Generative Essential Graph Convolutional Network for Multi-view Semi-supervised Classification

Jielong Lu, Zhihao Wu, Luying Zhong, Zhaoliang Chen, Hong Zhao, Shiping Wang

***Abstract*—Multi-view learning is a promising research field that aims to enhance learning performance by integrating information from diverse data perspectives. Due to the increasing interest in graph neural networks, researchers have gradually incorporated various graph models into multi-view learning. Despite significant progress, current methods face challenges in extracting information from multiple graphs while simultaneously accommodating specific downstream tasks. Additionally, the lack of a subsequent refinement process for the learned graph leads to the incorporation of noise. To address the aforementioned issues, we propose a method named generative essential graph convolutional network for multi-view semi-supervised classification. Our approach integrates the extraction of multi-graph consistency and complementarity, graph refinement, and classification tasks within a comprehensive optimization framework. This is accomplished by extracting a consistent graph from the shared representation, taking into account the complementarity of the original topologies. The learned graph is then optimized through downstream-specific tasks. Finally, we employ a graph convolutional network with a learnable threshold shrinkage function to acquire the graph embedding. Experimental results on benchmark datasets demonstrate the effectiveness of our approach.**

***Index Terms*—Muti-view learning, graph convolutional network, learnable graph, learnable threshold shrinkage activation.**

I. Introduction

With the rapid development of multimedia technology, acquiring multiple views of an object has become more accessible across various fields. For instance, news articles may be covered by multiple news outlets, and an image can be analyzed using different feature extractors. These different perspectives or representations of data are collectively referred to as multi-view data. Compared to single-view data, multi-view data provides more comprehensive feature information [1]. Building on these datasets, a learning paradigm known as multi-view learning has emerged. Multi-view learning has demonstrated remarkable success in various applications, including data mining [2], [3], [4], [5], [6], machine learning [7], [8], [9], [10], [11], and computer vision [12], [13], [14].

In real-world scenarios, collecting sufficient labeled data, especially for multi-view data, is often challenging due to the need for expertise and considerable time investment. Therefore, it is essential to effectively leverage unlabeled data in multi-view learning. Extensive research has demonstrated that multi-view semi-supervised classification methods can effectively utilize a small amount of labeled data to guide the prediction of a larger amount of unlabeled data [15], [16], [17], [18]. In recent years, graphs have emerged as a powerful tool for analyzing non-Euclidean data [19]. For example, graphs have been utilized to model the intricate relationships among individuals in social networks and the interaction forces between molecules. This trend has played a crucial role in advancing the development of graph-based methods. Among numerous multi-view semi-supervised algorithms, graph-based techniques have garnered significant attention due to their ability to construct a graph where the nodes represent labeled and unlabeled examples and propagate label information through the edges. However, numerous graph-based multi-view semi-classification methods are shallow models that face challenges in effectively integrate the structural information of the graph with the intrinsic characteristics of the data. Consequently, there is an increasing anticipation of harnessing the potential of deep learning to attain more favorable outcomes.

Graph Convolutional Network (GCN) [20] has garnered considerable attention due to its ability to encode both graph structure and sample features as potential representations. This unique property has led to significant improvements in various applications, including link prediction [21], [22], [23], action recognition [24], [25], [26] and recommendation systems [27], [28], [29]. Many researchers have introduced GCNs into the multi-view domain by mining hidden connections on multi-view data [30]. However, the majority of constructed multi-view graphs often neglect downstream-specific tasks while seeking consistent and complementary connections among views, leading to underutilization of feature information. Furthermore, there is a notable absence of a refinement process for the constructed graphs, which allows the noisy connections to impact the performance of GCNs.

To address the aforementioned challenges, we propose a framework named Generative Essential Graph Convolutional Network (GEGCN). This framework integrates the process of constructing the graph with the classification task while simultaneously extracting topological consistency and complementarity across multiple views. Specifically, the generator

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. (*Corresponding author: Shiping Wang.*)

Jielong Lu, Zhihao Wu, Luying Zhong, Zhaoliang Chen and Shiping Wang are with the College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China and also with the Key Laboratory of Intelligent Metro, Fujian Province University, Fuzhou 350108, China (email: jielonglu2022@163.com, zhihaowu1999@gmail.com, luyingzhongfzu@163.com, chenzl23@outlook.com, shipingwangphd@163.com).

Zhao Hong is with the School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China, and with the Key Laboratory of Data Science and Intelligence Application, Minnan Normal University, Zhangzhou, Fujian, 363000, China (e-mail: hongzhaocn@163.com).

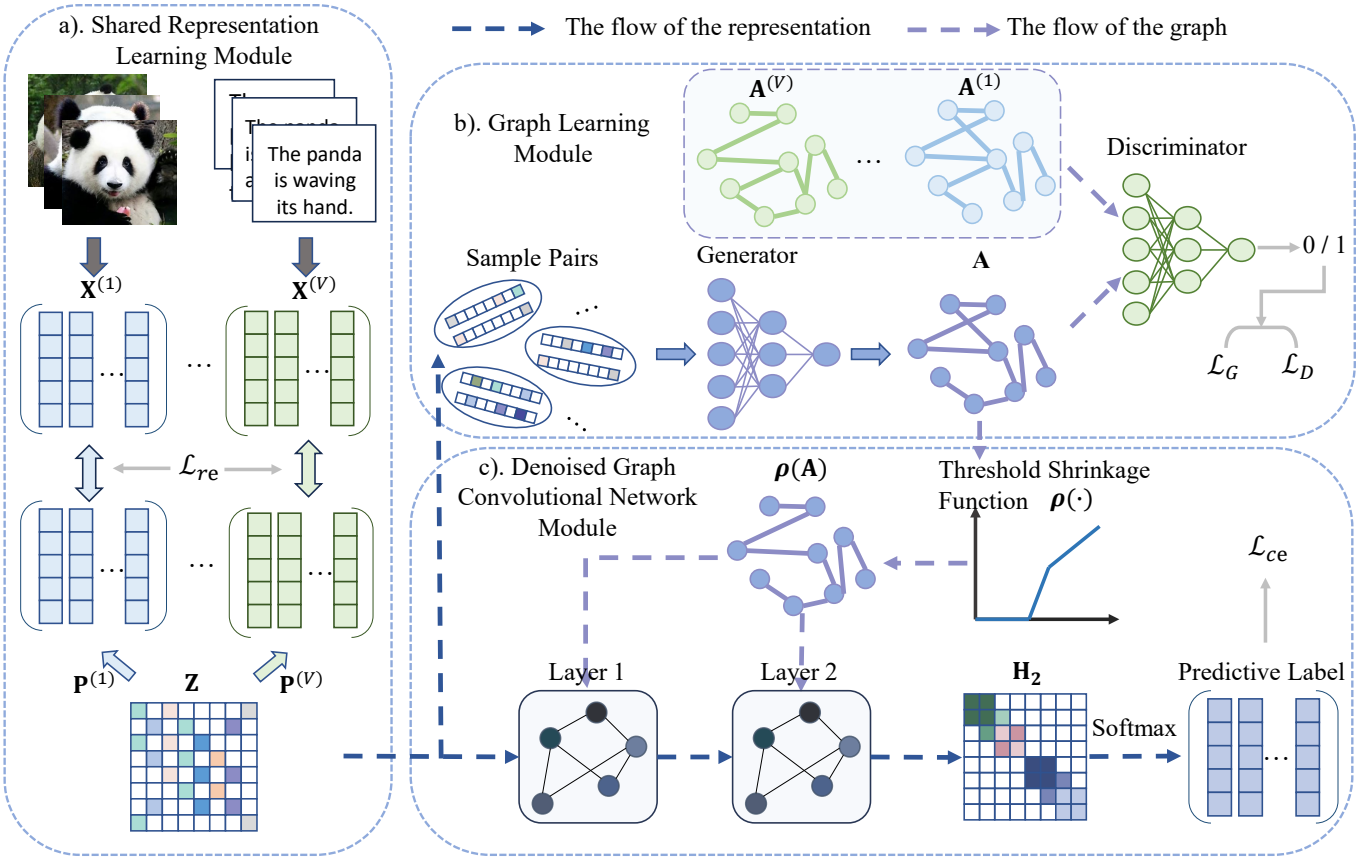


Fig. 1. Overview of the proposed GEGCN framework which includes: a) Shared Representation Learning Module; b) Graph Learning Module; c) Denoised GCN Module. The sample pairs from $\mathbf{Z}$ represent a set of tuples, each consisting of any two samples that are in the shared representation matrix $\mathbf{Z}$, such as $\{(\mathbf{z}_1, \mathbf{z}_1), (\mathbf{z}_1, \mathbf{z}_2), ..., (\mathbf{z}_n, \mathbf{z}_n)\}$.

network utilizes the shared representations obtained from multiple views to learn the graph and incorporate information from downstream tasks. In contrast, discriminator networks leverage geometric priors in the original space, considering the unique connectivity specific to each graph. Additionally, we introduce a denoised GCN that incorporates an adaptive threshold shrinkage function. This function plays a crucial role in the process of graph learning by selectively retaining strong connections while disregarding unimportant ones in the graph. As a result, the technique learns refined graph and extracts robust graph embeddings.

In terms of the training approach, we employ a strategy where each submodule optimizes its parameters independently. The framework is demonstrated in Fig. 1. The main contributions of this paper are summarized as follows:

- Propose a framework that extracts both the consistency and complementarity of multi-view topologies while linking them to downstream tasks.
- Develop a denoised GCN that contains a learnable threshold shrinkage function to adaptively filter the noise contained in the learned graph.
- Substantial experiments on benchmark datasets show that the proposed framework achieves superior performance compared to other state-of-the-art algorithms.

II. Related Work

In this section, a comprehensive review of the literature related to our study is provided. Firstly, an overview of graph convolutional networks is presented, followed by a detailed discussion of different approaches to multi-view learning. Lastly, the primary techniques used for constructing multi-view graphs are outlined.

A. Graph Convolutional Network

Graph Convolutional Network (GCN) was proposed by Kipf and Welling [20] as a powerful tool for analyzing graph data. The propagation rule of GCN is shown below:

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

where $\mathbf{H}^{(l)}$ and $\mathbf{H}^{(l+1)}$ represent the input and output of the l-th layer graph convolution, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with added self-loops, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{W}^{(l)}$ is the learnable parameter of the l-th layer, and $\sigma(\cdot)$ is the activation function. Here, $\mathbf{I}$ represents the identity matrix. Due to the effectiveness of GCN, many variants and extensions have been proposed, which have shown encouraging performance [31], [32]. Peng et al. [33] utilized a combination of graph convolutional networks and matrix decomposition to capture nonlinear interactions and leveraged the similarity of measurements. Wang et al. [34] dissected the steps of linear

GCN from the perspective of continuous graph diffusion, and analyzed why additional propagation steps did not benefit linear GCN, and proposed decoupling graph convolution to separate terminal time and feature propagation. Zhong et al. [35] proposed a comparative learning framework that jointly utilized topological graph and adaptive graph with feature information, and constructed a self-supervised mechanism to enrich the limited label information. Chen et al. [36] utilized two simple and effective techniques: initial residuals and identity mapping, and provided theoretical and empirical evidence that these two techniques can effectively mitigate the problem of over-smoothing. While existing GCN models demonstrate commendable performance on graph-structured data, certain limitations arise when dealing with data that lacks natural topology.

B. Multi-view Learning

Multi-view learning has demonstrated remarkable effectiveness in diverse practical domains. Processing multi-view data typically involves exploring the consistency and complementarity between diverse views to extract valuable information. Sun et al. [37] introduced a novel approach to leverage different views with consistent categorical margins, presenting a generalized form of a multi-view maximum entropy discriminant solution instantiated under a specific a priori formulation. Chao et al. [38] proposed a method for multi-view classification that aligned data from different views into a shared subspace while incorporating original features to enhance consensus and complementarity. Peng et al. [39] proposed a novel objective function to simultaneously obtain geometric consistency and clustering assignment consistency in the data, thereby facilitating the direct learning of almost any parameter from the data. Chao et al. [17] employed geometric information from marginal distributions in unlabeled data to construct semi-supervised classifiers and leveraged expectation Laplace regularization into probabilistic models for semi-supervised learning. Lin et al. [40] proposed a joint framework that aimed to maximize the mutual information of different views through contrastive learning for consistency and minimized the conditional entropy through pairwise prediction to recover missing views. Lin et al. [41] achieved the learning of consistent representations for different views and the recovery of missing views by simultaneously maximizing the mutual information across diverse views and minimizing the conditional entropy of pairwise predictions. Yang et al. [42] introduced a contrastive learning paradigm aimed at addressing both the issue of the partial view inconsistency and the problem of the partial sample missing within a unified framework. All of these studies collectively demonstrate that multi-view learning exhibits superior performance compared to single-view learning. However, a common drawback across them is the absence of topological consistency and specificity in extracting multi-view graphs. This limitation presents challenges in effectively developing graph-based multi-view methods.

C. Multi-view Graph Construction

When using graph-based algorithms, it is essential to construct graphs for real-world multi-view data, as they frequently lack a natural topology. One of the most commonly used graph construction algorithms is the k-nearest neighbors (kNN), which achieves promising results [43], [44], defined as

$$\mathbf{A}_{ij}^{(v)} = \begin{cases} 1, \ \mathbf{x}_i^{(v)} \in k\text{NN}(\mathbf{x}_j^{(v)}) \text{ or } \mathbf{x}_j^{(v)} \in k\text{NN}(\mathbf{x}_i^{(v)}), \\ 0, \ \text{otherwise}, \end{cases} \tag{2}$$

where $\mathbf{A}_{ij}^{(v)}$ is the (i,j)-th element of the v-th view adjacency matrix, $\mathbf{x}_i^{(v)}$ represents the i-th row vector of $\mathbf{X}^{(v)}$, and $k\text{NN}(\mathbf{x}_i^{(v)})$ denotes the set of k nearest neighbors of $\mathbf{x}_i^{(v)}$. Tang et al. [45] took a traditional predefined graph matrix, such as a cosine similarity graph, and learned an improved graph for each individual view to capture the geometric structure of the original space using an iterative cross-diffusion process. Li et al. [46] introduced a method that constructs an intrinsic similarity graph in a spectral embedding space rather than the original feature space. Wu et al. [47] learned the view-specific affinity matrix based on the projection map and its intrinsic tensor using low-rank tensor approximation, and jointly learned the optimal affinity matrix by integrating these two conditions. While current methods for constructing multi-view graphs have to be promising, they encounter challenges in integrating the extraction of both consistency and complementarity from multi-view graphs while linking them with specific downstream tasks. This limitation may lead to the constructed graphs that are not sufficiently informative and tend to be suboptimal for addressing downstream tasks.

III. The Proposed Method

A. Overview and Notations

In this section, we denote multi-view data to be $\mathcal{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(V)}\}$, where V is the number of views and $\mathbf{X}^{(v)}$ is the v-th view data. For the v-th view, we have $\mathbf{X}^{(v)} \in \mathbb{R}^{n \times d_v}$, where n represents the number of data and d_v represents the feature dimension. We present a unified optimization framework for multi-view semi-supervised classification. The framework extracts both topological consistency and complementarity while integrating multi-view graph construction and the downstream classification tasks. We introduce three iterative optimization networks to ensure that the learned graph effectively captures both the labeling information of the downstream task and the consistency and complementarity of the geometry of the original space. In addition, our method employs a learnable threshold shrinkage function that dynamically adjusts the threshold to improve the quality of the underlying graph.

To facilitate the understanding of the mathematical symbols used in this paper, we have included Table I with an explanation for each symbol. This table clarifies the meaning and interpretation of the symbols, enhancing the comprehension of the presented concepts.

TABLE I
SYMBOLIC NOTATIONS WITH THEIR DESCRIPTIONS.

Notations	Descriptions
$\mathbf{X}^{(v)}$	The v-th view feature matrix.
$\mathbf{A}^{(v)}$	The graph constructed from $\mathbf{X}^{(v)}$ by kNN.
$\mathbf{P}^{(v)}$	The v-th view projection matrix.
$\mathbf{Y}$	Label matrix with $n \times c$.
$\mathbf{Z}$	The shared representation of the features.
$\mathbf{O}^{(l,v)}$	Input of the network to the l-th layer.
$\mathbf{W}_p^{(l,v)}$, $b_p^{(l,v)}$	Weights and biases of the network.
Φ	Learnable parameters of the generator.
$\mathbf{W}_d$	Learnable parameters of the discriminator.
$\mathbf{W}^{(l)}$	Learnable weight matrix of the l-th layer of GCN.
$\mathbf{H}^{(l)}$	Input of GCNs in the l-th layer, where $\mathbf{H}^{(0)} = \mathbf{Z}$.
$\mathbf{A}$	The learned adjacency matrix.
$\rho(\cdot)$	Learnable threshold shrinkage function.
θ, θ_1, θ_2	Learnable variables in $\rho(\cdot)$.
ϵ	The gap between θ_1 and θ_2.
$\sigma(\cdot)$	Optional activation function.

B. Shared Representation Learning Module

Shared Representation Extraction Network. To investigate the intricate connections among different views, our method draws inspiration from the reconstruction approach and posits multiple views stemming from a shared latent representation. Given a multi-view dataset $\mathcal{X} = \{\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(V)}\}$, where $\mathbf{X}^{(v)} \in \mathbb{R}^{n \times d_v}$ denotes the features of the v-th view with d_v dimensions. We aim to obtain a shared representation $\mathbf{Z}$ that can be transformed back to the original feature space using different projection matrices. The objective function can be defined as

$$\min_{\mathbf{Z}, \mathbf{P}^{(v)}} \sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{Z}\mathbf{P}^{(v)}\|_F^2, \tag{3}$$

where $\mathbf{P}^{(v)}$ denotes the projection matrix of the v-th view and $\|\cdot\|_F$ represents the Frobenius norm. To enhance the flexibility of this process, we introduce a neural network-based approach for approximation. The formulation of this approach is as follows,

$$\mathbf{O}^{(l+1,v)} = \sigma(\mathbf{O}^{(l,v)}\mathbf{W}_p^{(l,v)} + \mathbf{b}_p^{(l,v)}), \tag{4}$$

where $\mathbf{O}^{(0,v)} = \mathbf{Z}$, $\mathbf{W}_p^{(l,v)}$ and $\mathbf{b}_p^{(l,v)}$ are weight and bias of l-th layer, respectively, and $\sigma(\cdot)$ is the layer-specific activation function. The loss function of this process is defined as

$$\mathcal{L}_{re} = \sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{O}^{(L,v)}\|_F^2, \tag{5}$$

where L is the layer number of the neural network. Considering that the shared representations above only take into account the views themselves without considering downstream tasks, our objective is to integrate both inter-view consistency and downstream task information into the learning process. To accomplish this, we combine the classification loss $\mathcal{L}_{ce}$ and the reconstruction loss $\mathcal{L}_{re}$ to derive the final reconstruction loss $\mathcal{L}_{re'}$. The specifics are as follows:

$$\mathcal{L}_{re'} = \mathcal{L}_{re} + \alpha\mathcal{L}_{ce}, \tag{6}$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss which will be defined in the following part and α is the hyperparameter.

C. Graph Learning Module

To capture the inherent connectivity patterns across multiple views, we employ a generator network tasked with transforming the acquired shared representations into adjacency matrices, reflecting the consistent similarity between samples. Subsequently, the discriminator network utilizes the graphs in the original space as a geometric prior, dynamically adjusting the weight assignment of each node's neighbors to capture complementarity. The details of this process are outlined below.

Graph Discriminator. The discriminator $\mathcal{D}$ assigns a score to each edge in a given adjacency matrix $\mathbf{A}$. A higher score indicates a better quality of the node connection. The form is as follows,

$$\mathcal{D}(\mathbf{A}) = Diag(\sigma(\mathbf{A}\mathbf{W}_d)), \tag{7}$$

where $Diag$: $\mathbb{R}^{n \times n} \to \mathbb{R}^{n \times 1}$ extracts the diagonal elements of the matrix as a vector, and $\mathbf{W}_d$ is a learnable scoring matrix. To ensure that the generator retains the distinctive information about the topology of the original space, we incorporate the node connections from the original space as the geometric prior with high scores. The primary objective of the discriminator is to assign a high score to the topology observed in the original space while assigning a low score to the topology generated by the generator. The objective function is as follows,

$$\max_{\mathbf{W}_d} \sum_{v=1}^{V} \sum_{i=1}^{n} \mathcal{D}(\mathbf{A}^{(v)})_i + \min_{\mathbf{W}_d} \sum_{i=1}^{n} \mathcal{D}(\mathcal{G}(\mathbf{Z}))_i, \tag{8}$$

where $\mathbf{A}^{(v)}$ is the adjacency matrix of the v-th view constructed by kNN, and $\mathcal{G}(\cdot)$ is the graph generator defined as follows. The loss function of the discriminator can be defined as

$$\mathcal{L}_D = -\sum_{v=1}^{V} \sum_{i=1}^{n} \log[\mathcal{D}(\mathbf{A}^{v})_i] - \sum_{i=1}^{n} \log[1 - \mathcal{D}(\mathcal{G}(\mathbf{Z})_i)]. \tag{9}$$

Graph Generator. The graph generator utilizes the shared representation to establish the similarities between pairs of samples in a given space. We propose that the graph structure is a function of the features and define $\mathbf{A}$ as follows,

$$\mathbf{A}_{ij} = \mathcal{G}(\mathbf{Z})_{ij} = \varphi\left(\frac{\mathrm{MLP}_\Phi([\mathbf{z}_i; \mathbf{z}_j]) + \mathrm{MLP}_\Phi([\mathbf{z}_j; \mathbf{z}_i])}{2}\right), \tag{10}$$

where $\varphi(\cdot) : \mathbb{R} \to [0, 1]$, $\mathbf{z}_i$ is the i-th sample of the shared representation $\mathbf{Z}$, MLP refers to a multi-layer perceptron parameterized with Φ and $[\cdot;\cdot]$ denotes the concatenation operator. We intentionally enforce $\mathbf{A}_{ij} = \mathbf{A}_{ji}$ to ensure the symmetry of the generative graph structure, as our primary focus is on dealing with symmetric graphs. However, it is also possible to adapt this approach to an asymmetric graph by adjusting the following $\mathbf{A}_{ij} = \varphi(\mathrm{MLP}_\Phi([\mathbf{z}_i; \mathbf{z}_j])$.

The goal of the graph generator is to learn node connectivity that can be assigned high scores from the discriminator. This objective is defined by the following objective function,

$$\max_{\Phi} \sum_{i=1}^{n} \mathcal{D}(\mathcal{G}(\mathbf{Z}))_i. \tag{11}$$

The loss function of the generator can be defined as

$$\mathcal{L}_G = -\sum_{i=1}^{n} \log[\mathcal{D}(\mathcal{G}(\mathbf{Z}))_i]. \tag{12}$$

Upon further consideration, we acknowledge that the previously mentioned objective function for obtaining the adjacency matrix does not explicitly consider the requirements of specific downstream tasks. To overcome this limitation, we propose incorporating information of the node labels to provide additional guidance for the generation of the underlying connections. This can be achieved by introducing a loss function,

$$\mathcal{L}_{ce} = \sum_{i\in\Omega} f(\text{GCN}(\mathcal{G}(\mathbf{Z}), \mathbf{Z})_i, \mathbf{Y}_i), \tag{13}$$

where the function $f(\cdot)$ represents the discrepancy between the predicted label obtained by the learned graph and the actual value of the sample. The sample set Ω represents the label matrix derived from $\mathbf{Y} \in \mathbb{R}^{n\times c}$. GCN$(\cdot)$ refers to a denoised graph convolutional network that incorporates a learnable threshold contraction function. The details of this function are described in the subsequent subsection. The modified loss function of the generator can be defined as

$$\mathcal{L}_{G'} = \mathcal{L}_G + \mathcal{L}_{ce}. \tag{14}$$

D. Denoised Graph Convolutional Network

Due to the possibility of minor noise in the essential graph obtained from the previous module, a refinement process is required to filter out insignificant values without compromising the strong connectivity relationships. In this regard, we introduce an adaptive threshold shrinkage function that can effectively refine the learned adjacency matrix. This function $\rho(\cdot)$ is defined as follows,

$$\rho(x;\theta_1,\theta_2) = w_1\text{ReLU}(x-\theta_1) - w_2\text{ReLU}(x-\theta_2), \tag{15}$$

where $w_1 = \frac{\theta_2}{\theta_2-\theta_1}$ and $w_2 = \frac{\theta_1}{\theta_2-\theta_1}$, θ_1 and θ_2 are learnable parameters with $0 < \theta_1 < \theta_2 < 1$. To ensure this constraint is satisfied during the training process, a learnable variable θ and an activation function Sigmoid are used to generate θ_1 and θ_2. The details are shown below,

$$\theta_1 = \frac{\text{Sigmoid}(\theta)}{2}, \theta_2 = \frac{\text{Sigmoid}(\theta)}{2} + \epsilon,$$

where ϵ is a pre-defined parameter used to measure the gap between θ_1 and θ_2. Equation (15) is called Learnable Piecewise ReLU (LPReLU). The comparison of ReLU, Soft Thresholding, and LPReLU is shown in Fig. 2. The update formula for GCN with threshold function is defined as follows

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\rho(\tilde{\mathbf{A}};\theta_1,\theta_2)\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{16}$$

where $\mathbf{H}^{(0)} = \mathbf{Z}$. The prediction carried out by a two-layer GCN can be written as

$$\mathbf{H}^{(2)} = Softmax(\hat{\mathbf{A}}\sigma(\hat{\mathbf{A}}\mathbf{H}^{(0)}\mathbf{W}^{(1)})\mathbf{W}^{(2)}), \tag{17}$$

where $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\rho(\tilde{\mathbf{A}};\theta_1,\theta_2)\tilde{\mathbf{D}}^{-\frac{1}{2}}$ and the loss of GCN with learnable threshold shrinkage function can be calculated as

$$\mathcal{L}_{ce} = -\sum_{i\in\Omega}\sum_{j=1}^{c} \mathbf{Y}_{ij}\log\mathbf{H}_{ij}. \tag{18}$$

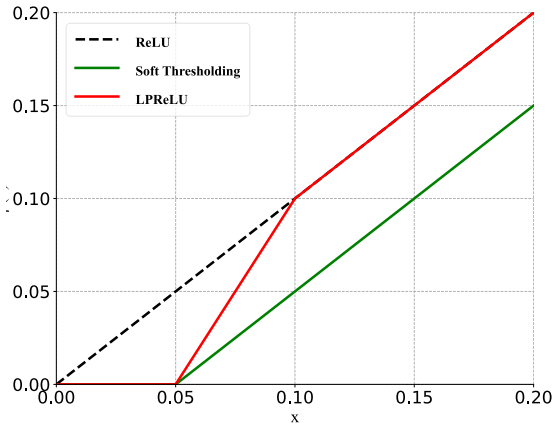


Fig. 2. The comparison among ReLU, Soft thresholding, and LPReLU functions has been conducted, where θ_1 is set to 0.05, θ_2 is set to 0.1, and the soft thresholding is defined as ReLU$(x-\theta_1)$.

E. Model Training

To obtain the final predictive representation, we initially acquire a shared representation across multiple views using the shared representation learning module. Subsequently, our graph learning module undergoes alternating training to learn an essential graph. This learned graph is further refined using a learnable threshold shrinkage function $\rho(\cdot)$. The refined essential graph $\mathbf{A}$ is then combined with the shared representation $\mathbf{Z}$ using GCN to derive the predictive representation $\mathbf{H}$.

The training strategy adopts a multi-step optimization approach. This approach consists of four main steps: optimizing the parameters of the shared representation learning network, optimizing the parameters of the generator, optimizing the parameters of the discriminator, and optimizing the network parameters of the GCN with a learnable threshold shrinkage function. Each module performs one step of forward propagation and one step of backward propagation in each iteration to update its network parameters based on its corresponding loss function. The procedure for the proposed method is outlined in Algorithm 1.

IV. Experimental Analysis

A. Experimental Setup

In this section, we evaluate the performance of the proposed method on seven real-world datasets. Table II comprises a description of these datasets, including the numbers of samples, views, and data types.

TABLE II
A BRIEF DESCRIPTION OF THE TESTED DATASETS.

Datasets	# Samples	# Views	# Classes	# Data types
GRAZ02	1,476	6	4	Object image
HW	2,000	6	10	Digit image
OutScene	2,688	4	8	Object image
100leaves	1,600	3	100	Object image
MNIST	10,000	3	10	Digit image
Scene15	4,485	3	15	Object image
NoisyMNIST	15,000	2	10	Digit image

Algorithm 1 Generative Essential Graph Convolutional Network (GEGCN)

Input: Multi-view data $\mathcal{X} = \{\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(V)}\}$, label set $\mathbf{Y}$, the number of GCN layers L, the number of neighbors k and the hyperparameters ϵ and α.

Output: Predictive output $\mathbf{H}$.

Initialize the parameters $\{\mathbf{W}_p^{(l,v)}, \mathbf{b}_p^{(l,v)}\}_{l=1}^{L}$ of the consistent representation learning module;
Initialize the parameters Φ and $\mathbf{W}_d$ of the generator $\mathcal{G}$ and the discriminator $\mathcal{D}$ respectively;
Initialize the parameters θ and $\{\mathbf{W}^{(l)}\}_{l=1}^{L}$ of the denoised graph convolutional network;
Calculate adjacency matrices $\{\mathbf{A}^v\}_{v=1}^{V}$ via kNN;
while not convergent **do**
  Compute $\mathbf{Z}$, optimize $\{\mathbf{W}_p^{(l,v)}, b_p^{(l,v)}\}_{l=1}^{L}$ by backward propagation with Equation (6);
  Compute $\mathbf{A}$ by $\mathcal{G}(\mathbf{Z})$, and optimize Φ by backward propagation with Equation (14);
  Compute $\{\mathcal{D}(\mathbf{A}^v)\}_{v=1}^{V}$ and $\mathcal{D}(\mathbf{A})$, and optimize $\mathbf{W}_d$ by backward propagation with Equation (9);
  Calculate $\mathbf{H}$ of the learnable GCN with Equation (17);
  Update $\{\mathbf{W}^{(l)}\}_{l=1}^{L}$ and θ of the learnable GCN with back propagation;
end while
return Predictive output $\mathbf{H}$.

- GRAZ02 [48]: This widely used object dataset comprises images from four different classes and includes six commonly used representations: 512-D GIST features, 225-D WT features, 256-D LBP features, 500-D SIFT features, 500-D SURF features, and 680-D PHOG features.
- HW [48]: It consists of 2,000 pictures categorized into 6 classes, with 153-D Profile-correlation features, 596-D Fourier-coefficient features, 301-D Karhunen-Loeve-coefficient features, 27-D Morphological features, 481-D intensity-averaged features, 157-D Zernike Moment features.
- OutScene [49]: This image dataset contains 2,688 instances categorized into eight classes. It includes 512-D GIST features, 59-D LBP features, 864-D HOG features, and 254-D GENT features.
- 100leaves [45]: This dataset contains 1,600 plant species categorized into 100 classes. Each plant species is characterized by three extracted visual features: 64-D shape descriptor features, 64-D fine scale margin features, and 64-D texture histogram features.
- MNIST [50]: It comprises 10,000 images from 10 classes and includes three commonly used features: 30-D Isoprojection, 9-D Linear Discriminant Analysis, and 9-D Neighborhood Preserving Embedding features.
- Scene15 [51]: This scene image dataset comprises 4,485 images categorized into 15 different categories, with three perspectives captured for each image. The feature dimensions for each perspective are 1,800, 1,180, and 1,240, respectively.
- NoisyMNIST [52]: It is comprised of randomly selected 15,000 samples from the MNIST image database in 10 classes. Therein, the given images come with white Gaussian noise of varied intensities.

We describe the compared algorithms and the experimental setup. The following algorithms are used for testing, including MVAR [48], WREG [53], HLR-M^{2}VS [54], ERL-MVSC [55] Co-GCN [56], DSRL [57], LGCN-FF [43], IMvGCN [44] and JFGCN [58]. A description of these methods is given below.

- MVAR: This framework utilizes the $\ell_{2,1}$-norm to compute the regression loss for each independent view. The trade-off weight for each view is set as $\lambda = 1e^3$, and the redistribution parameter r is fixed at 2.
- WREG: It is a framework that maps the concatenation of raw features onto a discriminative low-dimensional subspace to integrate multi-view data. We select the trade-off λ in $\{1e^{-3}, 1e^{-2}, 1e^{-1}, 1e^{0}, 1e^{1}, 1e^{2}, 1e^{3}\}$ and set the termination parameter $\epsilon = 1e^{-3}$.
- HLR-M^{2}VS: The framework constructs a unified tensor space to jointly explore the relationships among multiple views using a local geometric structure. We select the weighted factors as $\lambda_1 = 0.2$ and $\lambda_2 = 0.4$.
- ERL-MVSC: The framework integrates diversity, sparsity, and consensus to deftly handle multi-view data with limited labels. We set the smoothing factor $\alpha = 2$, the embedding parameter $\beta = 1$, the regularization parameter $\gamma = 1$, and the fitting coefficient $\delta = 10$.
- Co-GCN: The method introduces GCN into multi-view learning and obtains multi-view spectral information by adaptively combining Laplacian matrices. The settings for the graph convolutional layers are 2 and the number of neighbors is 10.
- DSRL: The framework uses a deep sparse regularizer learning model to adaptively learn data-driven sparse regularizers for multi-view clustering and semi-supervised classification. The number of layers is fixed at 10.
- LGCN-FF: The framework considers a joint neural network of both feature and graph fusion. The default setting for hyperparameters controlling the sparsity penalty degree is $\beta = 1$.
- IMvGCN: The framework introduces multi-view reconstruction errors paired with Laplace embeddings to capture independence and consistency. The default setting for hyperparameters $\lambda = 0.5$ and $\alpha = 1e^{-5}$.
- JFGCN: It is an end-to-end joint fusion framework designed to simultaneously execute consistent feature fusion and adaptive topology adjustment. The default setting for the hyperparameter is $\lambda = 1$.

In the proposed method, the shared representation $\mathbf{Z}$ is randomly initialized with a dimension of 100. The features of each view are reconstructed from the shared representation through a series of fully connected networks with the number of neurons set to (100, 1024, d_v), where d_v represents the dimension of the v-th view feature.

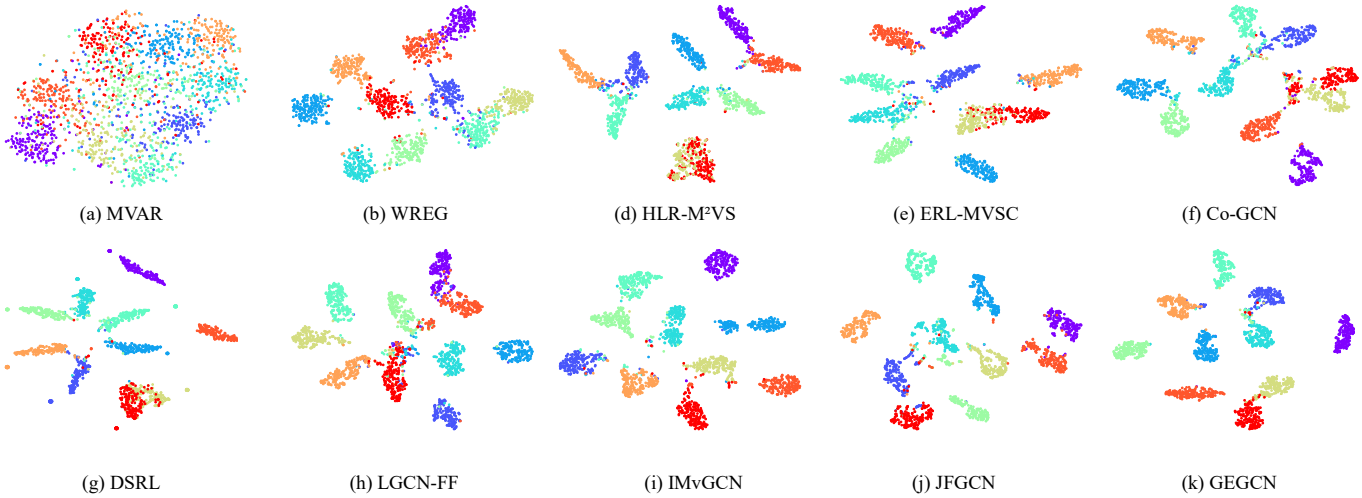
For the samples employed as labels, we independently selected 10% of the data for each category randomly. In

TABLE III

CLASSIFICATION RESULTS (MEAN% AND STANDARD DEVIATION%) OF ALL COMPARED SEMI-SUPERVISED CLASSIFICATION METHODS WITH 10% LABELED SAMPLES AS SUPERVISION, WHERE THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE SECOND-BEST RESULTS ARE HIGHLIGHTED IN UNDERLINED.

Dataset	Metrics	MVAR	WREG	HLR-M²VS	ERL-MVSC	Co-GCN	DSRL	LGCN-FF	IMvGCN	JFGCN	GEGCN-S	GEGCN
100leaves	ACC	39.1 (2.4)	67.7 (5.9)	54.4 (2.2)	58.1 (0.1)	30.3 (5.0)	67.9 (1.6)	62.1 (0.6)	85.0 (1.1)	81.6 (0.9)	88.4 (0.1)	**91.1 (0.5)**
	F1	40.1 (3.1)	61.3 (6.3)	52.8 (2.1)	57.7 (0.1)	29.1 (4.9)	64.4 (1.8)	61.7 (0.5)	84.4 (1.1)	81.4 (1.1)	87.8 (0.1)	**91.0 (0.5)**
HW	ACC	77.8 (2.6)	89.1 (0.4)	85.3 (0.0)	87.0 (0.4)	91.6 (2.7)	77.9 (0.9)	92.6 (0.1)	93.4 (0.8)	93.4 (2.5)	94.8 (0.2)	**96.0 (0.2)**
	F1	78.0 (2.5)	89.3 (0.3)	86.9 (0.0)	87.5 (0.3)	91.5 (2.8)	78.7 (0.7)	92.5 (0.2)	93.4 (0.9)	93.4 (2.5)	94.8 (0.3)	**96.0 (0.2)**
MNIST	ACC	84.4 (0.3)	88.9 (0.5)	84.9 (0.2)	91.7 (0.1)	90.7 (0.5)	89.2 (0.0)	88.8 (0.1)	88.3 (1.1)	90.0 (0.6)	93.3 (0.1)	**93.6 (0.1)**
	F1	84.6 (0.3)	88.8 (0.5)	83.2 (0.1)	91.6 (0.1)	90.6 (0.5)	89.5 (0.0)	86.9 (0.1)	88.1 (1.1)	89.9 (0.6)	93.2 (0.2)	**93.5 (0.1)**
GRAZ02	ACC	52.5 (1.8)	43.4 (3.5)	54.7 (2.6)	54.1 (1.3)	40.5 (2.6)	48.1 (1.0)	49.6 (2.5)	56.2 (0.5)	57.8 (1.2)	61.6 (0.5)	**61.6 (0.5)**
	F1	52.9 (2.0)	43.6 (3.6)	56.3 (1.8)	54.4 (1.2)	38.9 (1.5)	48.6 (1.0)	43.7 (1.0)	55.0 (0.6)	57.8 (1.2)	61.5 (0.3)	**61.5 (0.3)**
Scene15	ACC	44.3 (9.7)	52.3 (2.0)	67.4 (1.3)	63.1 (1.2)	58.7 (1.1)	61.8 (0.9)	50.1 (4.4)	65.6 (3.0)	72.2 (0.6)	71.8 (0.3)	**72.9 (0.4)**
	F1	45.8 (8.4)	52.6 (2.0)	67.3 (0.9)	63.9 (1.3)	56.7 (0.9)	60.5 (0.8)	42.3 (5.7)	62.0 (2.9)	70.7 (0.6)	70.1 (0.3)	**71.8 (0.5)**
OutScene	ACC	46.1 (11.3)	57.6 (1.8)	73.3 (1.3)	68.8 (1.4)	71.0 (2.1)	44.7 (0.8)	61.1 (11.0)	77.2 (0.7)	**79.3 (0.5)**	77.6 (0.3)	78.3 (0.6)
	F1	50.8 (11.1)	58.6 (1.6)	75.2 (1.2)	69.2 (1.4)	71.3 (2.0)	42.1 (2.9)	57.9 (15.6)	77.4 (0.8)	**79.4 (0.5)**	77.9 (0.3)	78.4 (0.7)
NoisyMNIST	ACC	70.1 (0.5)	76.5 (0.4)	OM	OM	83.6 (1.2)	91.6 (0.2)	70.7 (3.3)	88.6 (0.1)	87.8 (0.6)	95.7 (0.2)	**96.3 (0.3)**
	F1	69.4 (0.5)	76.1 (0.3)	OM	OM	83.0 (1.4)	91.4 (0.2)	70.2 (3.2)	88.2 (0.1)	87.6 (0.6)	95.0 (0.2)	**96.2 (0.2)**

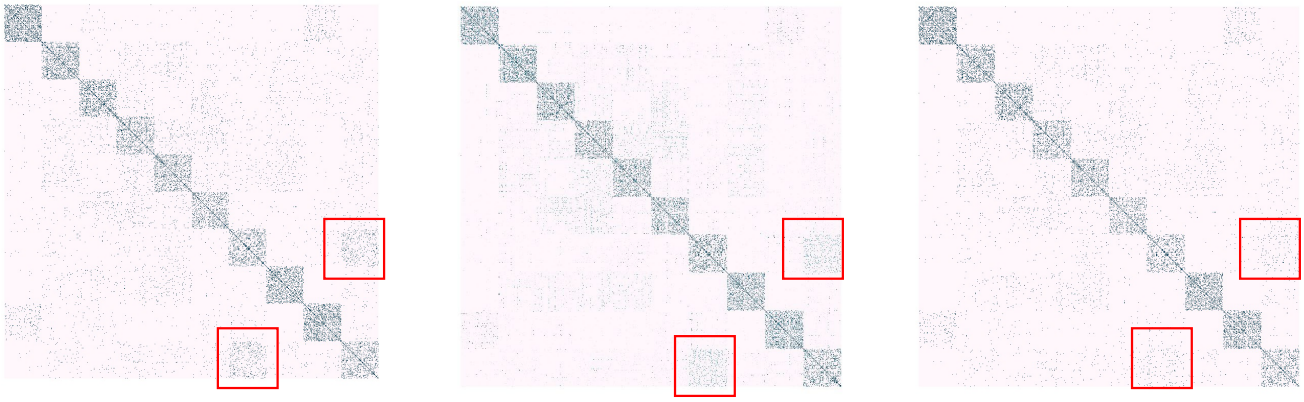
Limited by the computational complexity of the algorithm and machine resources, some models encounter out-of-memory errors when processing the NoisyMNIST dataset, indicated by "OM".



(a) MVAR (b) WREG (d) HLR-M²VS (e) ERL-MVSC (f) Co-GCN

(g) DSRL (h) LGCN-FF (i) IMvGCN (j) JFGCN (k) GEGCN

Fig. 3. T-SNE visualization of semi-supervised classification results from the compared methods on the HW dataset.



(a) Average (b) LGCN-FF (c) GEGCN

Fig. 4. Visualization of the weighted average adjacency matrix, the adjacency matrix obtained by LGCN-FF, and the adjacency matrix learned by GEGCN. The matrices are represented using colors, with darker shades indicating higher element values. Notably, red boxes are used to emphasize reduced or vanishing node connections.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3374579

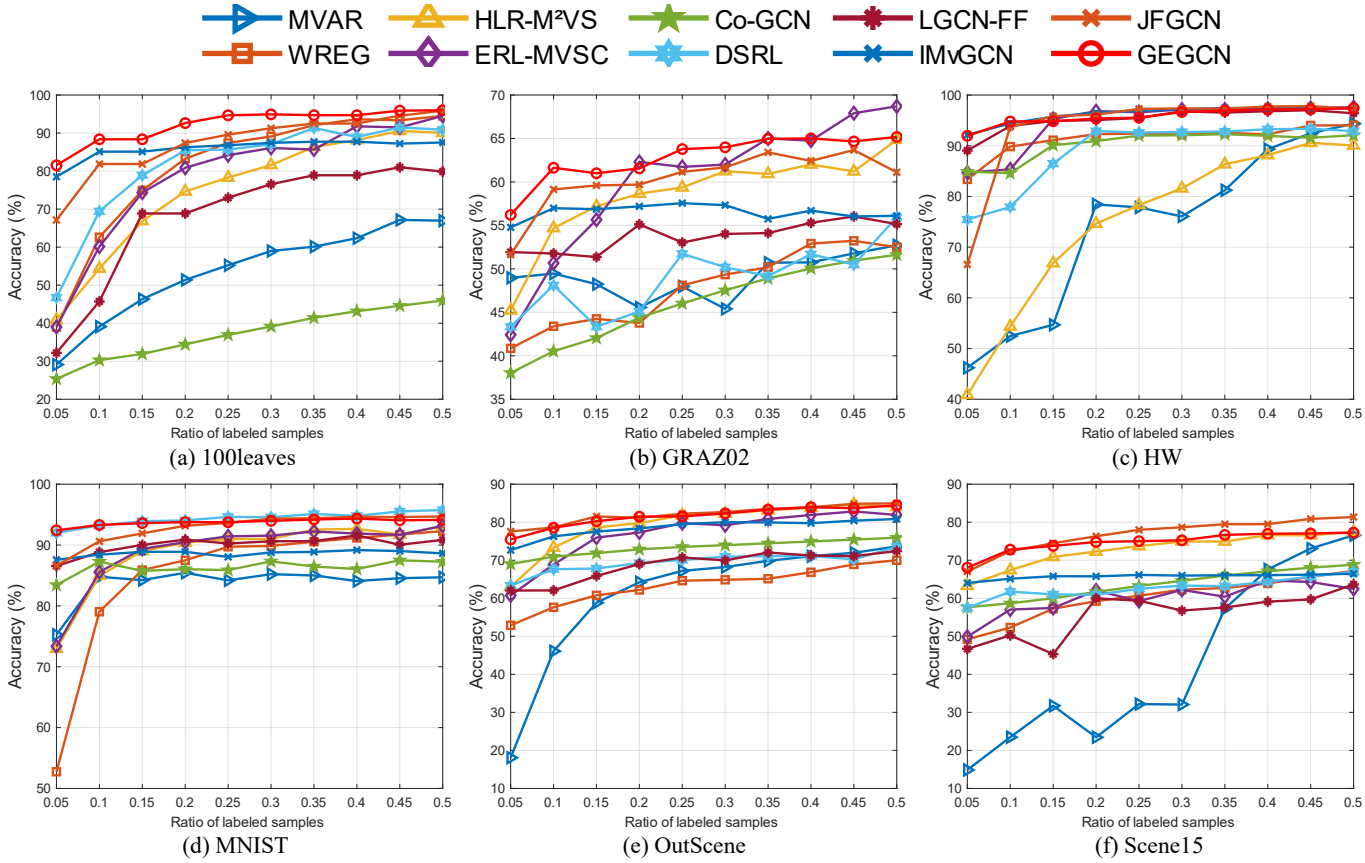


Fig. 5. The performance (Accuracy %) of all compared methods on six test datasets with the ratio of labeled samples ranging from 0.05 to 0.50.

instances where labeled samples fail to yield integer values after calculation, we applied upward rounding. For the graph generator, we utilize an MLP with the number of neurons fixed as $(200, 32, 1)$. The graph discriminator uses a randomly initialized learnable weight matrix, denoted as $\mathbf{W}_d$. We employ a two-layer GCN with an Adam optimizer, setting the learning rate as $lr = 0.01$ and the weight decay as $5e^{-4}$. The used activation function is the ReLU function. The gap between θ_1 and θ_2 is set as $\epsilon = 0.05$. For the kNN algorithm utilized to construct the adjacency matrices, the hyperparameter k is varied within a range from 5 to 15. Additionally, we establish a simplified version of GEGCN, denoted as GEGCN-S, where the optimization is solely based on the loss function $\mathcal{L}_{re}$. In this version, the shared representation can be precomputed and is not involved in the model's optimization process. The code has been uploaded to Github[1]

B. Multi-view Semi-supervised Classification

In this subsection, we performed experiments on seven real-world datasets using 10% labeled samples for supervision. The used evaluation metrics are classification accuracy and F1-score, and the results are presented in Table III. The experimental results demonstrate that the proposed approach outperforms most other methods on all the datasets, indicating its superiority in terms of classification performance. Notably, for the GRAZ02 and 100leaves datasets, GEGCN exhibits remarkable improvements, showcasing its capability to effectively capture correlations among multigraphs derived from small datasets.

To provide a visual representation of the classification performance of GEGCN, we acquire the representations of all methods and project them onto a two-dimensional space utilizing t-distributed random neighborhood embedding (t-SNE), respectively. Subsequently, color codes are assigned to the mapped 2D data on the ground truth. In Fig. 3, the consistency of our proposed approach with the ground truth class labels, along with its ability to generate clear inter-class partition lines, further strengthens the validity and superiority of our model.

Fig. 4 presents visualizations of the graphs learned through different approaches: the weighted average of the adjacency matrix, the graph derived from LGCN-FF, and GEGCN. Notably, the adjacency matrix learned by the proposed method demonstrates a distinctively pure and refined structure. The improved performance of GEGCN can be attributed to its utilization of downstream label information for learning and optimization, combined with the use of LPReLU. This process selectively eliminates small connections between samples of different categories while preserving connections among samples of the same category. Consequently, the resulting adjacency matrix becomes sparser and more robust, leading to an enhancement in the graph structure.

[1]https://github.com/long319/GEGCN.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3374579

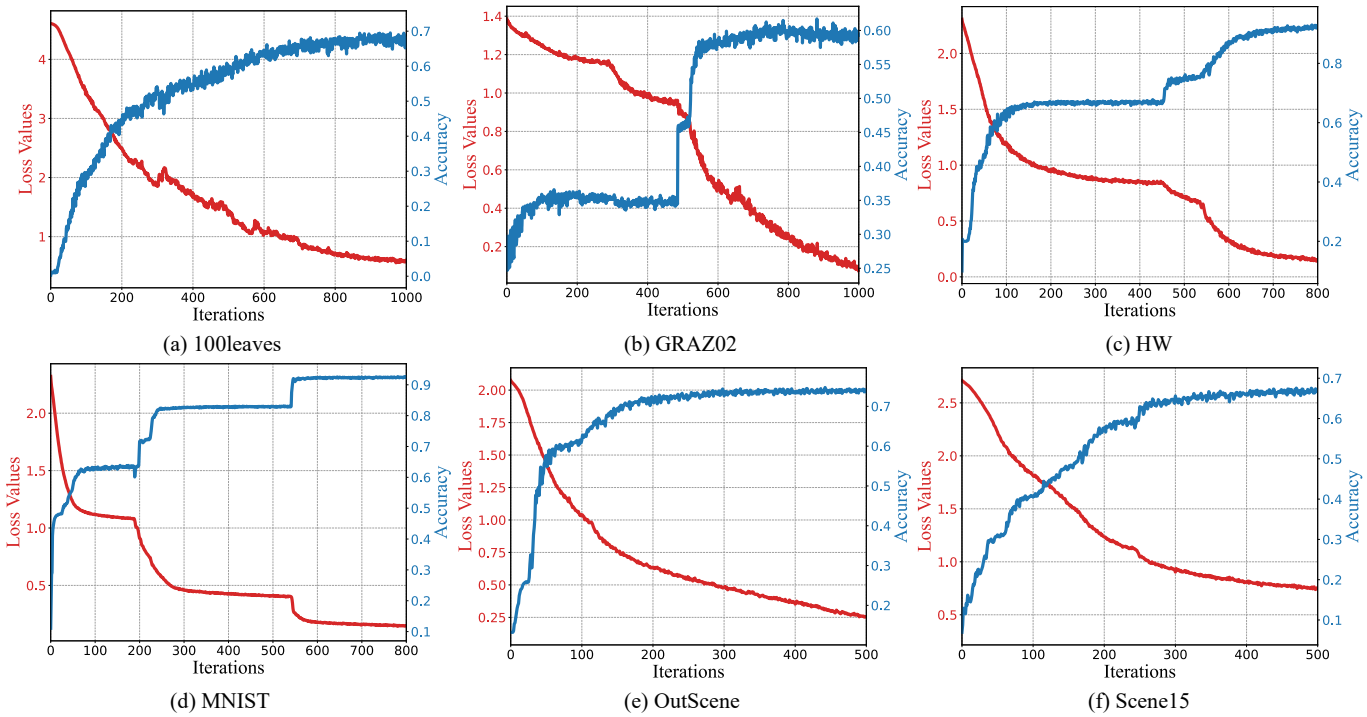


Fig. 6. The convergence curves of training loss values and test accuracy with GEGCN on six datasets.

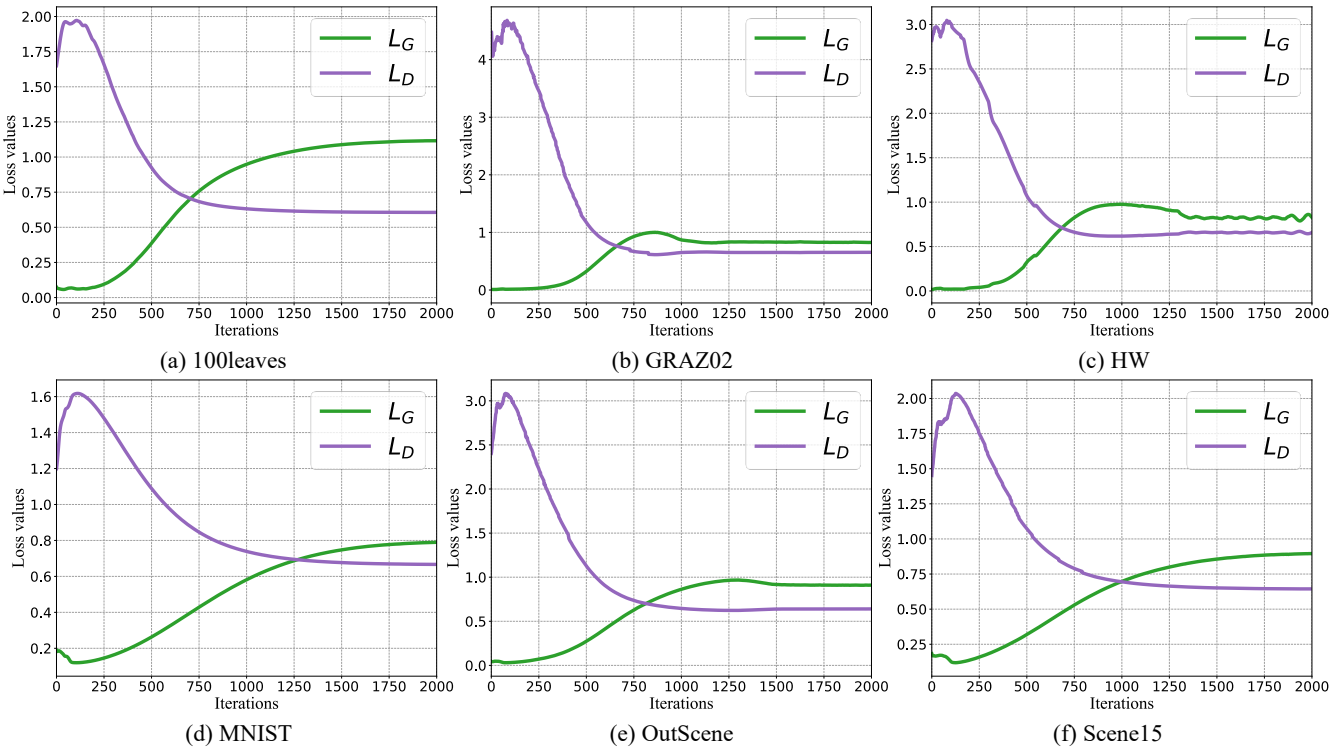


Fig. 7. The convergence curves of generator loss values and discriminator loss values with GEGCN on six datasets.

Additionally, Fig. 5 presents the performance of each comparative algorithm under varying labeled sample rates. The findings from the figure indicate that GEGCN is especially suitable for semi-supervised classification tasks, as it exhibits commendable performance even when confronted with limited sample label rates. Conversely, the alternative methods necessitate a substantially larger amount of supervised information to achieve comparable levels of performance. To assess the statistical significance of the experimental results at high supervision ratios, we employ the Friedman test [59]. Table IV displays the p-value for the traditional methods, the network-based methods, and all methods across six test datasets. Notably, all p-values at high supervision ratios are below 0.05. Consequently, we reject the null hypothesis, indicating that the performance of the compared methods at high ratios is significantly different at a confidence level of 95%.

TABLE IV
P-VALUE FOR TRADITIONAL ALGORITHMS (MVAR, WREG, HLR-M²VS, AND ERL-MVSC), NETWORK-BASED ALGORITHMS (CO-GCN, DSRL, LGCN-FF, IMVGCN, JFGCN, AND GEGCN), AND ALL ALGORITHMS ON SIX DATASETS.

Methods/Ratio	0.40	0.45	0.50
Traditional Methods	0.018	0.013	0.030
Network-based Methods	0.003	0.003	0.003
All Methods	$1.3e^{-5}$	$1.4e^{-5}$	$1.6e^{-5}$

C. Convergence Analysis

In this subsection, we provide a comprehensive analysis of the convergence behavior of GEGCN across all datasets. Fig. 6 showcases the convergence curves of cross-entropy loss and test accuracy, offering insights into the training process. From the figure, it can be observed that the loss and accuracy stabilize and converge after approximately 500 training iterations for all datasets. Notably, the loss curve for the MNIST dataset exhibits a steep decline during the training process, which may be attributed to the large number of samples and the presence of noisy connections in the learned graph. Once the threshold shrinkage function adjusts the threshold to an appropriate point, it effectively removes unreasonable connections and improves performance. Additionally, by gradually adjusting the threshold until the next equilibrium point is reached, larger noise is filtered out, leading to further enhancements in performance.

Fig. 7 presents the convergence curves for the generator and discriminator losses. From the figure, it can be observed that the loss of the discriminator gradually decreases over time, while the loss of the generator gradually increases. Eventually, both losses reach a point of stability or convergence. Both losses stabilize and converge after approximately $1,000$ training iterations. However, the proposed method requires a larger number of training iterations on the MNIST dataset due to its larger sample size.

TABLE V
ABLATION STUDY OF GEGCN ON SEVEN DATASETS WITH ACC (%) AND F1 (%).

Dataset/Method	Metric	G-WLT	G-WL	G-WT	GEGCN
GRAZ02	ACC	49.96	60.04	60.79	**61.63**
MNIST		87.77	87.91	93.07	**93.30**
HW		75.77	75.83	91.61	**94.78**
OutScene		70.40	70.98	76.10	**77.55**
100leaves		66.21	75.85	86.64	**88.36**
Scene15		62.55	63.74	67.23	**71.80**
NoisyMNIST		82.14	82.20	91.34	**95.70**
GRAZ02	F1	43.06	59.88	61.16	**61.54**
MNIST		87.61	87.74	92.98	**93.20**
HW		75.62	75.69	91.63	**94.78**
OutScene		67.05	67.60	76.56	**77.86**
100leaves		57.88	69.94	86.15	**87.78**
Scene15		56.71	59.74	65.55	**70.06**
NoisyMNIST		81.69	81.75	91.15	**95.00**

D. Ablation Study

To showcase the impact of the learned graph and the threshold shrinkage function, additional tests were conducted. These evaluations assessed the performance under various conditions, namely without utilizing the learned graph (referred to as G-WL), without applying the threshold shrinkage function (referred to as G-WT), and without employing both (referred to as G-WTL). The experimental results are presented in Table V. The results clearly show that the learned graph effectively captures the connectivity among the samples, surpassing the graphs derived from the original space. This enhancement is particularly noticeable in the 100leaves dataset which comprises multiple categories, where the graphs in the original space may encompass unreliable connections. Through the learned graph, we can identify connections that more accurately represent the data, thereby adapting to downstream tasks. Upon comparing the performance of G-WT and GEGCN, it is apparent that the performance of GEGCN is further enhanced, underscoring the effectiveness of the proposed learnable threshold shrinkage function.

E. Parameter Sensitivity

In this subsection, we conduct a parameter sensitivity analysis of our proposed method for all datasets using accuracy and F1-score as evaluation metrics. The impact of the parameter k is presented in the accompanying Fig. 8. The results indicate that the proposed model remains stable across most of the datasets and achieves optimal performance when the hyperparameter k is set to 10. Interestingly, we observe a peculiar phenomenon in the 100leaves dataset where the performance gradually decreases as k increases. Building on this observation, we provide the average homophilic rate of the constructed graph, as illustrated in Table VI. Notably, it is observed that the homophilic rate of the dataset 100leaves experiences a marked decrease as the value of k increases, contributing to the phenomenon of performance degradation. Simultaneously, we conduct a sensitivity analysis on the hyperparameter α, with the results depicted in Fig. 9. It is evident from the figure that $\alpha = 1$ or 0.1 consistently yields satisfactory results across

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2024.3374579

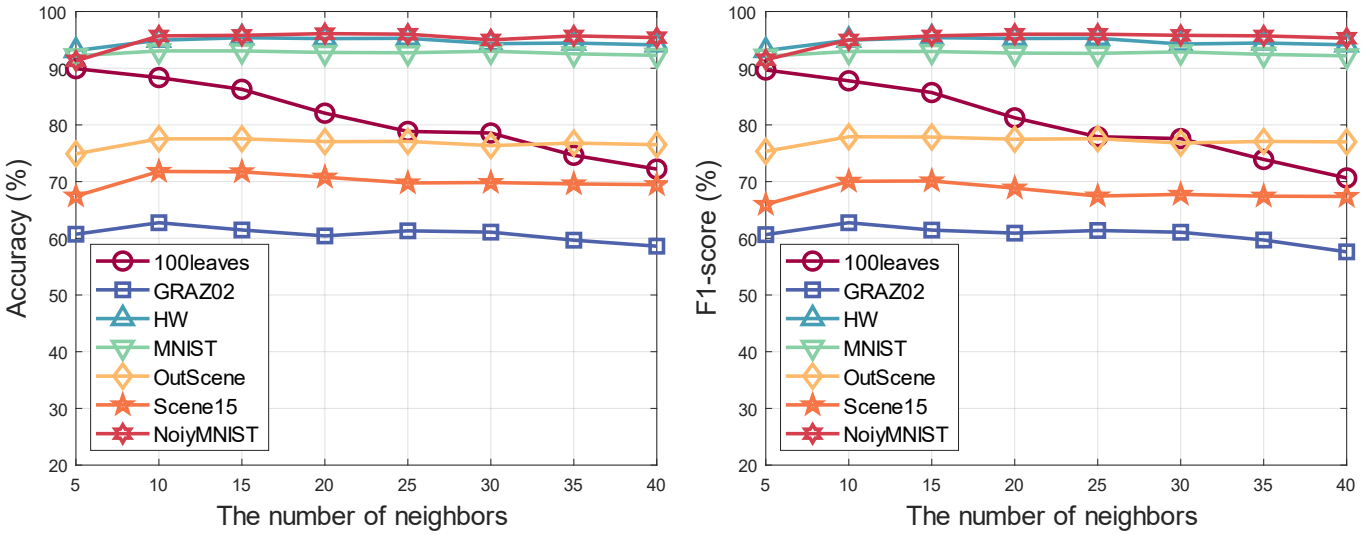


Fig. 8. The sensitivity of the proposed method's parameters, in terms of Accuracy (%) and F1-score (%), with respect to the number of neighbors is evaluated on all datasets.

the majority of datasets. Additionally, our experimental results indicate that the proposed framework consistently outperforms state-of-the-art algorithms, even without requiring specific parameter tuning. These findings highlight the effectiveness and competitiveness of our approach in addressing multi-view semi-supervised classification tasks.

TABLE VI
AVERAGE HOMOPHILIC RATE (%) OF THE DATASETS FOR DIFFERENT VALUES OF k.

Dataset/k	5	10	15	20	25	30	35	40
100leaves	61.6	44.7	35.9	29.7	25.5	22.4	20.0	18.1
GRAZ02	46.6	42.1	40.3	39.3	38.5	37.9	37.5	37.0
HW	74.7	68.0	64.0	61.1	58.7	56.8	55.1	53.8
MNIST	94.7	91.4	89.3	88.0	87.0	86.2	85.6	85.0
OutScene	57.5	52.0	49.2	47.6	46.4	45.4	44.5	43.8
Scene15	59.5	51.3	47.1	44.7	43.0	41.7	40.6	39.7
NoisyMNIST	77.4	75.7	72.9	70.4	68.3	66.8	65.5	64.4

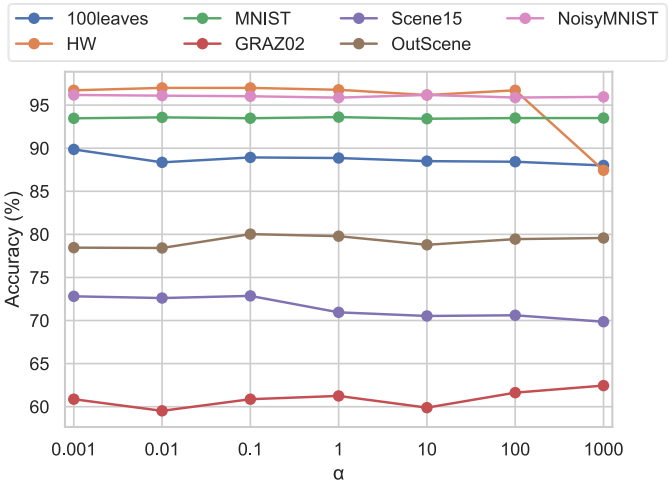


Fig. 9. Parameter sensitivity (Accuracy %) of the proposed method w.r.t. α on all datasets.

V. CONCLUSION

This paper introduces an end-to-end framework called GEGCN which aims to enhance the performance of graph embeddings by incorporating view consistency, complementarity, and downstream task information in graph construction. The framework begins by obtaining a shared representation, which is then transformed into a graph with consistency. This process gradually integrates the specificity of each view graph and combines it with information from downstream tasks. To enhance the reliability and sparsity of the learned graphs, we propose a denoised GCN with a learnable threshold shrinkage function. This function dynamically adjusts the threshold to effectively filter out noisy connections, resulting in a purer graph. The final step involves obtaining robust graph embeddings using the denoised GCN. Experimental results on various datasets showcase the effectiveness of the proposed framework.

This study proposes several intriguing potential research directions that warrant further exploration. Currently, GCN-based multi-view algorithms primarily focus on investigating the consistency and complementarity of the feature space. These approaches typically rely on the kNN algorithm for constructing multi-view data topology with notable limitations, particularly concerning the high heterogeneity of the resulting graphs. Exploring novel approaches to measuring inter-sample similarity represents a promising direction for enhancing the performance of GCN on multi-view data. In future research, we intend to extend the application of GEGCN beyond multi-view data to various other domains.

REFERENCES

[1] Y. Chen, X. Xiao, and Y. Zhou, "Jointly learning kernel representation tensor and affinity matrix for multi-view clustering," *IEEE Transactions on Multimedia*, vol. 22, pp. 1985–1997, 2019.

[2] M.-S. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 3513–3520, 2020.

[3] W. Xia, Q. Wang, Q. Gao, X. Zhang, and X. Gao, "Self-supervised graph convolutional network for multi-view clustering," *IEEE Transactions on Multimedia*, vol. 24, pp. 3182–3192, 2021.

[4] Z. Lu, F. Nie, R. Wang, and X. Li, "A differentiable perspective for multi-view spectral clustering with flexible extension," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 7087–7098, 2023.

[5] X. Wan, J. Liu, W. Liang, X. Liu, Y. Wen, and E. Zhu, "Continual multi-view clustering," in *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3676–3684, 2022.

[6] Y. Sun, Z. Ren, P. Hu, D. Peng, and X. Wang, "Hierarchical consensus hashing for cross-modal retrieval," *IEEE Transactions on Multimedia*, vol. 26, pp. 824–836, 2024.

[7] L. Fu, Z. Chen, Y. Chen, and S. Wang, "Unified low-rank tensor learning and spectral embedding for multi-view subspace clustering," *IEEE Transactions on Multimedia*, pp. 1–14, 2022.

[8] X. Shen, Y. Tang, Y. Zheng, Y.-H. Yuan, and Q.-S. Sun, "Unsupervised multiview distributed hashing for large-scale retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8837–8848, 2022.

[9] Z. Zhang, X. Han, X. Song, Y. Yan, and L. Nie, "Multi-modal interaction graph convolutional network for temporal language localization in videos," *IEEE Transactions on Image Processing*, vol. 30, pp. 8265–8277, 2021.

[10] X. Wan, X. Liu, J. Liu, S. Wang, Y. Wen, W. Liang, E. Zhu, Z. Liu, and L. Zhou, "Auto-weighted multi-view clustering for large-scale data," in *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pp. 10078–10086, 2023.

[11] X. Li, Y. Sun, Q. Sun, Z. Ren, and Y. Sun, "Cross-view graph matching guided anchor alignment for incomplete multi-view clustering," *Information Fusion*, vol. 100, p. 101941, 2023.

[12] C. Zhang, J. Cheng, and Q. Tian, "Multi-view image classification with visual, semantic and view consistency," *IEEE Transactions on Image Processing*, vol. 29, pp. 617–627, 2019.

[13] F. Wu, X. Jing, X. You, D. Yue, R. Hu, and J. Yang, "Multi-view low-rank dictionary learning for image classification," *Pattern Recognition*, vol. 50, pp. 143–154, 2016.

[14] X. Shen, G. Dong, Y. Zheng, L. Lan, I. W. Tsang, and Q.-S. Sun, "Deep co-image-label hashing for multi-label image retrieval," *IEEE Transactions on Multimedia*, vol. 24, pp. 1116–1126, 2021.

[15] B. Zhang, Q. Qiang, F. Wang, and F. Nie, "Fast multi-view semi-supervised learning with learned graph," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, pp. 286–299, 2022.

[16] X.-l. Wang, Z.-f. Zhu, Y. Song, and H.-j. Fu, "Grnet: Graph-based remodeling network for multi-view semi-supervised classification," *Pattern Recognition Letters*, vol. 151, pp. 95–102, 2021.

[17] G. Chao and S. Sun, "Semi-supervised multi-view maximum entropy discrimination with expectation laplacian regularization," *Information Fusion*, vol. 45, pp. 296–306, 2019.

[18] X. Jia, X.-Y. Jing, X. Zhu, S. Chen, B. Du, Z. Cai, Z. He, and D. Yue, "Semi-supervised multi-view deep discriminant representation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 2496–2509, 2020.

[19] Z. Wu, Z. Zhang, and J. Fan, "Graph convolutional kernel machine versus graph convolutional networks," in *Proceedings of 37th Conference on Neural Information Processing Systems*, 2023.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations*, pp. 1–13, 2017.

[21] J. Chen, H. He, F. Wu, and J. Wang, "Topology-aware correlations between relations for inductive link prediction in knowledge graphs," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 6271–6278, 2021.

[22] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang, "Neural bellman-ford networks: A general graph neural network framework for link prediction," in *Proceedings of the 34th Neural Information Processing Systems*, pp. 29476–29490, 2021.

[23] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proceedings of the 31st Neural Information Processing Systems*, pp. 5171–5181, 2018.

[24] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 7444–7452, 2018.

[25] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3595–3603, 2019.

[26] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning graph convolutional network for skeleton-based human action recognition by neural searching," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 2669–2676, 2020.

[27] J. Chang, C. Gao, X. He, D. Jin, and Y. Li, "Bundle recommendation with graph convolutional networks," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1673–1676, 2020.

[28] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 639–648, 2020.

[29] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 165–174, 2019.

[30] S. Huang, S. Xiao, W. Liu, J. Lu, Z. Wu, S. Wang, and J. C. Rajapakse, "Multi-level knowledge integration with graph convolutional network for cancer molecular subtype classification," in *Proceedings of the International Conference on Bioinformatics and Biomedicine*, pp. 1983–1988, 2023.

[31] Z. Chen, Z. Wu, Z. Lin, S. Wang, C. Plant, and W. Guo, "Agnn: Alternating graph-regularized neural networks to alleviate over-smoothing," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2023.

[32] Z. Chen, Z. Wu, S. Wang, and W. Guo, "Dual low-rank graph autoencoder for semantic and topological networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 4191–4198, 2023.

[33] P. Han, P. Yang, P. Zhao, S. Shang, Y. Liu, J. Zhou, X. Gao, and P. Kalnis, "Gcn-mf: disease-gene association identification by graph convolutional networks and matrix factorization," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 705–713, 2019.

[34] J. Gasteiger, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," in *Proceedings of the 32nd Neural Information Processing Systems*, pp. 13333–13345, 2019.

[35] L. Zhong, J. Yang, Z. Chen, and S. Wang, "Contrastive graph convolutional networks with generative adjacency matrix," *IEEE Transactions on Signal Processing*, vol. 71, pp. 772–785, 2023.

[36] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proceedings of the 37th International Conference on Machine Learning*, pp. 1725–1735, 2020.

[37] S. Sun and G. Chao, "Multi-view maximum entropy discrimination," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 1706–1712, 2013.

[38] G. Chao and S. Sun, "Consensus and complementarity based maximum entropy discrimination for multi-view classification," *Information Sciences*, vol. 367, pp. 296–310, 2016.

[39] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "Comic: Multi-view clustering without parameter selection," in *Proceedings of the 36 th International Conference on Machine Learning*, pp. 5092–5101, 2019.

[40] Y. Lin, Y. Gou, X. Liu, J. Bai, J. Lv, and X. Peng, "Dual contrastive prediction for incomplete multi-view representation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 4447–4461, 2023.

[41] Y. Lin, Y. Gou, X. Liu, J. Bai, J. Lv, and X. Peng, "Dual contrastive prediction for incomplete multi-view representation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4447–4461, 2022.

[42] M. Yang, Y. Li, P. Hu, J. Bai, J. Lv, and X. Peng, "Robust multi-view clustering with incomplete information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1055–1069, 2022.

[43] Z. Chen, L. Fu, J. Yao, W. Guo, C. Plant, and S. Wang, "Learnable graph convolutional network and feature fusion for multi-view learning," *Information Fusion*, vol. 95, pp. 109–119, 2023.

[44] Z. Wu, X. Lin, Z. Lin, Z. Chen, Y. Bai, and S. Wang, "Interpretable graph convolutional network for multi-view semi-supervised learning," *IEEE Transactions on Multimedia*, pp. 1–14, 2023.

[45] C. Tang, X. Liu, X. Zhu, E. Zhu, Z. Luo, L. Wang, and W. Gao, "Cgd: Multi-view clustering via cross-view graph diffusion," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 5924–5931, 2020.

[46] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Consensus graph learning for multi-view clustering," *IEEE Transactions on Multimedia*, vol. 24, pp. 2461–2472, 2021.

[47] J. Wu, X. Xie, L. Nie, Z. Lin, and H. Zha, "Unified graph and low-rank tensor learning for multi-view clustering," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 6388–6395, 2020.
[48] H. Tao, C. Hou, F. Nie, J. Zhu, and D. Yi, "Scalable multi-view semi-supervised classification via adaptive regression," *IEEE Transactions on Image Processing*, vol. 26, pp. 4283–4296, 2017.
[49] S. El Hajjar, F. Dornaika, and F. Abdallah, "Multi-view spectral clustering via constrained nonnegative embedding," *Information Fusion*, vol. 78, pp. 209–217, 2022.
[50] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4412–4419, 2020.
[51] Z. Niu, G. Hua, X. Gao, and Q. Tian, "Context aware topic model for scene recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 2743–2750, 2012.
[52] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1083–1092, 2015.
[53] M. Yang, C. Deng, and F. Nie, "Adaptive-weighting discriminative regression for multi-view classification," *Pattern Recognition*, vol. 88, pp. 236–245, 2019.
[54] Y. Xie, W. Zhang, Y. Qu, L. Dai, and D. Tao, "Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning," *IEEE Transactions on Cybernetics*, vol. 50, pp. 572–586, 2020.
[55] A. Huang, Z. Wang, Y. Zheng, T. Zhao, and C.-W. Lin, "Embedding regularizer learning for multi-view semi-supervised classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 6997–7011, 2021.
[56] S. Li, W. Li, and W. Wang, "Co-gcn for multi-view semi-supervised learning," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 4691–4698, 2020.
[57] S. Wang, Z. Chen, S. Du, and Z. Lin, "Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 5042–5055, 2022.
[58] Y. Chen, Z. Wu, Z. Chen, M. Dong, and S. Wang, "Joint learning of feature and topology for multi-view graph convolutional network," *Neural Networks*, vol. 168, pp. 161–170, 2023.
[59] M. Shang, Y. Yuan, X. Luo, and M. Zhou, "An α–β-divergence-generalized recommender for highly accurate predictions of missing user preferences," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 8006–8018, 2021.

Jielong Lu received his B.S. degree from the College of Mathematics and Statistics, Fuzhou University, Fuzhou, China in 2022. He is currently pursuing the M.S. degree with the College of Computer Science and Data Science, Fuzhou University, Fuzhou, China. His current research interests include machine learning, multi-view learning and graph neural networks.

Zhihao Wu received his B.S. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China in 2021. He is currently pursuing the M.S. degree with the College of Computer and Data Science, Fuzhou University, Fuzhou, China. He is also currently visiting Shenzhen Research Institute of Big Data and Chinese University of Hong Kong (Shenzhen), Shenzhen, China. His current research interests include machine learning, multi-view learning and graph neural networks.

Luying Zhong received her B.S. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China in 2021. She is currently pursuing the Ph.D. degree with the College of Computer and Data Science, Fuzhou University, Fuzhou, China. Her current research interests include edge computing, federated learning, graph learning, and graph neural networks.

Zhaoliang Chen received the B.E. degree in 2019 from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. He is currently working toward the Ph.D. degree with the College of Computer and Data Science, Fuzhou University. From 2022 to 2023, he was a Visiting Researcher with the Faculty of Computer Science, University of Vienna, Vienna, Austria.. His research interests include machine learning, deep learning, graph neural networks, and recommender systems.

Hong Zhao received the Ph.D. degree from Tianjin University, Tianjin, China, in 2019. She received her M.S. degree from Liaoning Normal University, Dalian, China, in 2006. She is currently a Professor of the School of Computer Science and the Key Laboratory of Data Science and Intelligence Application, Minnan Normal University, Zhangzhou, China. Her current research interests include rough sets, granular computing, and data mining for hierarchical classification.

Shiping Wang is currently working as a Professor with the College of Computer and Data Science, and the director of the Fujian Provincial Key Laboratory of Intelligent Metro, Fuzhou University, Fuzhou, China. He received his Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China in 2014. He was a Visiting Scholar in University of Alberta, Edmonton, Canada from August 2013 to August 2014. He worked as a Research Assistant in National University of Singapore from January 2014 to August 2014, and a Research Fellow in Nanyang Technological University of Singapore from August 2015 to August 2016. He was also a Visiting Researcher in Peking University, Beijing, China from August 2019 to August 2020. His research interests include machine learning, deep learning, feature representation and multi-modal fusion.