096	50	Animal images
Caltech102	9,144	6	48/40/254/1,984/512/928	102	Object images
MNIST	10,000	3	30/9/9	10	Digit images
OutScene	2,688	4	512/432/256/45	8	Object images

**100leaves**<sup>1</sup> contains 16 different plant leaves, each with 100 samples. Three view features are used, including shape description, edge and histogram features.

**ALOI**<sup>2</sup> is a set containing a large number of images of objects that were taken under different lighting illumination conditions and angles of rotation.

**Animals**<sup>3</sup> has the 30,475 animal photos organized into 50 categories. And two categories of features from these images were analyzed and extracted.

**Caltech102**<sup>4</sup> comprises 9,144 images distributed over 102 classes. Six extracted features are used: Gabor features, wavelet moments features, Centrist features, histogram of oriented gradients features, GIST features, and LBP features.

**MNIST**<sup>5</sup> is a handwritten digits dataset, which has three types features: projection, linear discriminant analysis and neighborhood preserving embedding.

**OutScene**<sup>6</sup> contains 2,688 images belonging to 8 groups. For each image, we extract GIST features, color moment features, HOG features, and LBP features.

#### 4.1.2. Compared Methods

To validate the proposed framework, we compare our method with several classical and state-of-the-art algorithms in multi-view semi-supervised classification tasks, including MVAR [38], WREG [39], HLR-M<sup>2</sup>VS [40], Co-GCN [25], ERL-MVSC [41], DSRL [42], LGCN-FF [43] and GCN-Fusion [23]. The descriptions of these methods and some detailed settings are provided below.

**MVAR** utilizes the  $l_{2,1}$  norm to evaluate the regression loss values for each individual view so that the weighted sum of all regression losses is used to construct the objective. We adjust the weight of each view to  $\lambda = 1,000$ , while the re-assignment parameter  $\gamma$  is fixed to 2.

**WREG** presents a supervised multi-view graph learning framework to process different perspectives with a uniform perceptual manner. Furthermore, it merges multi-view data through mapping raw features onto a discriminative low-dimensional subspace. With this strategy, both relevant and complementary information can be retained, which further enhances the discriminative capability for subspace classification.

**HLR-M<sup>2</sup>VS** generates a unified tensor space to jointly explore the relationships of multiple views through local geometric structures, where a low-rank tensor regularization is used to ensure that each view is consistently fused. In this method, the weight factors are set to  $\lambda_1 = 0.2$  and  $\lambda_2 = 0.4$ .

<sup>1</sup><https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set>

<sup>2</sup>[https://elki-project.github.io/datasets/multi\\_view](https://elki-project.github.io/datasets/multi_view)

<sup>3</sup><http://attributes.kyb.tuebingen.mpg.de/>

<sup>4</sup>[http://www.vision.caltech.edu/Image\\_Datasets/Caltech101/](http://www.vision.caltech.edu/Image_Datasets/Caltech101/)

<sup>5</sup><http://yann.lecun.com/exdb/mnist/>

<sup>6</sup><https://github.com/YuhongChen2320/multi-view-data>

**Co-GCN** introduces graph convolutional networks to multi-view learning and adaptively exploits graph information in multiple views by combining Laplace operators. A two-layer GCN is adopted and the learning rate is set to 0.001.

**ERL-MVSC** reformulates a linear regression model to generate view-specific representational regularizers and automatically identifies their weights, exploiting diversity, sparsity, and consensus information from multiple views.

**DSRL** proposes a deep sparse regularizer learning model that adaptively recovers a data-driven sparse regularizer. It constructs a neural network consisting of various blocks, each of which is differentiable and reusable.

**LGCN-FF** consists of two stages, where the previous one designed to train an underlying representation from heterogeneous views, and the latter one aims to explore a more discriminative graph fusion through learnable weights and parameterized activation functions.

**GCN-Fusion** is derived from the transformation of GCN. GCN is skilled at handling semi-supervised node classification tasks. Since the raw model cannot handle multi-view data directly, we compute the average adjacency matrix during the graph convolution, which is called GCN-Fusion. The learning rate is 0.01 and the convolution layer is set to 2.

Furthermore, we also detail several experimental metrics used in our experiments. The dimensionality of the potential representations of the graph autoencoder is set up as  $\{2,048, 1,024\}$ . The learning rate in the experiments is fixed as 0.001 and the weight decay rate is matched to 0.0005. All experiments are performed 5 times and we keep the mean and standard deviation as the final results. Eventually, two classification evaluation metrics, accuracy (ACC) and F1-score, are applied to the experiments.

## 4.2. Experimental Results

In this subsection, we perform semi-supervised classification experiments, parameter sensitivity analysis, and ablation study.

### 4.2.1. Semi-supervised Classification Performance

We conduct extensive experiments in multi-view semi-supervised classification tasks. The performance of all compared methods with 10% randomly labeled data is shown in Table 3. From the experimental results, the classification performance obtained from JFGCN is superior on the six datasets. Compared to the GCN-based method, the method showed more significant performance improvements on the 100leaves and OutScene datasets. This result shows that JFGCN has more powerful feature extraction and topology adjustment capabilities on the tested datasets.

Besides, we verify the effectiveness of our topology adjustment module. We plot the weight learning of  $\mathbf{A}_{neg}$  and  $\mathbf{A}_{pos}$  named  $\beta_1$  and  $\beta_2$ . The weights of two

**Table 3:** 10% labeled samples are used to compare the classification accuracy (mean % and standard deviation %) of all methods, where the best performance is highlighted in bold and the second-best results are underlined.

Datasets\Methods		MVAR	WREG	HLR-M <sup>2</sup> VS	Co-GCN	ERL-MVSC	DSRL	LGCN-FF	GCN-Fusion	JFGCN
100leaves	ACC	39.1 (2.4)	62.7 (5.9)	54.4 (2.2)	30.3 (5.0)	58.1 (0.1)	67.9 (1.6)	62.1 (0.6)	<u>76.3 (2.0)</u>	<b>92.1 (0.4)</b>
	Precision	38.9 (4.1)	58.3 (7.0)	49.3 (2.3)	34.8 (4.4)	56.3 (0.3)	58.9 (3.1)	64.7 (1.6)	<u>73.1 (3.6)</u>	<b>92.9 (0.3)</b>
	Recall	42.4 (2.3)	64.7 (5.4)	56.9 (2.1)	30.3 (5.0)	64.2 (2.0)	68.1 (2.4)	62.2 (0.7)	<u>77.2 (1.8)</u>	<b>92.1 (0.4)</b>
	F1-score	40.1 (3.1)	61.3 (6.3)	52.8 (2.1)	29.1 (4.9)	57.7 (0.1)	64.4 (1.8)	61.7 (0.5)	<u>73.5 (2.6)</u>	<b>91.9 (0.4)</b>
ALOI	ACC	55.9 (0.7)	90.2 (1.3)	89.7 (0.8)	85.9 (5.2)	87.9 (1.3)	90.9 (0.9)	<u>95.6 (0.3)</u>	91.9 (0.9)	<b>97.2 (1.7)</b>
	Precision	20.0 (11.0)	91.2 (1.1)	90.5 (0.8)	86.2 (7.1)	88.0 (1.2)	92.1 (0.5)	<u>95.9 (0.3)</u>	92.6 (0.6)	<b>97.4 (0.3)</b>
	Recall	31.0 (9.0)	90.3 (1.4)	89.8 (0.8)	86.1 (5.0)	88.8 (0.9)	91.0 (0.8)	<u>95.6 (0.3)</u>	92.0 (0.9)	<b>97.2 (0.3)</b>
	F1-score	49.7 (0.7)	90.7 (1.2)	90.1 (0.8)	85.1 (6.3)	88.4 (1.1)	91.1 (0.8)	<u>95.6 (0.3)</u>	92.1 (0.9)	<b>97.2 (1.7)</b>
Animals	ACC	81.5 (0.5)	<u>83.5 (0.3)</u>	72.7 (0.5)	80.2 (1.2)	70.1 (0.1)	78.7 (0.5)	<u>73.9 (0.3)</u>	64.8 (0.3)	<b>84.8 (0.1)</b>
	Precision	79.1 (1.8)	<u>80.5 (0.7)</u>	70.2 (0.9)	77.1 (1.7)	66.0 (0.1)	75.3 (1.5)	68.8 (0.1)	63.1 (1.0)	<b>81.6 (0.2)</b>
	Recall	74.5 (0.6)	<u>76.9 (0.8)</u>	66.0 (1.0)	73.2 (1.5)	64.8 (0.1)	71.7 (1.0)	65.6 (0.5)	60.6 (0.7)	<b>79.6 (0.1)</b>
	F1-score	76.7 (1.0)	<u>78.7 (0.5)</u>	68.1 (0.9)	73.7 (1.6)	65.4 (0.1)	71.9 (1.1)	66.1 (0.5)	60.9 (0.8)	<b>80.1 (0.1)</b>
Caltech102	ACC	46.1 (1.0)	46.9 (0.5)	48.1 (0.4)	38.0 (8.7)	46.3 (0.1)	<u>52.0 (0.4)</u>	36.9 (0.6)	40.8 (0.4)	<b>53.7 (0.6)</b>
	Precision	<u>40.4 (1.9)</u>	34.8 (2.6)	35.9 (0.7)	22.3 (6.5)	26.7 (1.2)	35.8 (1.2)	14.5 (1.7)	26.2 (0.4)	<b>39.6 (0.6)</b>
	Recall	22.7 (0.9)	24.2 (0.6)	27.6 (1.1)	21.2 (6.3)	<u>49.3 (1.9)</u>	33.6 (0.6)	13.2 (1.7)	24.9 (0.4)	<b>33.8 (0.9)</b>
	F1-score	29.0 (1.0)	28.6 (0.8)	31.2 (0.7)	21.0 (6.4)	27.6 (1.2)	<u>32.9 (0.7)</u>	12.5 (1.2)	24.5 (0.4)	<b>34.6 (0.9)</b>
MNIST	ACC	84.9 (0.1)	79.1 (0.5)	80.5 (0.1)	78.1 (4.0)	<u>91.7 (0.1)</u>	89.1 (0.5)	88.8 (0.1)	90.4 (0.9)	<b>93.8 (0.1)</b>
	Precision	81.3 (0.1)	77.4 (0.1)	85.8 (0.1)	76.6 (3.3)	91.6 (0.1)	<b>94.2 (0.2)</b>	87.2 (0.1)	89.4 (1.2)	<u>93.8 (0.1)</u>
	Recall	82.1 (0.1)	72.8 (0.1)	75.5 (0.1)	78.7 (3.1)	91.7 (0.1)	<b>94.1 (0.2)</b>	86.0 (0.1)	89.0 (0.9)	<u>93.6 (0.1)</u>
	F1-score	81.7 (0.1)	75.0 (0.1)	80.3 (0.1)	78.2 (2.1)	<u>91.6 (0.1)</u>	87.1 (0.8)	86.4 (0.1)	88.9 (1.0)	<b>93.6 (0.1)</b>
OutScene	ACC	46.1 (11.3)	57.6 (1.8)	73.3 (1.7)	70.1 (2.1)	70.8 (0.1)	71.0 (2.0)	67.2 (0.2)	<u>76.5 (1.0)</u>	<b>81.4 (0.6)</b>
	Precision	59.5 (8.9)	59.2 (1.6)	76.8 (0.5)	72.0 (2.0)	71.4 (0.1)	75.4 (0.4)	69.0 (0.1)	<u>78.3 (1.0)</u>	<b>81.7 (0.5)</b>
	Recall	45.0 (11.6)	58.0 (1.7)	73.8 (1.8)	71.3 (1.9)	71.1 (0.1)	71.1 (2.5)	67.3 (0.2)	<u>76.8 (1.0)</u>	<b>81.1 (0.6)</b>
	F1-score	50.1 (11.1)	58.6 (1.6)	<u>75.2 (1.2)</u>	71.3 (2.0)	71.3 (0.1)	71.0 (2.3)	67.5 (0.3)	74.1 (1.3)	<b>77.0 (1.0)</b>

adjacency matrices at loss convergence are depicted in Table 4. The results show that different datasets often have different weight distributions. As we can see that datasets with a small number of categories generally prefer to use more positive connective relations, while datasets with a large number of classes favor reverse connectivity information. This may be attributed to the difficulty of capturing positive connective relations in situations when the number of categories is a bit too much. Positive relations are more prone to errors than negative relations.

Furthermore, we test the accuracy of all compared algorithms at different ratios on all six datasets. Figs. 3 and 4 demonstrate the performance of the proposed JFGCN with ratios ranging from 5% to 50%. JFGCN acquires promising performance at small ratios, while other comparison algorithms require higher ratios to reach comparable accuracy. Overall, JFGCN is more consistent with the intent of the semi-supervised node classification.

For a more visual demonstration, the t-SNE visualization of node representations derived for MNIST is shown in Fig. 2. We first concatenate feature vectors

**Table 4:** The presentation of the weight parameters  $\beta_1$  and  $\beta_2$  after training.

Datasets	Weight parameter		ACC	F1-score
	$\beta_1$	$\beta_2$		
100leaves	0.21 (0.01)	0.79 (0.01)	92.1 (0.4)	91.9 (0.4)
ALOI	0.94 (0.05)	0.06 (0.05)	97.2 (1.7)	97.2 (1.7)
Animals	0.08 (0.03)	0.92 (0.03)	84.8 (0.1)	80.1 (0.1)
Caltech102	0.05 (0.01)	0.95 (0.01)	53.7 (0.6)	34.6 (0.9)
MNIST	0.77 (0.04)	0.23 (0.04)	93.8 (0.1)	93.6 (0.1)
OutScene	0.99 (0.01)	0.01 (0.01)	81.4 (0.6)	81.2 (0.6)

**Table 5:** Ablation study of the proposed JFGCN on all test datasets.

Datasets	JFGCN w/o $\mathbf{A}_{neg}^{(v)}$		JFGCN w/o $\mathbf{A}_{pos}^{(v)}$		JFGCN	
	ACC	F1	ACC	F1	ACC	F1
100leaves	90.44 (0.3)	90.21 (0.3)	83.07 (1.3)	82.61 (1.4)	90.80 (0.4)	90.68 (0.4)
ALOI	93.73 (0.6)	93.76 (0.6)	86.11 (1.4)	86.74 (1.4)	97.21 (0.1)	97.22 (0.1)
Animals	84.49 (0.1)	77.53 (0.1)	72.54 (0.1)	65.53 (0.1)	84.86 (0.1)	80.07 (0.1)
Caltech102	46.75 (0.3)	22.41 (0.5)	34.20 (0.6)	9.30 (0.6)	51.40 (0.6)	30.76 (0.9)
MNIST	93.41 (0.1)	93.33 (0.1)	86.95 (0.6)	86.75 (0.6)	93.75 (0.1)	93.66 (0.1)
OutScene	78.36 (0.4)	78.61 (0.4)	70.90 (1.0)	71.00 (1.0)	81.04 (0.5)	81.18 (0.5)

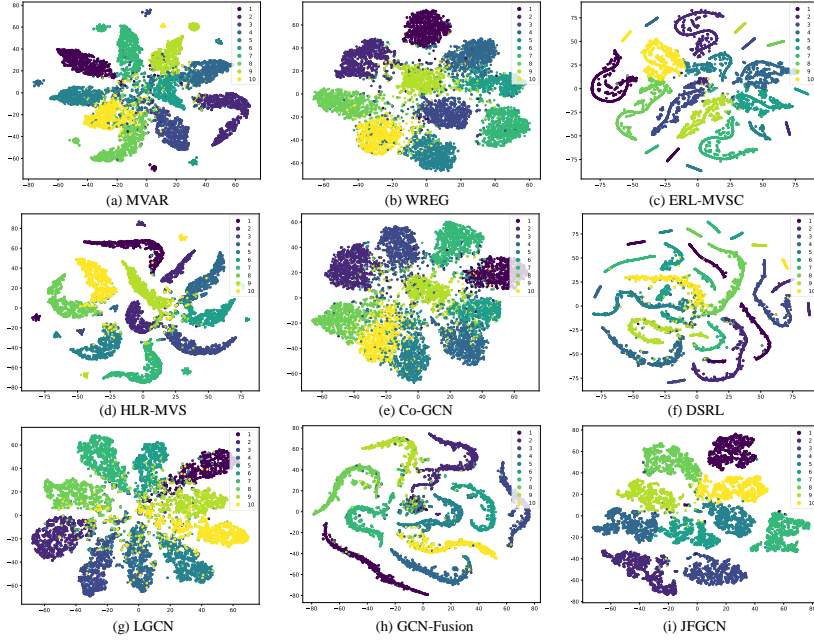
of 3 views, and project the original high-dimensional data onto a 2D space by using t-SNE. Then, the projected 2D data are colored with the class labels. It is shown that the proposed method learns visual representation with large inter-class distances and small intra-class distances.

#### 4.2.2. Ablation Study

In order to verify the effectiveness of each model component, we perform an ablation study as shown in Table 5. JFGCN achieves leading performance on each of these datasets, especially the Caltech102 dataset. The performance of  $\mathbf{A}_{pos}$  alone is a bit better than  $\mathbf{A}_{neg}$ , but adaptive weight learning for both  $\mathbf{A}_{pos}$  and  $\mathbf{A}_{neg}$  can better improve the performance. This demonstrated the feasibility of adaptively regularized GCN modules, which also verifies the validity of both  $\mathbf{A}_{neg}$  and  $\mathbf{A}_{pos}$ . Eventually, JFGCN designed a unified framework for two connective relations to learn better representation.

#### 4.3. Parameter Sensitivity

In this subsection, the parameter sensitivity of the proposed networks is analyzed to check the validity. We perform a sensitivity analysis on parameter  $k$ , which denotes the number of neighbors chosen by  $k$ NN. Fig. 5 shows the sensitivity w.r.t parameter  $k$  on tested datasets. We find that almost all the best performance happens at  $k = 10$ . Except for 100leaves and Caltech102, the optimal  $k$

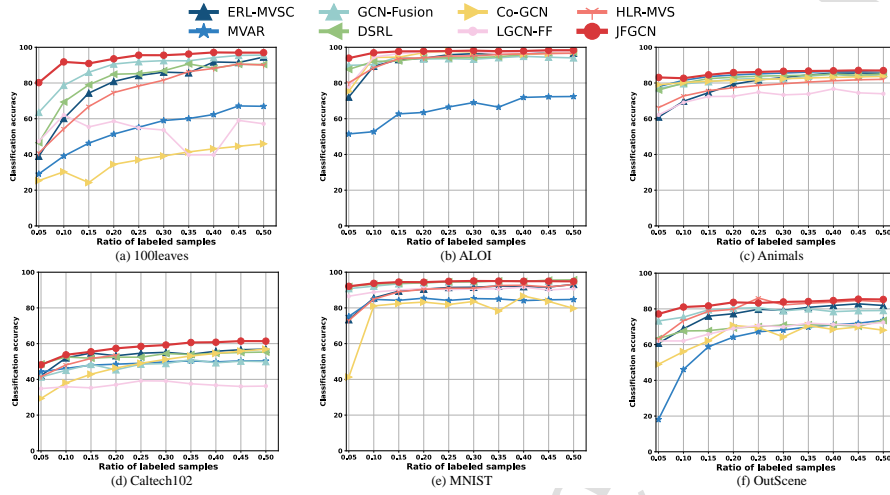


**Figure 2:** Visualization for multi-view semi-supervised classification on MNIST.

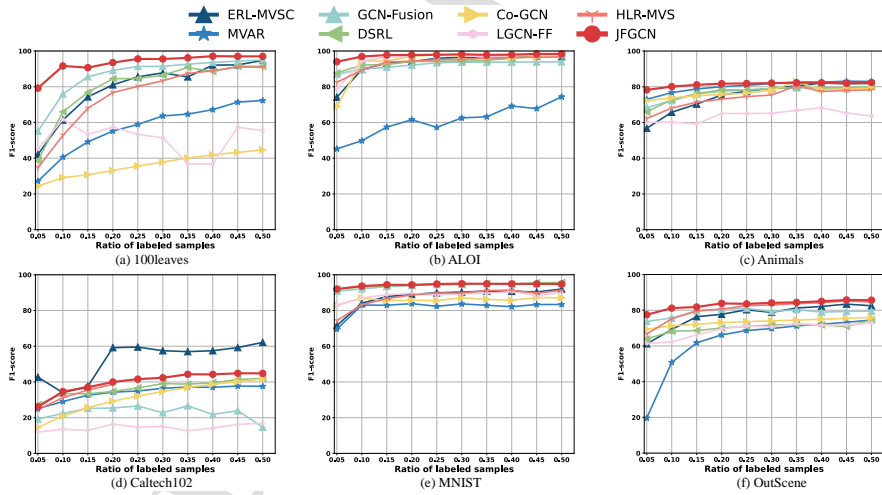
values differ slightly for different datasets. We can observe that  $k$  has little impact on the performance of these datasets, which demonstrates that JFGCN can acquire relatively accurate connective relations even when  $k$  is small. For 100leaves and Caltech102, the fluctuation of performance may be due to a large number of categories. When a large  $k$  is selected, the adjacency matrix generated for  $k$ NN will be dense, resulting in indistinguishable representations.

#### 4.4. Convergence Analyses

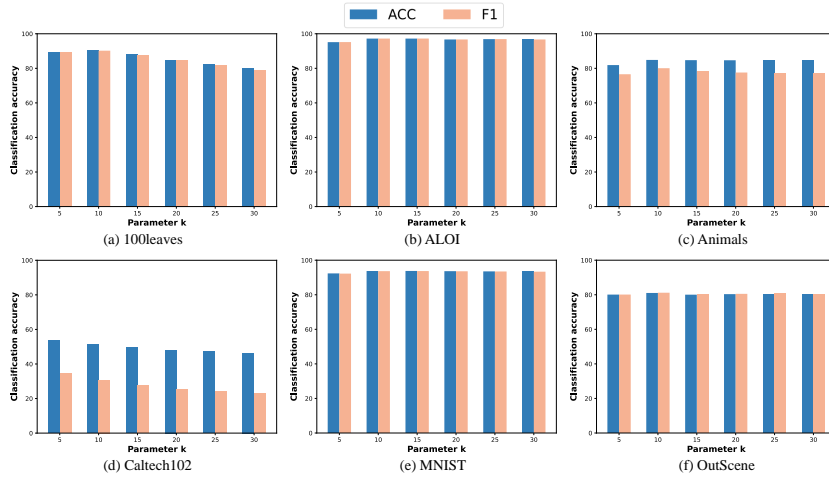
In this subsection, the convergence of the algorithm is analyzed in Fig. 6. The figures demonstrate that the loss curve of the proposed method initially decreases rapidly before reaching a plateau across these datasets as the training progresses. Specifically, for a fixed number of epochs, the loss declines steeply at the beginning while the performance improves significantly. After a sufficient number of epochs, the loss and accuracy attain stable values with minor fluctuations. With the exception of the ALOI and Caltech102 datasets, the loss functions on the remaining datasets converge after around 200 rounds of training, demonstrating that the model’s loss functions converge effectively and facilitate the network’s efficient learning.



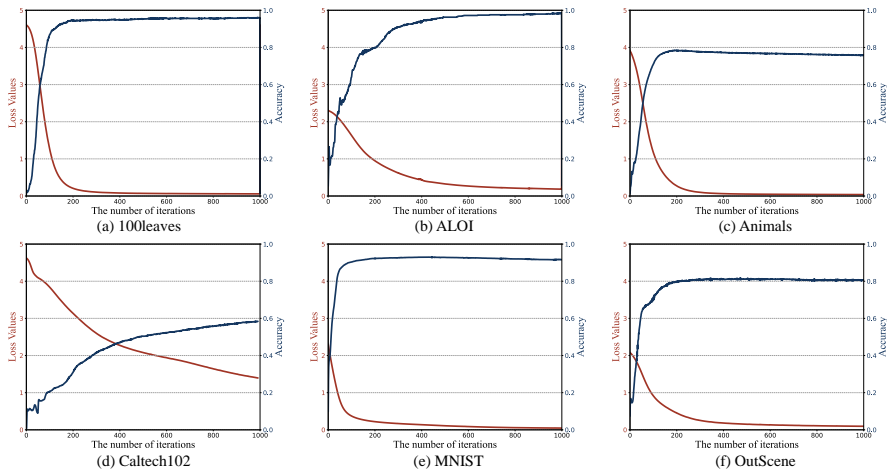
**Figure 3:** The accuracy of all compared methods as the ratio of labeled data ranges in  $\{0.05, 0.10, \dots, 0.5\}$  on the six datasets.



**Figure 4:** The F1-score of all compared methods as the ratio of labeled data ranges in  $\{0.05, 0.10, \dots, 0.5\}$  on the six datasets.



**Figure 5:** The parameter sensitivity analysis of JFGCN with respect to  $k$  on tested datasets.



**Figure 6:** Convergence curves of JFGCN on six tested datasets.



## 5. Conclusion

In this paper, we proposed an end-to-end neural network framework named JFGCN which attempted to accurately utilize the consistency and complementarity of multi-view data. JFGCN aimed to simultaneously implement feature fusion and topology adjustment, making these two modules supervise each other. The proposed JFGCN implicitly learned a consistent low-dimensional representation by using a multi-view autoencoder to approximate the matrix decomposition, which incorporated consistent information across multiple views. In addition, in order to inject a more robust representation of connective relations, JFGCN employed a filtering mechanism to adaptively adjust topological information. The experimental results further validated the superiority of the proposed framework for multi-view semi-supervised classification tasks.

There are still some potential research directions to be further explored. In most practical applications, most multi-view data is not available in natural topology and most topology structure is built by  $k$ NN algorithms, which inevitably leads to the aggregation of undesired connected relations. In future work, we will devote to explore how to further mitigate the propagation of errors in unnecessarily connected relations.

## Acknowledgments

This work is in part supported by the National Natural Science Foundation of China under Grant U21A20472 and 62276065, and the National Key Research and Development Plan of China under Grant 2021YFB3600503.

## References

- [1] C. Huang, J. Cui, Y. Fu, D. Huang, M. Zhao, L. Li, Incomplete multi-view clustering network via nonlinear manifold embedding and probability-induced loss, *Neural Networks* 163 (2023) 233–243.
- [2] H. Yin, W. Hu, Z. Zhang, J. Lou, M. Miao, Incremental multi-view spectral clustering with sparse and connected graph learning, *Neural Networks* 144 (2021) 260–270.
- [3] Z. Lin, Z. Kang, L. Zhang, L. Tian, Multi-view attributed graph clustering, *IEEE Transactions on Knowledge and Data Engineering* (2023) 1872–1880.
- [4] E. Pan, Z. Kang, Multi-view contrastive graph clustering, in: *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021, pp. 2148–2159.

- [5] D. Xie, Q. Gao, M. Yang, Enhanced tensor low-rank representation learning for multi-view clustering, *Neural Networks* 161 (2023) 93–104.
- [6] P. Hu, D. Peng, Y. Sang, Y. Xiang, Multi-view linear discriminant analysis network, *IEEE Transactions on Image Processing* 28 (2019) 5352–5365.
- [7] K. Chumachenko, J. Raitoharju, A. Iosifidis, M. Gabbouj, Speed-up and multi-view extensions to subclass discriminant analysis, *Pattern Recognition* 111 (2021) 107660.
- [8] M. Li, J. Liu, C. Zheng, X. Huang, Z. Zhang, Exploiting multi-view part-wise correlation via an efficient transformer for vehicle re-identification, *IEEE Transactions on Multimedia* 25 (2021) 919–929.
- [9] G. Liu, H. Yue, J. Wu, J. Yang, Intra-inter view interaction network for light field image super-resolution, *IEEE Transactions on Multimedia* 25 (2021) 256–266.
- [10] L. Wang, Z. Ding, Z. Tao, Y. Liu, Y. Fu, Generative multi-view human action recognition, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6212–6221.
- [11] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, T. Funkhouser, Ibrnet: Learning multi-view image-based rendering, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4690–4699.
- [12] Z. Song, Y. Zhang, I. King, Towards an optimal asymmetric graph structure for robust semi-supervised node classification, in: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1656–1665.
- [13] Q. Yue, J. Liang, J. Cui, L. Bai, Dual bidirectional graph convolutional networks for zero-shot node classification, in: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2408–2417.
- [14] D. Li, H. Wang, Y. Wang, S. Wang, Instance-wise multi-view representation learning, *Information Fusion* 91 (2023) 612–622.
- [15] W. Xia, Q. Wang, Q. Gao, X. Zhang, X. Gao, Self-supervised graph convolutional network for multi-view clustering, *IEEE Transactions on Multimedia* 24 (2021) 3182–3192.

- [16] X. Jia, X. Jing, Q. Sun, S. Chen, B. Du, D. Zhang, Human collective intelligence inspired multi-view representation learning - enabling view communication by simulating human communication mechanism, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023) 7412–7429.
- [17] E. Pan, Z. Kang, Beyond homophily: Reconstructing structure for graph-agnostic clustering, *CoRR* abs/2305.02931 (2023). doi:10.48550/arXiv.2305.02931.
- [18] N. Halliwell, Evaluating explanations of relational graph convolutional network link predictions on knowledge graphs, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 12880–12881.
- [19] S. Yun, S. Kim, J. Lee, J. Kang, H. J. Kim, Neo-GNNs: Neighborhood overlap-aware graph neural networks for link prediction, *Advances in Neural Information Processing Systems* (2021) 13683–13694.
- [20] M. Zhang, Y. Chen, Link prediction based on graph neural networks, *Advances in Neural Information Processing Systems* (2018) 5171–5181.
- [21] W. Guan, X. Song, H. Zhang, M. Liu, C.-H. Yeh, X. Chang, Bi-directional heterogeneous graph hashing towards efficient outfit recommendation, in: *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 268–276.
- [22] W. Guan, H. Wen, X. Song, C. Wang, C.-H. Yeh, X. Chang, L. Nie, Partially supervised compatibility modeling, *IEEE Transactions on Image Processing* (2022) 4733–4745.
- [23] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: *Proceedings of the International Conference on Learning Representations*, 2017, pp. 1–14.
- [24] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, J. Pei, Am-gcn: Adaptive multi-channel graph convolutional networks, in: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 1243–1253.
- [25] S. Li, W.-T. Li, W. Wang, Co-gcn for multi-view semi-supervised learning, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 4691–4698.
- [26] Y. Wang, D. Chang, Z. Fu, Y. Zhao, Consistent multiple graph embedding for multi-view clustering, *IEEE Transactions on Multimedia* 25 (2023) 1008–1018.

- [27] Y. Wang, D. Chang, Z. Fu, J. Wen, Y. Zhao, Incomplete multiview clustering via cross-view relation transfer, *IEEE Transactions on Circuits and Systems for Video Technology* 33 (2022) 367–378.
- [28] H. Yu, J. Tang, G. Wang, X. Gao, A novel multi-view clustering method for unknown mapping relationships between cross-view samples, in: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2075–2083.
- [29] C. Zhang, Y. Liu, Y. Liu, Q. Hu, X. Liu, P. Zhu, Fish-mml: Fisher-hsic multi-view metric learning., in: *Proceedings of the International Joint Conference on Artificial Intelligence*, 2018, pp. 3054–3060.
- [30] Z. Jiang, X. Liu, Adaptive knn and graph-based auto-weighted multi-view consensus spectral learning, *Information Sciences* 609 (2022) 1132–1146.
- [31] L. Zong, X. Zhang, L. Zhao, H. Yu, Q. Zhao, Multi-view clustering via multi-manifold regularized non-negative matrix factorization, *Neural Networks* 88 (2017) 74–89.
- [32] S. Pang, K. Zhang, G. Wang, J. C.-W. Lin, F. Wang, X. Meng, S. Wang, Y. Zhang, Af-gcn: Completing various graph tasks efficiently via adaptive quadratic frequency response function in graph spectral domain, *Information Sciences* 623 (2023) 469–480.
- [33] F. Feng, X. He, H. Zhang, T.-S. Chua, Cross-gcn: Enhancing graph convolutional network with  $k$ -order feature interactions, *IEEE Transactions on Knowledge and Data Engineering* 35 (2021) 225–236.
- [34] F. Nie, J. Li, X. Li, et al., Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification., in: *Proceedings of the International Joint Conference on Artificial Intelligence*, 2016, pp. 1881–1887.
- [35] X. Jia, X. Jing, X. Zhu, S. Chen, B. Du, Z. Cai, Z. He, D. Yue, Semi-supervised multi-view deep discriminant representation learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021) 2496–2509.
- [36] Z. Wu, X. Lin, Z. Lin, Z. Chen, Y. Bai, S. Wang, Interpretable graph convolutional network for multi-view semi-supervised learning, *IEEE Transactions on Multimedia* (2023) 1–14.

- [37] K. Yao, J. Liang, J. Liang, M. Li, F. Cao, Multi-view graph convolutional networks with attention mechanism, *Artificial Intelligence* (2022) 103708.
- [38] H. Tao, C. Hou, F. Nie, J. Zhu, D. Yi, Scalable multi-view semi-supervised classification via adaptive regression, *IEEE Transactions on Image Processing* 26 (2017) 4283–4296.
- [39] M. Yang, C. Deng, F. Nie, Adaptive-weighting discriminative regression for multi-view classification, *Pattern Recognition* 88 (2019) 236–245.
- [40] Y. Xie, W. Zhang, Y. Qu, L. Dai, D. Tao, Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning, *IEEE Transactions on Cybernetics* 50 (2020) 572–586.
- [41] A. Huang, Z. Wang, Y. Zheng, T. Zhao, C.-W. Lin, Embedding regularizer learning for multi-view semi-supervised classification, *IEEE Transactions on Image Processing* 30 (2021) 6997–7011.
- [42] S. Wang, Z. Chen, S. Du, Z. Lin, Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44 (2022) 5042–5055.
- [43] Z. Chen, L. Fu, J. Yao, W. Guo, C. Plant, S. Wang, Learnable graph convolutional network and feature fusion for multi-view learning, *Information Fusion* 95 (2023) 109–119.

## Highlights

- Propose an end-to-end framework for multi-view semi-supervised classification
- Design a multi-view auto-encoder to fuse feature by approximating the matrix decomposition
- Explore a more robust topology that fuses the adjacency matrices generated by  $k$ NN and  $k$ FN

**Declaration of interests**

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

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.