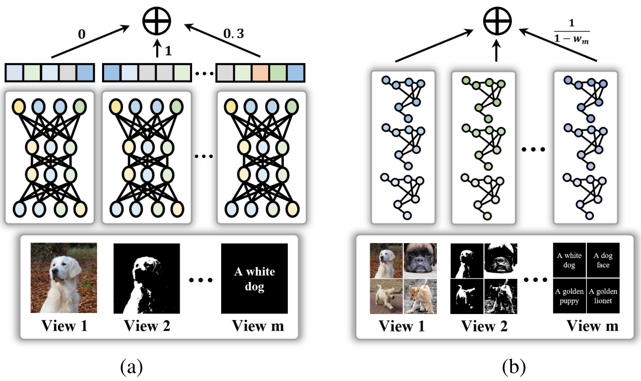
Deep Graph Multi-View Representation Learning With Self-Augmented View Fusion

Ziheng Jiao, Hongyuan Zhang, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Some current researchers attempt to extend the graph neural network (GNN) on multi-view representation learning and learn the latent structure information among the data. Generally, they concatenate the features of each view and employ a single GNN to extract the representations of this concatenated feature. It causes that the within-view information may not be learned and the pivotal view will not be strengthened during the concatenation. Although some GNN models introduce the Siamese structure to extract the within-view information, the learned representation may not be informative since the Siamese GNNs share the same parameters. To overcome these issues, we propose a novel deep graph auto-encoder for multi-view representation learning. Among them, a self-augmented view-weight technique is theoretically devised for cross-view fusion, which can highlight the pivotal views and maintain the rest views. Then, GNNs of different views can learn the informative representation without sharing parameters. Furthermore, by fitting the fusion distribution with a neural layer, the model unifies these two individual procedures and achieve to extract the fusion representation end-to-end. Compared with numerous recently proposed methods, extensive experiments on clustering and recognition tasks demonstrate our superior performance.**

***Index Terms*— Graph neural network (GNN), multi-view learning, self-augmented view fusion, sparse graph learning.**



(a)

(b)

Fig. 1. Comparison between the standard deep multi-view representation learning and our graph multi-view learning. (a) Standard multi-view learning methods utilize a DNN to extract the deep representation and fuse them by sparse or linear weight. (b) Compared with the standard form, ours extends the GNN on each view to extract the latent structure among the within-view instances. Then, a nonlinear weight is employed to fuse the representation across views, which can highlight the pivotal views while maintaining the complementary information from the insignificant views.

I. Introduction

As a fundamental tool in representation learning, multi-view representation learning attempts to exploit the pivotal or complementary information contained in different views and form a consensus representation, which can facilitate downstream tasks such as classification [1], [2], clustering [3], [4], and retrieval [5]. According to different application scenarios, these methods can be roughly divided into three categories [6], namely, supervised methods [7], semi-supervised methods [8], and unsupervised methods [9], [10]. In this article, we only investigate unsupervised multi-view representation learning, which will learn a consensus representation without introducing the label information. The related methods can be further classified into two categories, i.e., machine learning-based methods and deep learning-based methods.

Machine learning-based methods generally extract the representation by modeling the within-view similarity. Among them, canonical correlation analysis (CCA) utilizes the correlation of the data modalities to project the different views into one common subspace [11]. Besides, Zhang et al. [12] introduces the subspace projection for multi-view learning which mainly learns the specific-view features under the Laplacian matrix. These methods may not perform well on large and nonlinear data due to their linear operation. Benefiting from the powerful nonlinear representation ability, deep neural networks (DNNs) have made great progress on large-scale application scenarios [13], [14], [15], especially for supervised learning [13]. Inspired by these merits, some works design some deep multi-view models to learn the view-specific representation [16], [17]. Although these deep learning-based methods can handle more complex and nonlinear view data, the performance may drop sharply when lacking the label information in unsupervised scenarios. Notably, by utilizing the latent structure among the graph data, graph neural networks (GNNs) [18], [19] can extract meaningful representations and achieve excellent performance under unsupervised

Received 27 February 2024; revised 18 July 2024 and 10 October 2024; accepted 31 December 2024. Date of publication 4 February 2025; date of current version 6 August 2025. *(Corresponding authors: Hongyuan Zhang; Xuelong Li.)*

Ziheng Jiao is with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China, and also with the Institute of Artificial Intelligence (TeleAI), China Telecom, Shanghai 200030, P. R. China (e-mail: jzh9830@163.com).

Hongyuan Zhang is with the Institute of Artificial Intelligence (TeleAI), China Telecom, Shanghai 200030, P. R. China, and also with the Musketeers Foundation Institute of Data Science, The University of Hong Kong, Hong Kong (e-mail: hyzhang98@gmail.com).

Xuelong Li is with the Institute of Artificial Intelligence (TeleAI), China Telecom, Shanghai 200030, P. R. China (e-mail: xuelong_li@ieee.org).

Digital Object Identifier 10.1109/TNNLS.2025.3527928

scenarios. Although some methods have introduced GNN on multi-view datasets by Gaussian kernel [20], [21], they also introduce an extra threshold to control the sparsity, which may take the vast cost to tune and select the correct the proper value. Besides, these methods mainly concatenate the features of each view and employ a single GNN to extract the representations of this concatenated feature. It means the within-view information may not be learned and the pivotal view will not be strengthened during the concatenation. Therefore, there is a natural concern about how to extend GNN on multi-view datasets for learning the informative representation.

Besides, to learn the consensus representation, Zhang et al. [22] directly concatenates all view-specific features. Furthermore, Zheng et al. [23] introduces the $\ell_{2,1}$-norm term on the objective function to extract the feature on the consensus representation. Although these models based on direct concatenation can obtain the consensus representation, they may introduce some redundant information since all features are considered during the fusion and this information even degrades the performance. Wen et al. [24] can distinguish the different views by assigning them with different linear or sparse weights. However, these weight strategies may not strengthen the pivotal views and directly discard the complementary information from some insignificant views. Therefore, another concerned question is how to reinforce the pivotal views and maintain the complementary views simultaneously during fusion.

To obstacle the referred problems, we propose a novel deep graph autoencoder with self-augmented nonlinear weights for multi-view representation learning. Specifically, by learning the specific-view sparse graph structure, we can extend GNN on each view to extract the within-view latent representation without the label information. Then, as shown in Fig. 1, different from the linear or sparse weight in [24], a self-augmented nonlinear weight is designed to fuse the cross-view representation by strengthening the pivotal view and preserving the complementary view simultaneously. Our core contributions are as follows.

1) By learning the sparse graph on each view with the designed strategy, GNNs are successfully extended on the multi-view data to extract the within-view structure and obtain the specific-view representation under the unsupervised scenarios.
2) A self-augmented nonlinear weight is theoretically devised for cross-view fusion, which can reinforce the pivotal views while maintaining the complementary information from the rest views.
3) By fitting the distribution of fusion representation with a neural layer, the model successfully unifies the extraction and fusion stage into a deep differentiable framework, which can be optimized with gradient descent to generate the consensus representation end-to-end.

Notations: In this article, $(\cdot)_+ = \max(0, \cdot)$ and $\mathbf{1}_n = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$. The boldface capital and boldface lowercase letters represent the matrix $\boldsymbol{M}$ and vector $\boldsymbol{m}$, respectively. The ith column is $\boldsymbol{m}_i$ and the element in $\boldsymbol{M}$ is $\boldsymbol{M}_{ij}$.

II. Related Work

In this section, we briefly review several related graph representation learning models and multi-view representation learning methods.

A. Graph Representation Learning

To learn the reliable representation under unsupervised scenarios, graph representation learning concentrates on exploring the distribution among the instances. Among them, manifold learning has been a dominant graph method for a long time and contains many famous models including Laplacian Eigenmaps (LEs) [25], [26], locally linear embedding [27] and principal component analysis [28]. Recently, to improve the representational ability, some researchers have attempted to introduce the neural layer and extract the representation on graph data, such as citation networks and social networks [29]. Defferrard et al. [30] design a Chebyshev convolution kernel by fitting the spectral decomposition with the Chebyshev polynomial term. Then, by simplifying the kernel with the first-order expansion, the classic GNN is formed [18]. Although this form has made great progress in plenty of scenarios [31], it is difficult to extend on nongraph datasets such as images and text due to lacking the adjacent matrix. Although some models utilize the vector inner product or the Euclidean distance to learn the graph [32], [33], they need to introduce an extra threshold to control the sparsity, which is time-consuming to tune and select the proper value. To overcome these problems, we design a sparse graph learning strategy, which can learn a sparse graph on each view. Based on this, our model can learn the latent structure among within-view instances.

B. Multi-View Representation Learning

Multi-view representational learning mainly aims to extract the consensus representation across views and has played a vital role in many realistic scenarios. Among them, LEs are widely used in unsupervised multi-view learning [12]. By introducing the low-rank Laplacian matrix based on a sparse Markove chain, Xia et al. [34] propose a robust multi-view spectral learning to construct the low-rank consensus representation. Meanwhile, some researchers extend graph learning in multi-view scenarios [35], which aims to learn the graph across views for clustering. Nie et al. [36] develop a parameter-free multiple graph framework to learn the multi-view representation. The combination of the nearest-neighbor techniques has also been investigated to learn the latent graph structure [37]. Due to linear projection, these models may not achieve satisfactory performance on complex datasets. Besides, to fuse the different representations across views, Lin et al. [38] introduce the contrastive network and minimizes the conditional entropy. Jiang et al. [39] introduce the adaptive regression to discriminate diverse views in self-supervised learning. However, these models both utilize linear weight or sparse weight to ignore some views for fusion, which caused some complementary information to be discarded. To overcome this issue and improve the representational ability, we introduce a graph autoencoder on each view to

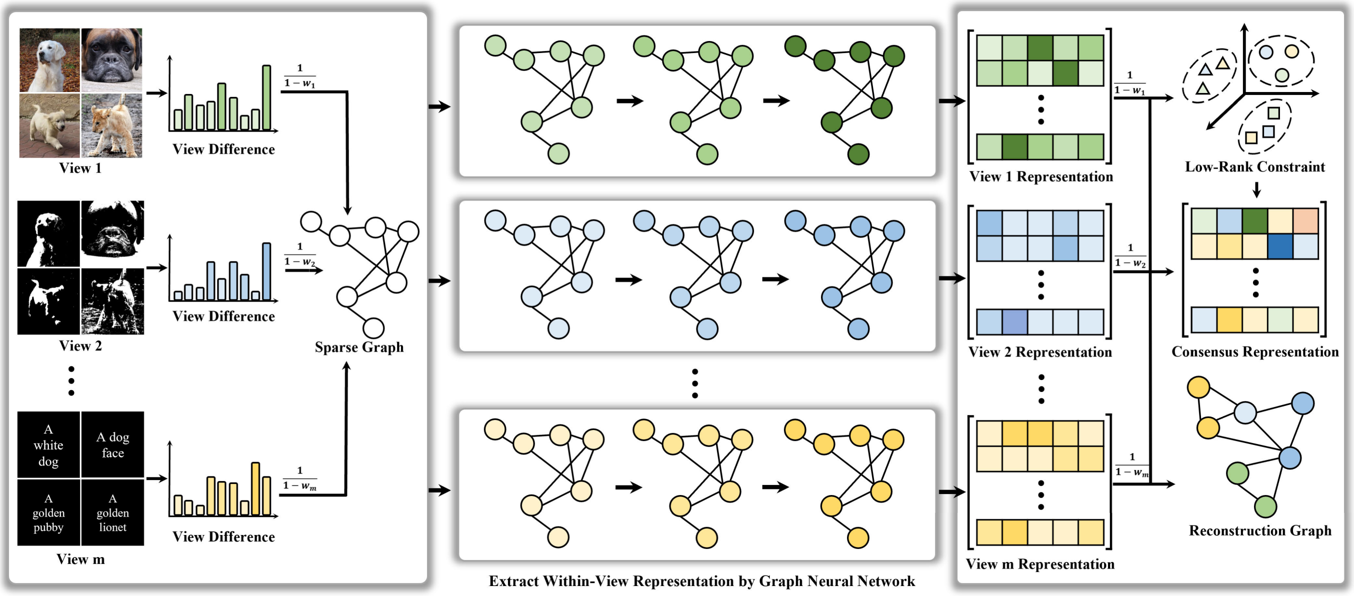


Fig. 2. Architecture of the deep graph autoencoder for multi-view representation learning. For m views data, our model consists of m GNNs on the encoder. It first learns a sparse graph on the original feature space. Then, the encoder extracts the latent structure among each view instances and generates the view-specific representations. Later, according to the low-rank constraint, the model will fuse these representations with the nonlinear weight and learn a consensus representation that cross-views these representations. Meanwhile, the decoder learns the reconstruction graph from the view-specific representations.

learn the latent structure information among the within-view instances and design a self-augmented nonlinear weight to generate a consensus representation across views, which can strengthen the pivotal views and maintain the complementary information from the insignificant views, simultaneously.

III. Framework

Because of the ability to extract the latent structure among the graph nodes, some current models attempt to extend the GNNs on multi-view representation learning. However, since these models generally deploy a GNN on concatenation features or utilize multiple GNNs with shared weights to extract features, the learned unified representation may not be informative enough. Thus, we propose a novel deep graph autoencoder for unsupervised multi-view representation learning. It can not only extend the graph network on the multi-view nongraph data but also learn the informative representation without sharing parameters for GNNs by devising a self-augmented view-weight technique for cross-view fusion. The framework is shown in Fig. 2.

A. Problem Formulation

Suppose a multi-view dataset $\{\boldsymbol{X}^{(v)}\}_{v=1}^{m}$ with m views, where each view $\boldsymbol{X}^{(v)}$ has n samples distributed over c classes. The mainstream models generally learn the multi-view representation under unsupervised scenarios by enforcing the within-view local invariance and cross-view consistency [40]. For within-view information extraction, some methods introduce the DNN f_θ to generate the final representation like $\boldsymbol{Y} = f_\theta(\boldsymbol{X}^{(v)})$. Although it can handle the high dimensional data by projecting the instances into the deep subspace, the within-view structure of the instances may not be extracted and preserved well during the projection. Recently, although some current models attempt to extend the GNNs on multi-view representation learning, they may not learn the informative unified representation due to deploying a GNN on concatenation features or utilizing multiple GNNs with shared weights to extract features. Besides, for cross-view feature learning, the existing multi-view models generally focus on alleviating the negative impact of insignificant views by assigning their corresponding linear weights to 0. However, it leads to some weak yet complementary information from these views may be ignored. Meanwhile, since the weight is assigned to 1 at most, the pivotal views may not be highlighted during the fusion.

Motivated by the above discussion, we will attempt to handle the following questions in this work. *Q1:* How to extend GNN into a multi-view dataset for learning the informative representation?

Q2: How to reinforce the pivotal views while maintaining the complementary views during fusion? Among them, the first question mainly handles the obstacle of generalization of GNN, which will be discussed in Section III-B. Then, we will elaborate on the second question in Section III-C.

B. Deep Graph Autoencoder Within View

In this section, we extend the graph autoencoder on the multi-view dataset to learn the latent structure information among the instances within the view. First, by measuring the global difference across the different views and introducing the ℓ_2-norm relaxation, the model can construct a unified sparse graph among the multi-view dataset. Then, the graph autoencoder can deeply extract the within-view structure information.

Definition 1: For instance $\boldsymbol{x}_i$, the underlying connectivity distribution of this instance is defined as the conditional probability $p(\boldsymbol{x}|\boldsymbol{x}_i)$, which satisfies $\sum_{j=1}^{n} p(\boldsymbol{x}_j|\boldsymbol{x}_i) = 1$.

Definition 2: If the connectivity of $\boldsymbol{x}_i$ is defined as $p(\boldsymbol{x}|\boldsymbol{x}_i)$, the edge can be viewed as a sampling from $p(\boldsymbol{x}|\boldsymbol{x}_i)$.

Sparse Graph Learning: According to Definition 1 and Assumption 2, graph learning is regarded as learning the $p(\boldsymbol{x}|\boldsymbol{x}_i)$ from the datasets in this article. Given the difference $\{\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle\}_{i,j=1}^{n}$ among the datasets, the objective is defined as follows:

$$\min_{p(\cdot|\boldsymbol{x}_i)} \sum_{i,j=1}^{n} p(\boldsymbol{x}_j|\boldsymbol{x}_i)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \mathcal{R}(p(\cdot|\boldsymbol{x}_i)) \tag{1}$$

where $\mathcal{R}(\cdot)$ is the regularization term to prevent the trivial solution: $p(\boldsymbol{x}_i|\boldsymbol{x}_i) = 1$ and $p(\boldsymbol{x}_j|\boldsymbol{x}_i) = 0$ if $i \neq j$. For brevity, $p(\cdot|\boldsymbol{v}_i)$ is simplified as $\boldsymbol{p}_i$. Notably, different from generating the independent graph of each view, we attempt to learn the global difference $\{\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle\}$ across the multi-views. It is defined as follows:

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle = \sum_{v=1}^{m} \frac{1}{1 - w_v} \left\| \boldsymbol{x}_i^{(v)} - \boldsymbol{x}_j^{(v)} \right\|_2^2 \tag{2}$$

where $\boldsymbol{x}_i^{(v)}$ is the ith instance in the vth view. In (2), w_v is a nonlinear weight to fuse the information across the views and can be learned by the proposed self-augmented weight fusion strategy in Section III-C.

Besides, Theorem 1 proves that ℓ_2-norm relaxation can guarantee steerable sparsity and be solved with the closed-form solution. Then, the undirected sparse graph is learned via $\boldsymbol{A}_{ij} = (p_{ij} + p_{ji})/2$. Notably, compared with the vector inner product [32] or the Euclidean distance-based graph learning strategies [33] which introduces an extra threshold to control the sparsity and takes a high cost for tuning and selecting the proper value, the proposed method can adaptively obtain a sparse graph with a steerable sparsity.

Theorem 1 [41]: The ℓ_2-norm relaxation of (1) is

$$\min_{\boldsymbol{p}_i^T \mathbf{1}_n = 1, \boldsymbol{p}_i \geq 0} \sum_{i,j=1}^{n} p(\boldsymbol{x}_j|\boldsymbol{x}_i)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \gamma_i \|\boldsymbol{p}_i\|_2^2 \tag{3}$$

where γ_i is a hyperparameter. $\boldsymbol{p}_i$ has s-sparsity if $\gamma_i = (s/2)\langle \boldsymbol{x}_i, \boldsymbol{x}_{(s+1)}\rangle - (1/2)\sum_{j=1}^{s}\langle \boldsymbol{x}_i, \boldsymbol{x}_{(j)}\rangle$ where $\langle \boldsymbol{x}_i, \boldsymbol{x}_{(\cdot)}\rangle$ is the ·th smallest distance to $\boldsymbol{x}_i$. Then, $\boldsymbol{p}_i$ is solved with

$$p_{ij} = \left(\frac{\langle \boldsymbol{x}_i, \boldsymbol{x}_{(s+1)}\rangle - \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle}{\sum_{v=1}^{s}\langle \boldsymbol{x}_i, \boldsymbol{x}_{(s+1)}\rangle - \langle \boldsymbol{x}_i, \boldsymbol{x}_v\rangle} \right)_+ . \tag{4}$$

Graph Autoencoder: After obtaining graph $\boldsymbol{A}$, the encoder will extend the GNN into the multi-view datasets and deeply extract the within-view structure information. Specifically, the embedding in each view is generated by l graph layers as follows:

$$\boldsymbol{Z}^{(v)} = \varphi_l\left(\boldsymbol{P}\varphi_{l-1}\left(\cdots \varphi_1\left(\boldsymbol{P}\boldsymbol{X}^{(v)}\boldsymbol{\Theta}_1^{(v)}\right)\cdots\right)\boldsymbol{\Theta}_l^{(v)}\right) \tag{5}$$

where $\boldsymbol{\Theta}^{(v)}$ is the trainable parameter in GNN for the v-th view and $\boldsymbol{P} = \phi(\boldsymbol{A})$ is a function of $\boldsymbol{A}$ and $\varphi(\cdot)$ is an activation function. For the decoder, we attempt to reconstruct the connectivity distribution $p(\boldsymbol{x}_j|\boldsymbol{x}_i)$ instead of $\boldsymbol{A}$ like

$$q\left(z_j^{(v)}|z_j^{(i)}\right) = \frac{\exp\left(-\left\|z_i^{(v)} - z_j^{(v)}\right\|_2^2\right)}{\sum_{j=1}^{n}\exp\left(-\left\|z_i^{(v)} - z_j^{(v)}\right\|_2^2\right)}. \tag{6}$$

Inspired by the widely used manifold assumption [25], we encapsulate the local invariance on the manifold to explore the latent structure among the within-view instances. The objective function is defined as follows:

$$\mathcal{L}_1 = \frac{1}{mn^2}\sum_{v=1}^{m}\sum_{i,j=1}^{n} \boldsymbol{A}_{ij}\left\|z_i^{(v)} - z_j^{(v)}\right\|_2^2 + \lambda \mathrm{KL}\left(\boldsymbol{p}_i \| \boldsymbol{q}_i^{(v)}\right) \tag{7}$$

where λ is a hyperparameter. KL$(\cdot||\cdot)$ is the Kullback–Leibler divergence to measure the distance between two distributions.

C. Self-Augmented Weight Fusion

Notably, a nonlinear weight learning, $w^a > 0$ if $w \in [0, \infty)$ and $a \in [0, \infty)$, has been widely utilized in data mining [22], [42], [43]. Inspired this achievement, we integrate the view weights under the different exponential levels ($a = 0, 1, 2, \ldots, \infty$) as follows:

$$\min_{w_v} \sum_{a=0}^{\infty}\sum_{v=1}^{m} w_v^a f(\boldsymbol{X}^{(v)}), \quad \text{s.t. } \boldsymbol{w}^T\mathbf{1}_m = 1,\ w_v \in [0, 1] \tag{8}$$

where $f(\cdot)$ measures the importance of the views and a smaller $f(\boldsymbol{X}^{(v)})$ means the higher significance of the vth view.

Lemma 1: For a series of numbers $\{r^a\}_{a=0}^{\infty}$, $\sum_{a=0}^{n} r^a = (1/(1-r))$ if $|r| < 1$.

According to Lemma 1, the dual problem of (8) is

$$\min_{w_v} \sum_{v=1}^{m} \frac{1}{1 - w_v} f(\boldsymbol{X}^{(v)}), \quad \text{s.t. } \boldsymbol{w}^T\mathbf{1}_m = 1,\ w_v \in [0, 1]. \tag{9}$$

In general, the higher significance view will have a smaller $f(\cdot)$ and be assigned a smaller weight to minimize the global loss. Notably, (9) introduces the nonlinear weights on different views, which can **retain the less important views** by assigning small values $w_v \to 0$ and $(1/(1 - w_v)) \to 1$, and **strengthen the pivotal views** (e.g., $w_v \to 1$ and $(1/(1 - w_v)) \to \infty$). Thus, this weighting strategy is named self-augmented weight. Furthermore, it is crucial that (9) can be solved with the closed-form solution shown in Section IV-A.

Therefore, by applying the designed nonlinear weight, the global difference across views in (2) can be obtained. Meanwhile, it is also introduced in the deep subspace to fuse the multi-view embeddings and strengthen the pivotal view. It is formulated as follows:

$$\min_{\boldsymbol{w}^T\mathbf{1}_m = 1, w_v \in [0,1], \boldsymbol{S}\mathbf{1}_n = \mathbf{1}_n^T, \boldsymbol{S} \geq 0} \sum_{i,j=1}^{n} \boldsymbol{S}_{ij}\langle z_i, z_j\rangle + \hat{\boldsymbol{\gamma}}\|\boldsymbol{S}\|_F^2 \tag{10}$$

where $\boldsymbol{S}_{ij}$ is the similarity between z_i and z_j in the deep subspace, and $\hat{\boldsymbol{\gamma}}$ is a hyperparameter. Furthermore, since we hope the embeddings are discriminative in the deep subspace, the optimal deep similarity $\boldsymbol{S}$ includes exact c connected components. It can be achieved by adding the low-rank constraint as rank$(\boldsymbol{L}) = n - c$ [44] and (10) is transformed to

$$\begin{aligned} \min_{\boldsymbol{w}, \boldsymbol{S}} \sum_{i,j=1}^{n} \boldsymbol{S}_{ij}\langle z_i, z_j\rangle + \hat{\boldsymbol{\gamma}}\|\boldsymbol{S}\|_F^2, \quad & \text{s.t. } \boldsymbol{w}^T\mathbf{1}_m = 1 \\ w_v \in [0, 1], \quad \boldsymbol{S}\mathbf{1}_n = \mathbf{1}_n^T,\ & \boldsymbol{S} \geq 0,\ \mathrm{rank}(\boldsymbol{L}) = n - c \end{aligned} \tag{11}$$

where $\boldsymbol{L} = \boldsymbol{D} - ((\boldsymbol{S} + \boldsymbol{S}^T)/2)$ is the Laplacian matrix and $\boldsymbol{D}$ is diagonal with the ith entry $\boldsymbol{D}_{ii} = \sum_j ((\boldsymbol{S}_{ij} + \boldsymbol{S}_{ji})/2)$. According to [45], (11) is solved with the dual problem as follows:

$$\begin{aligned}
&\min_{\boldsymbol{w},\boldsymbol{S},\boldsymbol{F}} \sum_{i,j=1}^{n} \boldsymbol{S}_{ij}\langle z_i, z_j\rangle + \hat{\boldsymbol{\gamma}}\|\boldsymbol{S}\|_F^2 + \beta \mathrm{tr}(\boldsymbol{F}^T \boldsymbol{L}\boldsymbol{F})\\
&\text{s.t. } \boldsymbol{w}^T \boldsymbol{1}_m = 1, w_v \in [0,1], \quad \boldsymbol{S}\boldsymbol{1}_n = \boldsymbol{1}_n^T, \ \boldsymbol{S} \geq 0\\
&\qquad \boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}_c
\end{aligned} \tag{12}$$

where β is a hyperparameter and $\boldsymbol{F} \in \mathbb{R}^{n\times c}$ is the spectral matrix, i.e., the eigenvector matrix of $\boldsymbol{L}$. Then, it can be solved with the coordinate descent method shown in Section IV. Notably, although (12) will act as a layer in the proposed framework and can be solved with an analytical solution, it has a high time complexity $O(n^3)$ due to containing the eigen decomposition. To handle this problem, we utilize a neural layer to fit the complex matrix computation and the whole framework can be optimized with the gradient descent in Section III-D.

D. Deep Graph Framework With Self-Augmented Fusion

To generate a reliable deep fusion representation, (12) is unified as the self-augmented fusion layer (AdaLayer) in the deep graph framework. It is formulated as follows:

$$\begin{cases} \boldsymbol{Z}^{(v)} = \mathrm{GNN}^{(v)}\left(\boldsymbol{P}, \boldsymbol{X}^{(v)}, \boldsymbol{\Theta}^{(v)}\right)\\ \boldsymbol{F}, \boldsymbol{S}, \boldsymbol{w} = \mathrm{AdaLayer}\left(\left\{\boldsymbol{Z}^{(v)}\right\}_{v=1}^{m}\right). \end{cases} \tag{13}$$

Among them, the framework has m GNN for extracting the multi-view embeddings and $\{\boldsymbol{Z}^{(v)}\}_{v=1}^m$ is the set of multi-view embeddings. However, since AdaLayer contains the eigen decomposition of $\boldsymbol{L}$, the time complexity $O(n^3)$ may be unaffordable if n is large. Meanwhile, we hope that the fusion representation can be obtained end-to-end. Fortunately, inspired by [46], the complex matrix computation can be fit with a neural layer and solved by gradient descent to minimize the distribution of the results. Therefore, we can transfer the self-augmented cross-view fused representation into the deep graph networks by the following equation:

$$\mathcal{L}_2 = \mathrm{KL}(\boldsymbol{F}||\boldsymbol{Z}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{F}_i \log \frac{1}{\boldsymbol{Z}_i} \tag{14}$$

where $\boldsymbol{F}$ is solved from (12) and $\boldsymbol{Z} = \sum_{v=1}^m (1/(1-w_v))\boldsymbol{Z}^{(v)}$. Notably, since AdaLayer is solved with the analytical solution, the convergence rate is faster than gradient descent. Therefore, AdaLayer will be frozen between every τ epoch to smooth the convergence gap. Finally, the objective function is

$$\mathcal{L} = \mathcal{L}_1 + \eta\mathcal{L}_2 \tag{15}$$

where η is a hyperparameter. By gradient descent, $\mathcal{L}_1$ mainly learns the latent within-view structure, and $\mathcal{L}_2$ achieves the pivotal and complementary cross-view fusion. The specific procedure is summarized in Algorithm 1.

Merits of the Proposed Model: Different from the current deep unsupervised multi-view representation learning, we design a sparse graph learning and extend GNN on the multi-view dataset to extract the latent structure information within the view. Meanwhile, a self-augmented nonlinear weight is developed, which can strengthen and balance the pivotal or complementary information across views to reliably fuse the multi-view representation. Furthermore, by fitting the complex matrix computation with a neural layer, the self-augmented weight fusion is embedded into the GNNs, which can achieve to extract the within-view and cross-view information simultaneously.

Algorithm 1 Graph Multi-View Representation Learning With Within-View Structure and Cross-View Fusion

```
Input: Hyperparameters $\lambda$, $\eta$, $\beta$, sparsity $s$, frozen gap $\tau$,
       multi-view data $\{X^{(v)}\}_{v=1}^{m}$;
Output: view weight $w_v$, fusion representation $z$;
Initialize the $\Theta$ in GNN;
Initialize random $w_v \in \mathbb{R}^+$ and $F \in \mathbb{R}^{n\times c}$ satisfying
    $\sum_v w_v = 1$ and $F^T F = I_c$, respectively;
Calculate $\langle x_i, x_j\rangle$ with Eq. (2);
Calculate $s$-sparse $p_i$ via Theorem 1;
Calculate adjacent $A_{ij} \leftarrow \frac{p_{ij}+p_{ji}}{2}$;
Set $iter = 0$;
repeat
    for $v = 1$ to $m$ do
        Extract the within-view $Z^{(v)}$ via Eq. (5);
        Calculate $q(z_j^{(v)}|z_j^{(i)})$ via Eq. (6);
    end
    Calculate the within-view loss $\mathcal{L}_1$ via Eq. (7);
    if $iter$ not in $\tau$ then
        Update $F$ and $w_v$ via Algorithm 2;
    end
    Calculate cross-view fusion $Z \leftarrow \sum_{v=1}^{m} \frac{1}{1-w_v} Z^{(v)}$;
    Calculate the cross-view loss $\mathcal{L}_2$ via Eq. (14);
    Optimize $\mathcal{L}$ in Eq. (15) via gradient descent;
    $iter = iter + 1$;
until convergence;
```

IV. Optimization and Analysis

In this section, we derive the optimization procedure for the proposed self-augmented weight fusion across views in (12), followed by the theoretical and computational analysis.

A. Optimization Procedure

1) Optimizing $\boldsymbol{w}$ With Fixing $\boldsymbol{F}$ and $\boldsymbol{S}$: Equation (12) is reformulated as follows:

$$\begin{aligned}
&\min_{\boldsymbol{w}^T\boldsymbol{1}_m=1, w_v\in[0,1]} \sum_{i,j=1}^{n} \boldsymbol{S}_{ij}\langle z_i, z_j\rangle\\
\Rightarrow &\min_{\boldsymbol{w}^T\boldsymbol{1}_m=1, w_v\in[0,1]} \sum_{v=1}^{m}\sum_{i,j=1}^{n} \frac{\boldsymbol{S}_{ij}}{1-w_v}\left\|z_i^{(v)} - z_j^{(v)}\right\|_2^2\\
\Rightarrow &\min_{\boldsymbol{w}^T\boldsymbol{1}_m=1, w_v\in[0,1]} \sum_{v=1}^{m} \frac{1}{1-w_v}\mathrm{tr}(\boldsymbol{Z}^{(v)T}\boldsymbol{L}\boldsymbol{Z}^{(v)}).
\end{aligned} \tag{16}$$

If regard $f(\boldsymbol{Z}^{(v)}) = \mathrm{tr}(\boldsymbol{Z}^{(v)T}\boldsymbol{L}\boldsymbol{Z}^{(v)})$, (16) is equivalent to (9). Since (9) is a constrained convex problem, we introduce the Largarian multiplier method and the problem is reformulated as follows:

$$\mathcal{J} = \sum_{v=1}^{m} \frac{f(\boldsymbol{Z}^{(v)})}{1-w_v} - \alpha(\boldsymbol{w}^T\mathbf{1}_m - 1) - \boldsymbol{\xi}^T\boldsymbol{w} \tag{17}$$

where $\boldsymbol{\xi} \in \mathbb{R}^{m\times 1}$ and α are Largarian multipliers. According to KKT conditions, we have

$$\begin{cases} \dfrac{\partial \mathcal{J}}{\partial w_v} = 0 \\ \boldsymbol{\xi}^T\boldsymbol{w} = 0 \\ \xi_v \geq 0 \\ \boldsymbol{w}^T\mathbf{1}_m = 1 \end{cases} \Rightarrow w_v = \left(1 - \sqrt{\frac{f_v}{\alpha}}\right)_+ \tag{18}$$

where $f_v = f(\boldsymbol{Z}^{(v)})$. Furthermore, suppose $f_{(1)} \leq f_{(2)} \leq ,\ldots, \leq f_{(m)}$ and $w_1 \geq w_2 \geq ,\ldots, \geq w_m$. Based on (18), we have

$$\begin{cases} w_p > 0 \Rightarrow \sqrt{\alpha} > \sqrt{f_{(p)}} \\ w_{p+1} = 0 \Rightarrow \sqrt{\alpha} \leq \sqrt{f_{(p+1)}} \end{cases} \tag{19}$$

where $w_v = 0$ if $v > p$. Accordingly, α is solved as follows:

$$\boldsymbol{w}^T\mathbf{1}_m = p - \sum_{v=1}^{p}\sqrt{\frac{f_{(v)}}{\alpha}} = 1 \Rightarrow \sqrt{\alpha} = \frac{1}{p-1}\sum_{v=1}^{p}\sqrt{f_{(v)}}. \tag{20}$$

Substitute (20) into (18), the final solution w_v is

$$w_v = \left(1 - \frac{(p-1)\sqrt{f_{(v)}}}{\sum_{i=1}^{p}\sqrt{f_{(j)}}}\right)_+ \tag{21}$$

when p satisfies the following constraint as

$$\begin{aligned} &\sqrt{f_{(p)}} < \sqrt{\alpha} \leq \sqrt{f_{(p+1)}} \\ \Rightarrow &\frac{\sum_{j=1}^{p}\sqrt{f_{(j)}}}{\sqrt{f_{(p+1)}}} + 1 \leq p < \frac{\sum_{j=1}^{p}\sqrt{f_{(j)}}}{\sqrt{f_{(p)}}} + 1. \end{aligned} \tag{22}$$

Therefore, the optimal self-augmented weight vector $\boldsymbol{w} \in \mathbb{R}^m$, whose the vth element is w_v, has p nonzero entries. Notably, p is a unique constant for the optimal $\boldsymbol{w}$, which is proved by Theorem 2.

Theorem 2: For a constraint problem $\min \sum_{v=1}^{m} (f_v/(1-w_v))$ which satisfies $\boldsymbol{w}^T\mathbf{1}_m = 1, w_v \in [0, 1]$, the number of nonzero activated samples p in the optimal $\boldsymbol{w}$ is unique if $f_v > 0$ for any v.

2) Optimizing $\boldsymbol{S}$ With Fixing $\boldsymbol{w}$ and $\boldsymbol{F}$: Since $\boldsymbol{w}$ and $\boldsymbol{F}$ are fixed, (12) is rewritten as follows:

$$\begin{aligned} &\min_{\boldsymbol{S}\mathbf{1}_n = \mathbf{1}_n^T, \boldsymbol{S}\geq 0} \sum_{i,j=1}^{n} \boldsymbol{S}_{ij}\langle z_i, z_j\rangle + \hat{\boldsymbol{\gamma}}_i\|\boldsymbol{S}_i\|_2^2 + \beta\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{F}) \\ \Rightarrow &\min_{\boldsymbol{S}\mathbf{1}_n = \mathbf{1}_n^T, \boldsymbol{S}\geq 0} \sum_{i,j=1}^{n} \boldsymbol{S}_{ij}\langle z_i \oplus \boldsymbol{F}_i, z_j \oplus \boldsymbol{F}_j\rangle + \hat{\boldsymbol{\gamma}}_i\|\boldsymbol{S}_i\|_2^2 \end{aligned} \tag{23}$$

where $\langle z_i \oplus \boldsymbol{F}_i, z_j \oplus \boldsymbol{F}_j\rangle = \langle z_i, z_j\rangle + \beta\langle \boldsymbol{F}_i, \boldsymbol{F}_j\rangle$, $\langle z_i, z_j\rangle$ is defined with (2), and $\langle \boldsymbol{F}_i, \boldsymbol{F}_j\rangle = \|\boldsymbol{F}_i - \boldsymbol{F}_j\|_2^2$. Therefore, $\boldsymbol{S}_{ij}$ in (23) can be solved according to Theorem 1.

Algorithm 2 Self-Augmented Weight across Views

```
Input: Hyperparameter $\beta$, sparsity $s$, deep embedding $\{\boldsymbol{Z}^{(v)}\}_{v=1}^{m}$;
Output: view weight $\boldsymbol{w}$, spectral matrix $\boldsymbol{F}$;
Initialize $w_v \in \mathbb{R}^+$ and $F \in \mathbb{R}^{n\times c}$ satisfying $\sum_v w_v = 1$ and $F^TF = I$, respectively;
while not convergence do
    Calculate $\langle z_i \oplus \boldsymbol{F}_i, z_j \oplus \boldsymbol{F}_j\rangle$ in Eq. (23);
    Calculate $\boldsymbol{S}_i$ via Theorem 1, $\boldsymbol{L} \leftarrow \frac{\boldsymbol{S}^T+\boldsymbol{S}}{2}$;
    For each view, $f(\boldsymbol{Z}^{(v)}) \leftarrow \mathrm{tr}(\boldsymbol{Z}^{(v)T}\boldsymbol{L}\boldsymbol{Z}^{(v)})$;
    Initialize $p = 0$;
    repeat
        $p \leftarrow p + 1$;
        $w_v \leftarrow (1 - \frac{(p-1)\sqrt{f_{(v)}}}{\sum_{i=1}^{p}\sqrt{f_{(j)}}})_+$;
    until $p$ satisfies constraint Eq. (22);
    Update $F$ by solving problem (24);
end
```

3) Optimizing $\boldsymbol{F}$ With Fixing $\boldsymbol{w}$ and $\boldsymbol{S}$: The loss function (12) can be written as follows:

$$\min_{\boldsymbol{F}} \mathrm{tr}(\boldsymbol{F}^T\boldsymbol{L}\boldsymbol{F}), \quad \text{s.t. } \boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I}_c. \tag{24}$$

According to spectral decomposition [47], the optimal solution of $\boldsymbol{F}$ is composed of the c smallest eigenvectors of $\boldsymbol{L}$. Consequently, Algorithm 2 summarizes the solution procedure of (12). It converges when the difference between two iterations is less than 10^{-3}.

B. Proof of Theorem 2

Without loss of generality, suppose that $f_v > 0$ for any v and $\mathcal{J} = \sum_{v=1}^{m}((f_{(v)}/1 - w_v))$. Under the constraint $\boldsymbol{w}^T\mathbf{1}_m = 1$, we have

$$\mathcal{J} = \frac{f_{(1)}}{\sum_{v=2}^{m} w_v} + \sum_{v=2}^{m}\frac{f_{(v)}}{1-w_v}. \tag{25}$$

We take the derivative of (25) w.r.t. w_v

$$\frac{\partial \mathcal{J}}{\partial w_v} = -\frac{f_{(1)}}{\left(\sum_{v=2}^{m} w_v\right)^2} + \frac{f_{(v)}}{(1-w_v)^2}. \tag{26}$$

Then, the Hessian matrix of (26) is

$$\boldsymbol{H}_{vu} = \frac{\partial \mathcal{J}}{\partial w_v \partial w_u} = \begin{cases} \dfrac{2f_{(1)}}{\left(\sum_{v=2}^{m} w_v\right)^3} + \dfrac{2f_{(v)}}{(1-w_v)^3}, & v = u \\ \dfrac{2f_{(1)}}{\left(\sum_{v=2}^{m} w_v\right)^3}, & v \neq u. \end{cases} \tag{27}$$

Furthermore, it can be unified as follows:

$$\boldsymbol{H} = \frac{2f_{(1)}}{\left(\sum_{v=2}^{m} w_v\right)^3}\mathbf{1}_m\mathbf{1}_m^T + \mathrm{diag}\left(\frac{2f_{(v)}}{(1-w_v)^3}\right). \tag{28}$$

Equation (28) indicates that $\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} > 0, \forall \boldsymbol{x} \in \mathbb{R}^{l\times 1}$. Meanwhile, since (9) is convex, the optimal $\boldsymbol{w}$ is unique.

Notably, $w_v = (1 - \sqrt{(f_{(v)}/\alpha)})_+$ will not decrease if α increases. Due to $p \geq 2$ and $\boldsymbol{w}^T\mathbf{1}_m = 1$, there exists only one α. Combining the constraint (22), p is a unique integer.

C. Computational Complexity

For self-augmented weight fusion across views, the optimization procedure is decomposed into three sub-problems, (16), (23), and (24). Among them, (16) will be solved with (21). Equation (23) is decoupled and solved column-wise. Equation (24) is solved by eigen-decomposition. Meanwhile, due to fusion in the deep subspace, the dimensional of embedding are generally small such as $c \ll n$. Thus, the computational complexity is about $O(n^3)$ caused by the eigen-decomposition. Notably, thanks to the frozen gap τ during knowledge transferring, self-augmented weight fusion may not need to optimize each epoch. Especially for a larger τ, the complexity of Algorithm 1 is close to the gradient descent such as $O(n)$.

V. Experiment

In this section, the two mainstream downstream tasks, clustering and recognition, are employed to evaluate the performance of the proposed model.

A. Implementation Details

Before the training and testing, the same normalization strategy is employed in both the proposed method and the comparative models. Among them, graph $\boldsymbol{A}$ in (5) is normalized via $\boldsymbol{P} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ and the primal feature from the multi-view datasets is accordingly (row-) normalized to the range of [0, 1]. Besides, the proposed model consists of a two-layer GNN. The structure is $d^{(v)}$-2c-c, where $d^{(v)}$ is the input dimension of the vth view and c is the cluster numbers or the categories numbers. All hyperparameters in the proposed method are selected from $\{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$ via grid search. Then, the frozen gap τ is 50 and we follow [48] to initialize the proposed model. Furthermore, the stochastic gradient descent is utilized to train the model and the training epoch is 200. The learning rate is 0.01.

The proposed model and comparative schemes are all implemented with PyTorch 1.2.0 on Windows 10. This device contains an Intel i7-10700F 2.9 GHz CPU, a Nvidia 1660 s GPU, and 32G memory.

B. Clustering Performance

Clustering, as a basic task in the unsupervised scenario, mainly groups a set of points into clusters whose assigned instances are as similar as possible. For the extracted multi-view representation, k-means is employed to learn the final clustering assignments.

1) Experimental Setting: To evaluate the performance of our multi-view deep graph model, 12 state-of-the-art models including three single-view models and nine multi-view models are chosen as the competitors. Among them, three single-view models are spectral clustering (SC) [49], low-rank representation (LRR) [50], and SpectralNet (SNet) [51]. Nine multi-view models are binary multi-view clustering (BMVC) [52], multi-view clustering via deep matrix factorization (MvDMF) [53], latent multi-view subspace clustering (LMSC) [22], multi-view Laplacian network (MvLNet) [40], multi-view deep subspace clustering networks (DSCNs) [54], deep CCA (DCCA) [55], deep multi-view robust representation learning (DCCAE) [56], self-weighted multi-view clustering (SwMC) [57], metaviewer (MVer) [58], and reconstructed graph constrained autoencoder (RGCAE) [59], and graph contrastive multi-view learning (GCP) [60]. The last six methods are based on graph learning.

Besides, the experiments are conducted on three popular multi-view datasets, including Caltech101-20 [61], Reuters [62], and NUS-WIDE-OBJ [63]. The details are listed:

1) *Caltech101-20:* It has 2386 images distributed on 20 classes, which are selected from the Caltech101 dataset. As the same in [53], we extract six handcrafted features to form six views, 48-dimension Gabor, 40-dimension Wavelet Moments, 254-dimension Cenhist, 1984-dimension HOG, 512-dimension GIST, 928-dimension LBP.
2) *Reuters:* It has 18 758 text instances distributed on six classes, which is a subset of the original Reuters dataset. This dataset has five views, an English version and four translation versions (i.e., French, German, Spanish, and Italian). Notably, the views are both high dimensions like 21 531, 24 892, 34 251, 15 506, and 11 547.
3) *NUS-WIDE-OBJ:* It has 30 000 images distributed in 31 classes. This dataset has five views, 65-dimension Color Histogram, 226-dimension Color Moments, 145-dimension Color Correlation, 74-dimension Edge Distribution, and 129-dimension Wavelet Texture.

To be fair, each dataset is randomly split into two sets with equal size. Among them, they are used to tune the hyperparameters and evaluate the performance, respectively. Meanwhile, three clustering metrics, accuracy (ACC), normalized mutual information (NMI), and adjusted mutual information (AMI), are employed for comprehensive evaluation. The higher value represents higher performance.

2) Analysis of Experiments: To verify the performance, we run the proposed model and all comparative models ten times. The mean and variance value of the three evaluation metrics is recorded in Table I. From the results, the proposed model outperforms the other comparative methods on all benchmarks. Specifically, on the Caltech101-20 and Reuters datasets, our accuracy is much higher than the second model. It is mainly because ours introduces a nonlinear weight fusion, which can not only strengthen the pivotal views but also consider the complementary information from the insignificant views compared with others. Furthermore, by employing GNN to mine the latent structure within views and avoid the trivial solutions, we can achieve an excellent NMI and AMI on the high-dimension Reuters dataset. Meanwhile, due to the large size and complex distribution, the AMI of some graph-based multi-view models such as SwMC or DCCAE is only 2.36% and 8.65%. On the contrary, with the assistance of local structure preservation, ours can still work well. Besides, compared with the three single-view models, SC, LRR, SNet,

TABLE I

CLUSTERING PERFORMANCE ON CALTECH101-20, REUTERS, AND NUS-WIDE-OBJ

	Caltech101-20			**Reuters**			**NUS-WIDE-OBJ**		
Models	ACC (%)	NMI (%)	AMI (%)	ACC (%)	NMI (%)	AMI (%)	ACC (%)	NMI (%)	AMI (%)
SC	41.45±3.52	63.21±2.68	51.13±3.08	43.14±2.11	20.21±3.04	23.13±2.84	15.64±3.05	15.31±2.18	12.87±1.88
LRR	39.05±3.37	59.43±2.59	50.89±2.98	41.89±3.55	25.77±3.61	23.78±3.86	14.10±2.18	13.57±2.81	10.24±3.17
SNet	51.14±2.09	63.75±2.48	57.05±2.64	46.77±4.47	24.35±3.51	23.12±3.89	15.80±3.77	15.09±2.89	12.08±2.58
BMVC	34.85±1.57	56.78±1.29	48.69±1.78	46.82±3.17	21.06±2.88	19.52±4.05	13.25±4.38	12.34±3.76	9.11±3.85
MvDMF	34.18±2.19	47.02±2.54	38.88±1.82	42.77±1.24	23.15±1.76	17.88±1.47	12.77±0.62	7.88±0.89	8.15±0.99
LMSC	37.12±1.74	56.74±2.25	47.36±2.67	39.15±1.86	28.46±2.38	27.97±2.28	15.29±0.85	16.78±0.73	13.07±1.22
MvLNet	50.38±1.38	63.88±1.85	56.15±1.63	48.86±2.15	26.75±1.73	25.64±1.82	13.64±1.06	11.13±0.95	13.51±1.21
DSCN	52.31±3.41	49.38±2.42	51.46±2.67	48.55±2.74	31.29±3.85	34.57±1.38	14.88±2.11	15.71±1.84	11.32±3.18
DCCA	42.83±2.59	62.53±2.18	52.17±2.76	28.54±1.07	7.78±0.55	8.76±1.27	15.78±2.18	11.75±1.79	8.17±1.81
DCCAE	44.81±3.81	61.07±2.69	52.71±3.53	30.19±1.72	8.95±1.48	19.27±1.75	14.71±0.93	11.52±0.86	8.65±1.08
SwMC	49.97±1.77	62.16±1.53	57.86±1.28	32.76±2.46	23.75±2.07	14.56±1.83	13.79±1.31	9.46±1.08	2.36±1.27
MVer	45.12±1.87	60.86±2.07	57.72±1.95	54.39±1.65	37.56±1.47	39.14±1.27	15.59±0.93	13.27±0.58	10.72±0.64
RGCAE	39.77±2.91	50.39±2.44	43.85±3.51	51.73±1.87	35.22±2.03	34.85±1.38	14.26±0.71	16.44±0.83	15.07±0.59
GCP	53.22±3.51	58.74±2.65	52.41±2.14	51.08±1.33	35.85±2.81	30.51±2.09	14.74±1.24	15.38±0.74	13.68±0.55
Ours	**56.79±1.15**	**64.87±0.92**	**58.68±1.02**	**57.61±1.53**	**41.41±1.33**	**39.65±1.09**	**15.98±0.52**	**16.82±0.31**	**15.72±0.19**

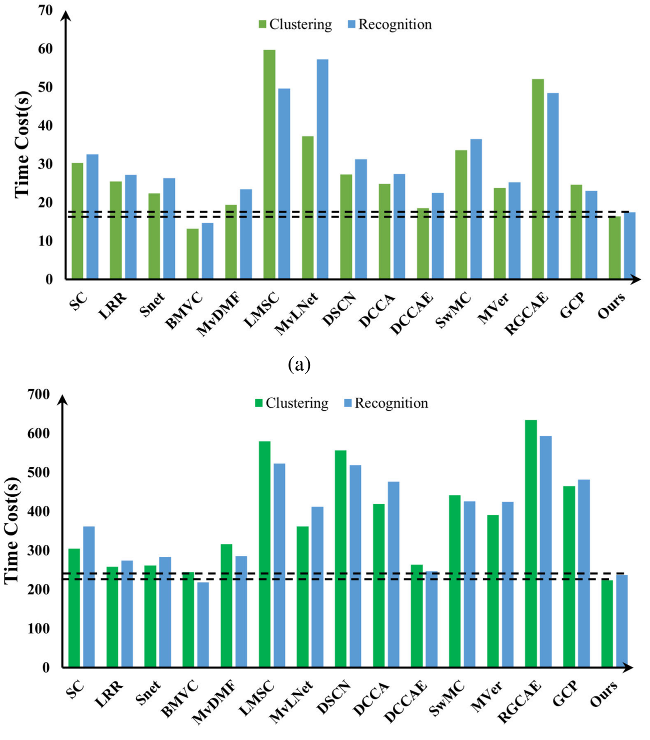
nine multi-view comparative models can achieve better performance on these multi-view benchmarks due to utilizing the multi-view information. And the graph-based methods can also obtain better accuracy thanks to mining the latent structure among the samples. Moreover, since the samples have low dimensionality and contain less information, almost all models cannot mine the discriminative representations and fail to perform well on NUS-WIDE-OBJ.

C. Recognition Performance

Apart from boosting the clustering, we also attempt to evaluate the improvement from the proposed model on recognition tasks. Different from clustering, recognition tasks need to predict the classification probability according to the label information. Thus, a two-layer perceptron is utilized as the classifier. Among them, the middle and last layer has 64 and c neurons respectively, where c is the number of categories on each benchmark.

1) Experimental Setting: Similar to the clustering task, we employ Caltech101-20, Reuters, and NUS-WIDE-OBJ as benchmark datasets. They are also split into two parts of equal size, one for training and another for testing. Since SwMC could not extract the representation for classification, these results are not reported. Besides, as SC will have the same result with the combination of LE [25] and k-means, SC is replaced with LE for recognition tasks. During the inference, the proposed model and all comparative schemes will first extract the representation. Then, the two-layer perceptron will perform the recognition and three metrics, accuracy (ACC), F1-score (F1), and Precision, are employed for evaluation.

2) Analysis of Experiments: The proposed model and all comparative methods are run ten times. The mean and variance results are recorded in Table II. Based on this, ours achieves great performance on three datasets. It obtains the top ACC, F1-score, and precision of classification. Although NUS-WIDE-OBJ has complex distribution and almost all models fail to perform well on clustering, these models can achieve higher performance in recognition tasks thanks to introducing the label information. In particular, ours obtain the highest three metrics than the others. It is mainly because ours can extract the latent structure within the view and learn the more abundant information including pivotal and complement knowledge across views compared with the other multi-view



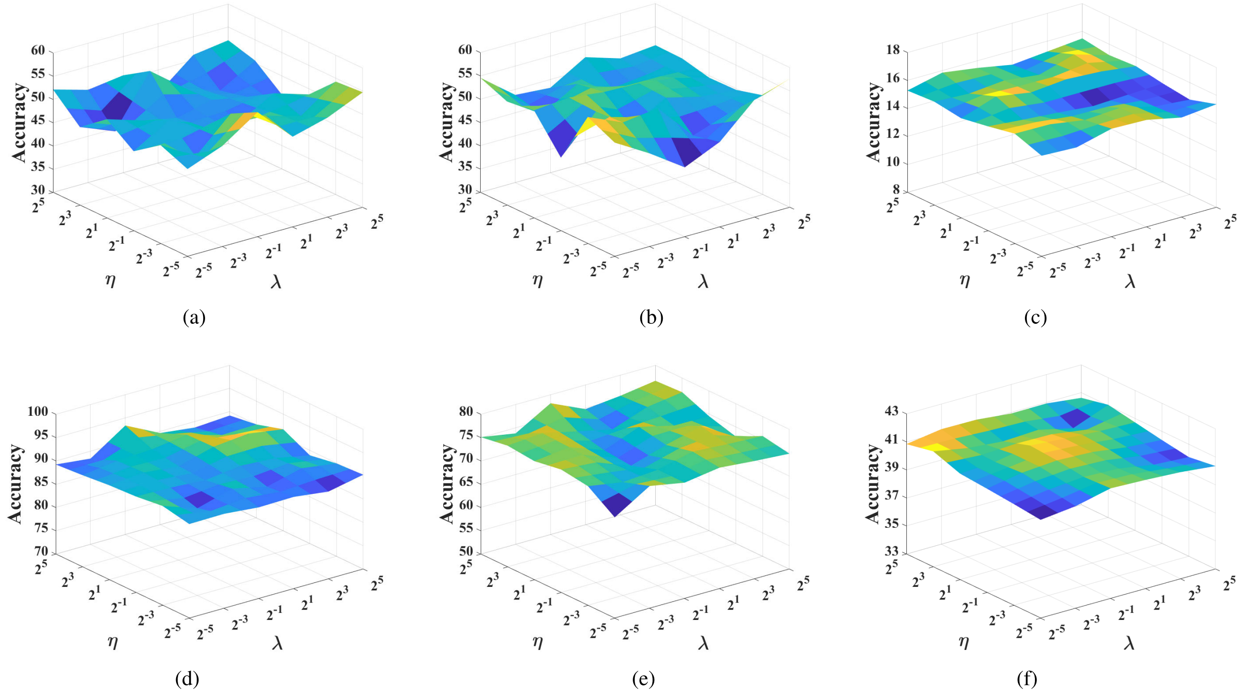
(a)

(b)

Fig. 3. Time cost comparative of the clustering and recognition tasks on (a) Caltech101-20 and (b) reuters datasets. The green and blue bars are the time cost of the clustering and recognition tasks, respectively.

TABLE II
RECOGNITION PERFORMANCE ON CALTECH101-20, REUTERS, AND NUS-WIDE-OBJ

	Caltech101-20			Reuters			NUS-WIDE-OBJ		
Models	ACC (%)	F1 (%)	Precision (%)	ACC (%)	F1 (%)	Precision (%)	ACC (%)	F1 (%)	Precision (%)
LE	83.56±2.75	83.79±3.79	84.56±4.17	71.89±2.67	72.56±2.06	74.89±2.13	21.03±3.55	20.78±3.79	20.16±3.14
LRR	82.76±1.88	83.05±2.02	83.12±1.28	57.81±3.37	58.12±2.14	58.36±2.72	21.13±1.67	22.01±2.64	21.46±1.83
SNet	81.76±1.36	80.72±2.84	80.19±2.09	73.26±3.11	73.27±1.58	73.51±1.29	22.46±3.18	22.71±2.51	22.67±2.87
BMVC	83.16±4.52	83.46±4.84	83.69±3.69	70.89±2.15	71.38±3.81	71.84±1.76	13.02±2.81	12.83±1.76	13.04±2.43
MvDMF	67.03±5.77	68.49±3.08	72.84±4.52	40.75±3.84	40.81±1.82	41.77±2.64	7.56±3.05	7.14±2.83	8.49±3.18
LMSC	83.75±1.28	82.96±1.73	82.72±1.50	58.37±3.38	58.48±2.06	58.61±2.32	22.76±1.63	22.34±2.04	22.61±2.62
MvLNet	84.59±1.29	83.15±2.53	83.16±1.38	75.16±2.58	73.02±1.77	75.46±1.05	23.04±2.83	23.16±1.74	23.71±1.93
DSCN	88.58±2.40	78.23±2.71	82.75±1.87	71.82±1.73	64.65±2.35	69.39±2.13	22.58±1.67	24.81±2.42	23.63±1.86
DCCA	82.13±1.53	81.02±2.87	81.32±2.02	74.13±3.61	72.21±2.17	73.15±2.58	16.31±1.88	16.52±2.50	16.74±1.48
DCCAE	83.37±5.09	82.45±4.58	82.86±5.35	74.22±4.51	73.09±2.78	74.92±3.51	14.90±2.99	15.17±1.45	15.09±2.84
MVer	92.16±3.21	84.72±1.78	81.18±2.66	75.46±5.09	73.12±4.87	75.36±4.55	25.89±2.72	25.11±2.15	20.12±1.83
RGCAE	89.86±0.87	78.01±1.35	81.46±1.08	74.80±1.57	71.05±1.38	76.12±1.86	27.03±1.31	26.36±1.08	27.55±0.95
GCP	91.72±2.33	80.37±1.05	74.66±2.16	75.82±2.81	64.38±1.97	66.28±2.58	29.38±2.54	24.58±1.73	24.37±1.72
Ours	**93.38±1.19**	**84.83±1.28**	**85.11±1.47**	**78.15±2.13**	**73.89±2.41**	**79.21±2.28**	**41.24±2.07**	**28.20±1.68**	**39.24±1.35**



(a) (b) (c)

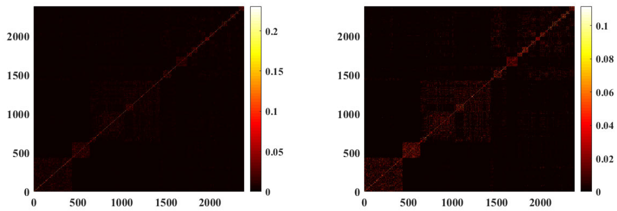
(d) (e) (f)

Fig. 4. Accuracy of ours w.r.t. the varying parameter, η in (15) and λ in (7). They are chosen from $\{2^{-5}, 2^{-4}, \ldots, 2^{4}, 2^{5}\}$. (a)–(c) Clustering accuracy results on all benchmark datasets. (d)–(f) Recognition accuracy results on all benchmark datasets. (e) Reuters recognition ACC.

models. Meanwhile, when handling complex datasets such as the high-dimensional Reuters dataset and large categories NUS-WIDE-OBJ dataset, a nongraph-based model such as MvDMF and a graph-based model like DCCAE cannot have satisfactory performance. Since the proposed model utilizes the self-augmented weight to reliably fuse the different views, it can still achieve excellent performance on complex datasets. Besides, because of guiding with the label information for recognition tasks, the graph-based models can accurately explore the latent structure among the samples and achieve better performance than three single-view or nongraph-based methods. Furthermore, as shown in Fig. 3, thanks to (14) and the frozen gap τ, the matrix decomposition can be fit with the neural layer and our time cost has been significantly reduced.

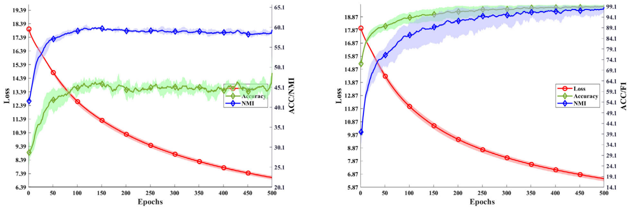
D. Analysis Experiment

In this section, we conduct some experiments including sensitivity analysis, convergence, and ablation study to further investigate the performance of the proposed model. To be fair, the implementation details are the same as in Section V-A.



(a) (b)

Fig. 5. Visualization of the learned sparse graph on Caltech101-20. (a) Sparse graph learned by the designed graph learning strategy. (b) Reconstructed graph based on the deep graph embedding.



(a) (b)

Fig. 6. Influence of epoch w.r.t. the loss, clustering performance and recognition performance. Among them, (a) and (b) influence on the clustering task and recognition task, respectively.

We follow the same settings in Sections V-B1 and V-C1 to conduct the experiment on clustering and recognition tasks, respectively.

1) Sensitivity Analysis: We investigate the impacts of the parameters including the η in (15) and λ in (7) on all benchmark datasets. As shown in Fig. 4, due to the lack of label information, the model is more sensitive to the hyperparameters in clustering compared with recognition. Furthermore, considering all results, we can find that η has more effect on the performance. It is mainly because η can balance the within-view and cross-view information. For a complex NUS-WIDE-OBJ dataset, different values seemly have an insignificant influence on the results. In conclusion, we can set $\eta = 1$ and $\lambda = 1$ simply.

2) Graph Visualization: To investigate the performance of the designed sparse graph learning strategy, we visualize the learned sparse graph $\boldsymbol{A}$ and the final reconstructed graph based on deep embeddings. As shown in Fig. 5(a), the samples in Caltech101-20 mainly are located on the diagonal. It means that the designed strategy has mined the latent connectivity among the samples. Meanwhile, these connectivities are also preserved and even strengthened in Fig. 5(b). It also proves that the distribution in the original feature space can be maintained in the deep subspace under unsupervised scenarios.

3) Convergence Study: Meanwhile, to study the convergence and performance of the proposed method, we study the loss, clustering performance, and recognition performance w.r.t. training epochs in Fig. 6. From this, the proposed model can converge within 500 epochs, which suggests that the designed transferring strategy can unify the within-view and cross-view representation learning reliably. Furthermore, by introducing the frozen gap τ in the Algorithm 1, the loss can decrease quickly and smoothly.

TABLE III

ABLATION STUDY ON VIEW-SPECIFIC VERSUS FUSION REPRESENTATION

Dataset		Clustering		Recognition	
		ACC	NMI	ACC	F1
Caltech101-20	View 1	33.36±1.43	44.01±2.08	78.42±1.63	56.21±1.54
	View 2	34.37±3.55	43.79±1.94	74.26±2.82	44.88±1.15
	View 3	31.97±2.88	40.37±1.58	80.59±1.38	60.37±2.74
	View 4	40.28±1.35	56.16±0.95	71.37±2.49	38.82±2.05
	View 5	30.26±1.53	36.13±2.26	65.34±2.56	33.92±2.08
	View 6	25.02±1.47	34.69±1.25	41.62±2.11	15.45±2.43
	Linear	40.91±1.82	59.06±2.09	83.71±1.70	77.35±2.18
	Self-Aug	**56.79±1.15**	**64.87±0.92**	**93.38±1.19**	**84.83±1.28**
Reuters	View 1	46.35±2.42	37.71±3.43	70.50±3.73	63.87±2.86
	View 2	46.16±2.45	39.29±2.17	71.73±3.07	66.09±3.84
	View 3	46.55±1.78	40.06±2.01	70.81±2.47	62.37±4.56
	View 4	48.78±3.61	40.91±4.11	68.59±3.74	61.40±4.57
	View 5	44.29±2.08	40.59±1.99	68.77±3.69	62.21±4.22
	Linear	38.66±1.74	24.62±1.95	65.03±2.71	55.19±3.64
	Self-Aug	**57.61±1.53**	**41.41±1.33**	**78.15±2.13**	**73.89±2.41**

4) Ablation Study: Furthermore, to explore the effectiveness of the fused representation based on the self-augmented weight, we adopt the k-means and a two-layer perceptron on each view-specific, linear weight fused (Linear) [64], and self-augmented weight fused (Self-Aug) representation for clustering and recognition tasks, respectively. According to Table III, our fused representation outperforms the view-specific remarkably in all scenarios. In other words, the designed self-augmented nonlinear weight learning can learn more discriminative and informative representations by strengthening the pivotal views and preserving the complementary information.

VI. Conclusion

In this article, we propose a novel deep graph multi-view representation learning with self-augmented view fusion, which can learn the within-view structure and fuse cross-view information simultaneously. First, by designing a sparse graph learning strategy, GNN will be generalized on each view for extracting the latent structure among the within-view instances. Meanwhile, different from linear weight to ignore some views, we introduce a self-augmented nonlinear weight which can not only strengthen the pivotal view but also balance the complementary information for fusing the feature across views. Furthermore, by fitting the fusion results with a neural layer, the self-augmented weight fusion is embedded into the GNNs, which can be optimized with the gradient descent and reasonably learn the multi-view fused representation. On the benchmark datasets, the proposed model achieves excellent results. Although the designed graph learning strategy can learn the unified graph among the multi-view data, the learned graph is fixed during the optimization of the whole framework. It means that the learned graph may not accurately reflect the data distribution in the deep subspace. Therefore, we will attempt to introduce a neural layer to differentially learn a dynamical graph in the future, which can not only improve the model performance but also solve the out-of-sample problem to extend the model on more large multi-view scenarios.

REFERENCES

[1] X. Li, “Positive-incentive noise,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 8708–8714, 2024.

[2] W. Zhao, C. Xu, Z. Guan, and Y. Liu, “Multiview concept learning via deep matrix factorization,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 814–825, Feb. 2021.

[3] F. Nie, S. Shi, J. Li, and X. Li, “Implicit weight learning for multi-view clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4223–4236, Aug. 2023.

[4] Y. Liang, D. Huang, C.-D. Wang, and P. S. Yu, “Multi-view graph learning by joint modeling of consistency and inconsistency,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2848–2862, Feb. 2024,

[5] C. Yan, B. Gong, Y. Wei, and Y. Gao, “Deep multi-view enhancement hashing for image retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1445–1451, Apr. 2021.

[6] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” 2013, *arXiv:1304.5634*.

[7] D. Wu, F. Nie, X. Dong, R. Wang, and X. Li, “Parameter-free consensus embedding learning for multiview graph-based clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7944–7950, Dec. 2022.

[8] F. Nie, L. Tian, R. Wang, and X. Li, “Multiview semi-supervised learning model for image classification,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 12, pp. 2389–2400, Dec. 2020.

[9] A. Khan and P. Maji, “Multi-manifold optimization for multi-view subspace clustering,” *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3895–3907, Feb. 2022.

[10] T. Wu, R. Zhang, Z. Jiao, X. Wei, and X. Li, “Adaptive spectral rotation via joint cluster and pairwise structure,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 71–81, Jan. 2023.

[11] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.

[12] C. Zhang et al., “Generalized latent multi-view subspace clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[14] H. Zhang, S. Huang, and X. Li, “Variational positive-incentive noise: How noise benefits models,” 2023, *arXiv:2306.07651*.

[15] H. Zhang, Y. Xu, S. Huang, and X. Li, “Data augmentation of contrastive learning is estimating positive-incentive noise,” 2024, *arXiv:2408.09929*.

[16] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, “MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3539–3553, Mar. 2023.

[17] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, “Image manipulation detection by multi-view multi-scale supervision,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 14165–14173.

[18] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, *arXiv:1609.02907*.

[19] H. Zhang, Y. Zhu, and X. Li, “Decouple graph neural networks: Train multiple simple GNNs simultaneously instead of one,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 11, pp. 7451–7462, Nov. 2024.

[20] S. Xiao, S. Du, Z. Chen, Y. Zhang, and S. Wang, “Dual fusion-propagation graph neural network for multi-view clustering,” *IEEE Trans. Multimedia*, vol. 25, pp. 9203–9215, 2023.

[21] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng, “Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2348–2359, May 2022.

[22] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, “Latent multi-view subspace clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4279–4287.

[23] Q. Zheng, J. Zhu, Z. Li, S. Pang, J. Wang, and Y. Li, “Feature concatenation multi-view subspace clustering,” 2019, *arXiv:1901.10657*.

[24] J. Wen et al., “DIMC-Net: Deep incomplete multi-view clustering network,” in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 3753–3761.

[25] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, Nov. 2002, pp. 585–592.

[26] R. Zhang, Z. Jiao, H. Zhang, and X. Li, “Manifold neural network with non-gradient optimization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3986–3993, Mar. 2023.

[27] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[28] H. Abdi and L. J. Williams, “Principal component analysis,” *WIREs Comput. Statistic*, vol. 2, no. 4, pp. 433–459, Jul./Aug. 2010.

[29] H. Zhang, J. Shi, R. Zhang, and X. Li, “Non-graph data clustering via $\mathcal{O}(n)$o(n) bipartite graph convolution,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8729–8742, Dec. 2023.

[30] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 3844–3852.

[31] X. Li, Z. Jiao, H. Zhang, and R. Zhang, “Deep manifold learning with graph mining,” 2022, *arXiv:2207.08377*.

[32] R. Caramalau, B. Bhattarai, and T.-K. Kim, “Sequential graph convolutional network for active learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9583–9592.

[33] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, “Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 14424–14432.

[34] R. Xia, Y. Pan, L. Du, and J. Yin, “Robust multi-view spectral clustering via low-rank and sparse decomposition,” in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2014, vol. 28, no. 1, pp. 2149–2155.

[35] Y. Yang and H. Wang, “Multi-view clustering: A survey,” *Bid Data Mining Anal.*, vol. 1, no. 2, pp. 83–107, Jun. 2018.

[36] F. Nie, J. Li, and X. Li, “Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification,” in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.

[37] F. Nie, G. Cai, and X. Li, “Multi-view clustering and semi-supervised classification with adaptive neighbours,” in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, vol. 31, no. 1.

[38] Y. Lin, Y. Gou, X. Liu, J. Bai, J. Lv, and X. Peng, “Dual contrastive prediction for incomplete multi-view representation learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4447–4461, Apr. 2023.

[39] B. Jiang et al., “Robust multi-view learning via adaptive regression,” *Inf. Sci.*, vol. 610, pp. 916–937, Sep. 2022.

[40] Z. Huang, J. T. Zhou, H. Zhu, C. Zhang, J. Lv, and X. Peng, “Deep spectral representation learning from multi-view data,” *IEEE Trans. Image Process.*, vol. 30, pp. 5352–5362, 2021.

[41] X. Li, H. Zhang, and R. Zhang, “Adaptive graph auto-encoder for general data clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9725–9732, Dec. 2022.

[42] Y. Li, F. Nie, H. Huang, and J. Huang, “Large-scale multi-view spectral clustering via bipartite graph,” in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2015, vol. 29, no. 1, pp. 2750–2756.

[43] Z. Wang, F. Nie, C. Zhang, R. Wang, and X. Li, “Joint nonlinear feature selection and continuous values regression network,” *Pattern Recognit. Lett.*, vol. 150, pp. 197–206, Oct. 2021.

[44] F. R. Chung, *Spectral Graph Theory*, vol. 92. Providence, RI, USA: American Mathematical Society, 1997.

[45] K. Fan, “On a theorem of Weyl concerning eigenvalues of linear transformations I,” *Proc. Nat. Acad. Sci. USA*, vol. 35, no. 11, pp. 652–655, Nov. 1949.

[46] Z. Jiao, H. Zhang, and X. Li, “Learn topological representation with flexible manifold layer,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[47] J. P. Castagna and S. Sun, “Comparison of spectral decomposition methods,” *First Break*, vol. 24, no. 3, Mar. 2006.

[48] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[49] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, Jan. 2001, pp. 849–856.

[50] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[51] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, “SpectralNet: Spectral clustering using deep neural networks,” 2018, *arXiv:1801.01587*.

[52] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[53] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, vol. 31, no. 1, pp. 2506–3406.

[54] P. Zhu, X. Yao, Y. Wang, B. Hui, D. Du, and Q. Hu, "Multiview deep subspace clustering networks," *IEEE Trans. Cybern.*, vol. 54, no. 7, pp. 4280–4293, Jul. 2024.

[55] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.

[56] Z. Jiao and C. Xu, "Deep multi-view robust representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2851–2855.

[57] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2564–2570.

[58] R. Wang, H. Sun, Y. Ma, X. Xi, and Y. Yin, "MetaViewer: Towards a unified multi-view representation," 2023, *arXiv:2303.06329*.

[59] J. Gou, N. Xie, Y. Yuan, L. Du, W. Ou, and Z. Yi, "Reconstructed graph constrained auto-encoders for multi-view representation learning," *IEEE Trans. Multimedia*, vol. 26, pp. 1319–1332, 2023.

[60] M. Adjeisah, X. Zhu, H. Xu, and T. A. Ayall, "Graph contrastive multi-view learning: A pre-training framework for graph classification," *Knowl.-Based Syst.*, vol. 299, Sep. 2024, Art. no. 112112.

[61] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[62] D. Lewis, *Reuters-21578 Text Categorization Collection*. Irvine, CA, USA: UCI Machine Learning Repository, 1997, doi: 10.24432/C52G6M.

[63] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr.*, Jul. 2009, pp. 1–9.

[64] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2833–2843, Jun. 2016.

Ziheng Jiao received the B.E. degree in computer science from Chang'an University, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an.

Hongyuan Zhang received the B.E. degree in software engineering from Xidian University, Xi'an, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an.

Xuelong Li (Fellow, IEEE) is a Full Professor with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China.