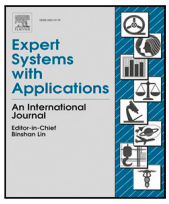
Expert Systems With Applications 263 (2025) 125677

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Multi-scale structure-guided graph generation for multi-view semi-supervised classification

Yilin Wu [a,b], Zhaoliang Chen [c], Ying Zou [a,b], Shiping Wang [a,b], Wenzhong Guo [a,b,*]

[a] College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
[b] Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350108, China
[c] Department of Computer Science, Hong Kong Baptist University, Hong Kong Special Administrative Region of China

ARTICLE INFO

Keywords:
Multi-view learning
Graph generation
Structure preservation
Semi-supervised classification
Multi-scale fusion

ABSTRACT

Graph convolutional network has emerged as a focal point in machine learning because of its robust graph processing capability. Most existing graph convolutional network-based approaches are designed for single-view data, yet in many practical scenarios, data is represented through multiple views. Moreover, due to the complexity of multiple views, normal graph generation methods cannot mitigate redundancy to generate a high quality graph. Although the ability of graph convolutional network is undeniable, the quality of graph directly affects its performance. To tackle the aforementioned challenges, this paper proposes a multi-scale graph generation deep learning framework, called multi-scale semi-supervised graph generation based multi-view classification, consisting of two modules: edge sampling and path sampling. The former aims to generate an adjacency graph by selecting edges based on the maximum likelihood among graphs from different views. Meanwhile, the latter seeks to construct an adjacency graph according to the characteristics of paths within the graphs. Finally, the statistical technique is employed to extract commonality and generate a fused graph. Extensive experimental results robustly demonstrate the superior performance of our proposed framework, compared to other state-of-the-art multi-view semi-supervised approaches.

1. Introduction

In the past decade, there has been a growing interest among researchers in the field of multi-view learning, which involves analyzing data from different perspectives. Most of the time, quality information is hidden in the various sources data. Multi-view learning refers to extracting information from data that comes from multiple sources and has made great progress recently. Therefore, an abundance of multi-view learning approaches is proposed for application fields, such as text categorization (He et al., 2024; Qu et al., 2020; Yang et al., 2020), video analysis (Wang, Tan et al., 2023; Wu et al., 2021; Yan et al., 2022), and object detection (Labbé et al., 2020; Xu et al., 2023; Zou, Cheng et al., 2024). However, the process of obtaining supervisory information in these areas is costly due to the reliance on manual labeling methods.

In response to the above problem, semi-supervised learning has been proposed. Many technologies that require only a small portion of supervisory information to make accurate predictions in machine learning are proposed, such as graph neural network (Bronstein et al., 2017), generative adversarial network (Goodfellow et al., 2014), and Graph Convolutional Network (GCN) (Seo et al., 2018). Specifically speaking, graph neural network leverages the connectivity patterns within graph structures by iteratively updating node representations through message passing. Generative adversarial network incorporates attention mechanisms that allow nodes to selectively aggregate information from their neighbors based on learned weights, leading to more adaptable and powerful node representations for diverse graph-related tasks. GCN extracts meaningful representations from graphs by propagating and aggregating information across nodes via graph convolutional layers. Notably, the remarkable capability of GCNs in multi-view scenarios has been demonstrated by various approaches. To be specific, the ability to acquire semantic features through multiple convolutional layers greatly enhances its performance. Numerous existing GCN variants make substantial contributions to multi-view learning and address the interrelationships among different views. However, these GCN variants generally assume that all views reflect the same underlying type of relational information, and ignore the fact that multi-view data has different feature spaces, where each space has its graph. This has resulted in these GCN variants ignoring the importance of graph quality in the GCN process, making them not well adapted to multi-view learning.

Improving graph quality requires an effective graph generation method. The goal of graph generation is parsing the feature space with

* Corresponding author at: College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China.
E-mail addresses: yilinwu41@163.com (Y. Wu), chenzl23@outlook.com (Z. Chen), zouying5419@163.com (Y. Zou), shipingwangphd@163.com (S. Wang), guowenzhong@fzu.edu.cn (W. Guo).

https://doi.org/10.1016/j.eswa.2024.125677
Received 7 August 2024; Received in revised form 18 October 2024; Accepted 30 October 2024
Available online 8 November 2024

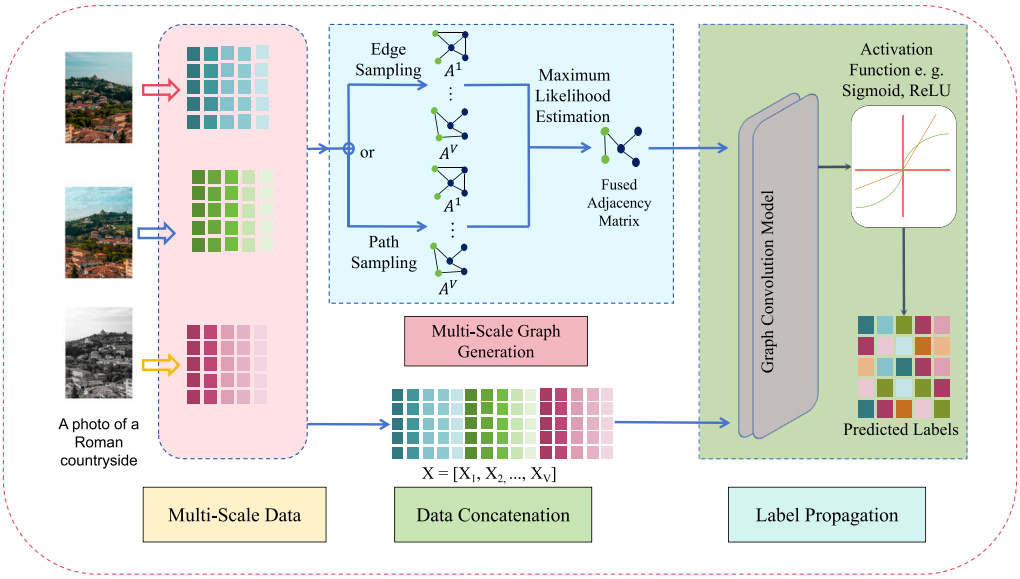


Fig. 1. The proposed method includes the construction of a basis adjacency matrix, the statistical technique, and the label propagation module. It utilizes the first two parts to explore the full information in edge or path separately and obtain a graph corresponding to the statistical technique. With this technique, the complete information is obtained. The label propagation module obtains the final representation.

the purpose of generating a structured representation of instances. Li et al. (2024) To deal with how to learn complete structural information, several attempts on graph generation methods have been proposed to explore graph structures among different feature spaces. In the field of multi-view learning, the information spanning different views is complex and intricate, which makes the simple graph generation difficult to achieve an unified assimilation of information. Moreover, the various graph generation methods are diverse, and there is a lack of a standardized framework to unify the graph generation process.

In this paper, we propose a multi-scale structure-guided graph generation for multi-view learning. The proposed framework has two sampling ways: edge sampling and path sampling. First, a simple yet effective process is used to create the basic adjacency graph from multi-view feature spaces. After that, the structural information from multiple adjacency graphs is collected by alternative sampling methods. Finally, the statistical technique is employed to extract commonality and generate a fused graph from the structural information. This specific graph can share information from each view, allowing better message passing and disturbance resistance. Finally, this graph can be applied to any GCN method with its high quality information in multi-view learning. Therefore, the proposed method is a joint framework that takes messages with edge sampling or path sampling, respectively. The framework is presented in Fig. 1. In a nutshell, our contributions are summarized as follows.

- Two alternative sampling methods are provided to collect the structural information, and then a statistical technique is employed to extract commonality and generate a fused graph.
- An efficient framework for graph generation, integrated with reinforcement learning, has been developed to facilitate the construction of graphs within the context of multi-view learning.
- Comprehensive practical evaluations of six real-world datasets show the superiority of the proposed framework over the state-of-the-arts.

The subsequent sections of this paper are structured as follows. Section 2 provides a thorough review of semi-supervised learning and GCN-based methods on multi-view data. In Section 3, we introduce the proposed method for a detailed description. In Section 4, we show the basic experimental setup, and comparative analysis of experimental results. Finally, Section 5 concludes this paper and presents the directions for future improvement.

2. Related work

In this section, three related areas are introduced, i.e. multi-view learning, graph convolutional network, and graph generation learning.

2.1. Multi-view learning

Multi-view learning is prevalent across diverse fields, which requires multiple observations to obtain the complete information. The downstream tasks in multi-view learning are mainly categorized into clustering and classification. Clustering is an unsupervised learning technique where the goal is to group a set of instances into clusters based on their similarities (Borlea et al., 2017). Cai et al. (2024) presented a representative clustering method applied to the unbalanced incomplete multi-view field. While, classification is a supervised learning technique where the goal is to assign predefined labels to new data points based on a model trained (Yan et al., 2024). However, obtaining sufficient supervision information presents challenges in terms of the time and manpower required (Kilic et al., 2023). Due to the challenges associated with acquiring supervised information, semi-supervised learning has been developed as a viable approach to address this deficiency (Tan et al., 2014). In this context, the methods of semi-supervised classification in multi-view data hold great importance and practical value, especially methods using neural networks (Chiang et al., 2014). Jia et al. (2020) comprehensively exploited the consensus and complementary properties as well as learning both shared and specific representations by employing the shared and specific representation learning network. Wang, Zhao et al. (2023) introduced a cross-view consistency strategy which will lead to a stable training process. Besides, some researchers have tried to apply Graph Convolutional Network (GCN) to multi view semi-supervise and take results. Cheng et al. (2021) developed GCN with clustering on graph-structured data to improve the robustness in multi-view attributes. Yao et al. (2022) conducted a theoretical analysis which effectively addressed the limitations of GCN in accurately assessing the importance of multi-view topologies. Their proposal to enhance GCN with an attention mechanism allowed for the assignment of varying degrees to different topologies, thereby resulting in performance improvements.

2.2. Graph convolutional network

Graph Convolutional Network (GCN) was originally introduced by Kipf and Welling (2017), with the aim of enhancing semi-supervised classification performance. The rule of GCN is formulated as

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

where $\mathbf{H}^{(l)}$ and $\mathbf{H}^{(l+1)}$ are the input data and output data of the $l-$th graph convolutional layer. Here, $\sigma(\cdot)$ denotes a replaceable activation function, and $\tilde{\mathbf{A}} = \mathbf{A}+\mathbf{I}$ represents the self-connection adjacency matrix for any $l \in [l]$. And, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ for any $i \in [N]$, and $\mathbf{W}^{(l)}$ is a learnable weight matrix.

The promising capability of GCN is based on its ability to concurrently aggregate feature and spatial information (Pozna & Precup, 2012). This feature allows for the analysis of concealed relationships among instances, ultimately aiding in the development of a targeted, discerning representation for a specific task. Due to the remarkable effectiveness of GCN, numerous extensions and variants have exhibited notable performance gains. Li et al. (2018) suggested that graph convolution within the GCN represents a specific form of Laplacian smoothing. They subsequently proposed enhancements to the GCN model for semi-supervised learning, offering profound insights into understanding the GCN framework. Chen et al. (2018) interpreted graph convolution as an integral transformation of embedding functions with respect to probability measures. Utilizing Monte Carlo methodology, they implemented a systematic approach for accurate estimation of integrals, implementing a batched training procedure called FastGCN.

However, applying GCN to individual views and then combining the outputs with a multi-view approach inadequately harnesses the interrelationships among views. To address this problem, Khan and Blumenstock (2019) pioneered the use of subspace analysis to combine multiple views and derived the relationships between them before employing GCN on the integrated representation. Nevertheless, their approaches conflated diverse topologies in multi-view data as manifestations of identical relationships, neglecting the advantages of utilizing heterogeneous information. Additionally, Wang et al. (2021) discovered that sparse regularizer learning could be likened to acquiring a parameterized activation function. They developed a differentiable and reusable neural network to dynamically learn data-driven sparse regularizers. This approach aims to facilitate the learning of sparse representations for tasks such as multi-view clustering and semi-supervised classification. With the problem of over-smoothing, Chen et al. (2020) employed two techniques: initial residuals and identity mapping, and successfully demonstrated their effectiveness through theoretical and empirical evidence. Huang et al. (2024) presented a framework that utilizes a diffusion map to obtain geometric information in multiple views. In order to leverage the benefits of GCN, Li et al. (2020) introduced a multi-view semi-supervised learning model. This model effectively integrates graph information from multiple views by incorporating combined Laplacian matrices. This method brings together the principles of co-training, spectral graph information, and the expressive nature of neural networks into a unified framework.

To reduce complexity computation, Wu et al. (2019) introduced a simplified graph convolutional network, which involves the gradual removal of non-linearities and consolidation of weight matrices between adjacent layers. This simplified approach results in a significant speedup of up to two orders of magnitude. Chen et al. (2023) performed a spectral rotation fusion method to obtain a common graph matrix at a feature level. Wu et al. (2024) constructed a special GCN framework to distill node features, with the help of Laplacian embedding. However, current GCN models have shown satisfying performance with single-view graph data, extending GCN to multi-view data remains a significant challenge. Meanwhile, most GCN models follow a conventional approach to constructing graphs, which may limit their ability to explore more comprehensive information.

Table 1
Commonly used notations with their descriptions.

Notations	Descriptions
$\{\mathbf{X}^{(v)} \in \mathbb{R}^{N\times D_v}\}_{v=1}^{V}$	Multi-view data of V views, N samples and D_v features.
$\mathbf{Y}_{\Omega}$	The training label matrix by the label index set Ω.
$\mathbf{S}^{(v)}$	The affinity matrices of each view data.
$\mathbb{N}_q(v)$	The set of k-hop nodes away from node v.
$\mathbf{P}^{(v)}$	The transition possibility matrix for the vth view data.
$\mathbb{S}_{edge}$	The set of affinity matrices from edge sampling.
$\mathbb{S}_{path}$	The set of affinity matrices from path sampling.

2.3. Graph generation learning

Graph generation is a successful strategy for graph learning with GCN. Two major paradigms have been developed in the context of graph generative models. The first category is graph generation with neural networks independently. Representative examples of this model include the variational auto-encoder (VAE) model for sequentialized molecule graphs (Gomez-Bombarelli et al., 2018). For instance, Liu et al. (2018) made a graph-structured variational auto-encoder model by opting to generate graphs sequentially. With the help of VAE, Jiang et al. (2024) obtained a latent space harnessing the representations of the essential views on graphs, attributes, and structures. These models create the individual entries in the graph adjacency matrix (i.e., edges) independently based on VAE. Although these methods enhance efficiency and allow for parallelization, it is still highly computationally efficient.

Another strategy involves the adaptation of traditional classifiers to create graphs. One of the most commonly classifiers is the k-nearest neighbors (kNN), which achieved promising results defined as:

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

where $\mathbf{A}_{ij}^{(v)}$ is the $(i,j)-$the element of the vth view adjacency matrix, $\mathbf{x}_i^{(v)}$ denotes the ith row vector of $\mathbf{X}^{(v)}$, and $\mathcal{N}_k(\mathbf{x}_j^{(v)})$ represents the k nearest neighbors of $\mathbf{x}_i^{(v)}$. Li et al. (2021) introduced a technique that constructs an intrinsic similarity graph in a spectral embedding space rather than the original feature space. Wu, Xie et al. (2020) developed a method to learn the view-specific affinity matrix using the projection map and its intrinsic tensor through low-rank tensor approximation. They also integrated these conditions to jointly learn the optimal affinity matrix. Although current methods for constructing multi-view graphs show promise, they still face challenges in effectively integrating both consistency and complementarity from multi-view graphs. This limitation results in constructing graphs that lack sufficient informativeness.

In this paper, we assume the graph is a kind of state transition matrix. Afterward, we obtain adjacency graphs for each view using edge sampling or path sampling methods, respectively. With the use of the maximum likelihood similarity method, the final graph can comprehensively evaluate the data information of different views. In the end, the final obtained graph could be placed into any GCN-based model, where GCN is used as the backbone classifier.

3. The proposed method

In order to clarify the model formulation and optimization procedure, the commonly used notations are summarized in Table 1. Then, we will elaborate on the problem formulation and optimization procedure of the proposed method.

3.1. Problem formulation

Given multi-view data $\mathcal{X} = \{\mathbf{X}^{(v)}, \mathbf{Y}_{\Omega}\}_{v=1}^{V}$ associated with the label index set Ω, multi-view semi-supervised classification aims to predict test samples using very limited labels. Most of the existing works regard the multi-view semi-supervised classification problems as multi-source fusion problems, including data-level early fusion, feature-level middle fusion, and decision-level late fusion. These fusion approaches can be roughly categorized as discriminative methods. In contrast, we attempt to provide a new perspective of generative models for this type of problem. In particular, these problems are viewed as label propagation problems on a generative graph guided by multi-scale structure priors.

Definition 1 (*Structure-Aware Embedding You, 2021*). For v ∈ V, the node embedding $h_v = f_v(v)$ is structure-aware if there exists a mapping of up to q-hop network neighborhood of node v. That is, there exists a function $g(\cdot)$ satisfying $h_v = g(\mathbb{N}_1(v), \ldots, \mathbb{N}_q(v))$, where $\mathbb{N}_k(v)$ is the set of k-hop nodes away from node v for all $k \in [q]$.

As it tells in Definition 1, we attempt to generate an exact graph under the guidance of multi-scale structure-aware embeddings. In a multi-view learning environment, in order to more accurately characterize the sample, it will be observed from different views, i.e., sampling. Each observation provides a different piece of information. Accordingly, the following two types of sampling methods are utilized to discretize the latent probability space. The approaches for constructing the state transfer matrix of a Markov chain can be categorized into two types: edge sampling and path sampling. The detail definitions of these approaches are presented as follows:

Edge Sampling. [Based on the distance or similarity among instances in feature space, the state transition of each node can be sampled] Denote the set of sampled edges as $\mathbb{S}_{edge} = \{(u_i, v_i)\}_{i=1}^{e} = \{\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \ldots, \mathbf{S}^{(v)}\}$.

Path Sampling. [Based on the path among instances in feature space, the state transition of each nodes can be sampled] Denote the set of sampled paths as $\mathbb{S}_{path} = \{u_1^j, u_2^j, \ldots, u_{m_j}^j\}_{j=1}^{p} = \{\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \ldots, \mathbf{S}^{(v)}\}$.

3.2. Optimization methodology

Assuming an unobserved hidden graph $\mathcal{G}_g$, its observed state set is $\mathbb{S}_{edge} = \{(u^{(i)}, v^{(i)})\}_{i=1}^{e}$ from edges or $\mathbb{S}_{path} = \{\mathrm{p}_j \doteq u_1^{(j)} u_2^{(j)} \cdots u_{t_j}^{(j)}\}_{j=1}^{s}$ from paths sub-graphs. From the perspective of Markov chain, it is to find a transition probability matrix $\mathbf{A}_g$ as an unknown joint probability distribution such that the sampling $\mathbb{S}_{edge}$ or $\mathbb{S}_{path}$ have the highest joint probability. Naturally, the resultant transition probability matrix $\mathbf{A}_g$ corresponds to the adjacency matrix of a graph. Here, $\mathbf{A}_g$ acts as a latent variable to be optimized.

Graph Generation from Edge Sampling. Given the sampled edge set $\mathbb{S}_{edge} = \{(u^{(i)}, v^{(i)})\}_{i=1}^{e}$, we aim to find the graph adjacency matrix $\mathbf{A}_g$ for which the observed data $\mathbb{S}_{edge}$ has the highest joint probability. The joint density at the observed edge samples $\mathbb{S}_{edge} = \{(u^{(i)}, v^{(i)})\}_{i=1}^{e}$ obeying the probability distribution parameterized by $\mathbf{A}_g$ is given as

$$\mathcal{P}(\mathbb{S}_{edge}; \mathbf{A}_g) = \mathcal{P}\left(\mathbb{X}_1 = (u^{(1)}, v^{(1)}), \ldots, \mathbb{X}_e = (u^{(e)}, v^{(e)}); \mathbf{A}_g\right), \tag{3}$$

where $\mathbb{X}_i$ is a random variable obeying the distribution $\mathbf{A}_g$, i.e., $\mathbb{X}_i \sim \mathbf{A}_g$ for any $i \in [e]$. For the assumption of independent and identically distributed random variables, $\mathcal{P}(\mathbb{S}_{edge}; \mathbf{A}_g)$ could be the product of univariate density functions, i.e.,

$$\mathcal{P}(\mathbb{S}_{edge}; \mathbf{A}_g) = \prod_{i=1}^{e} \mathcal{P}\left(\mathbb{X}_i = (u^{(i)}, v^{(i)}); \mathbf{A}_g\right). \tag{4}$$

The optimal generative graph adjacency matrix $\mathbf{A}_g$ can be obtained by solving the following optimization problem

$$\max_{\mathbf{A}_g \in \mathbb{R}^{N \times N}} \mathcal{P}(\mathbb{S}_{edge}; \mathbf{A}_g) \text{ s.t. } \mathbf{A}_g \mathbf{1} = \mathbf{1}, \mathbf{A}_g \geq \mathbf{0}, \tag{5}$$

where $\mathbf{1} \in \mathbb{R}^{N \times 1}$ is a column vector whose entries are all equal to one. Here, the constraint $\mathbf{A}_g \geq \mathbf{0}$ ensures the non-negativity, and $\mathbf{A}_g \mathbf{1} = \mathbf{1}$ guarantees that the sum of each row in $\mathbf{A}_g$ is equal to one. Both constraints make sure that $\mathbf{A}_g$ is a valid probability distribution as well as a valid graph generation. By minimizing the negative log-likelihood and constructing the Lagrange function $\mathcal{L}(\mathbf{A}_g, \lambda)$ as

$$\mathcal{L}(\mathbf{A}_g, \lambda) = -\sum_{i=1}^{e} \ln\left(\mathcal{P}\left(\mathbb{X}_i = (u^{(i)}, v^{(i)}); \mathbf{A}_g\right)\right) + \lambda^{\top}(\mathbf{A}_g \mathbf{1} - \mathbf{1}), \tag{6}$$

where $\lambda = [\lambda_1; \cdots; \lambda_N] \in \mathbb{R}^{N \times 1}$ is a Lagrange multiplier. The method of Lagrange multipliers indicates that the optimal solution is attained when all partial derivatives are equal to zero. Note that $\mathcal{P}\left(\mathbb{X}_i = (u^{(i)}, v^{(i)}); \mathbf{A}_g\right) = [\mathbf{A}_g]_{u^{(i)} v^{(i)}}$. Taking derivatives of $\mathcal{L}(\mathbf{A}_g, \lambda)$ with respect to $\mathbf{A}_g$ and λ, we know

$$\begin{cases} \frac{\partial \mathcal{L}(\mathbf{A}_g, \lambda)}{\partial [\mathbf{A}_g]_{jk}} = -\sum_{i=1}^{e} \sum_{j=u^{(i)}, k=v^{(i)}} \frac{1}{[\mathbf{A}_g]_{jk}} + \lambda_j, \\ \frac{\partial \mathcal{L}(\mathbf{A}_g, \lambda)}{\partial \lambda} = \mathbf{A}_g \mathbf{1} - \mathbf{1}. \end{cases} \tag{7}$$

Setting $\frac{\partial \mathcal{L}(\mathbf{A}_g, \lambda)}{\partial \mathbf{A}_g} = \mathbf{0}$ and $\frac{\partial \mathcal{L}(\mathbf{A}_g, \lambda)}{\partial \lambda} = \mathbf{0}$, we obtain the optimal $\mathbf{A}_g^* \in \mathbb{R}^{N \times N}$ and $\lambda^* \in \mathbb{R}^{N \times 1}$ as

$$\begin{cases} [\mathbf{A}_g^*]_{jk} = \frac{\sum_{i=1}^{e} \delta(u^{(i)} = j, v^{(i)} = k)}{\sum_{k=1}^{N} \sum_{i=1}^{e} \delta(u^{(i)} = j, v^{(i)} = k)}, \\ [\lambda^*]_j = \sum_{k=1}^{N} \sum_{i=1}^{e} \delta(u^{(i)} = j, v^{(i)} = k). \end{cases} \tag{8}$$

Here, $\delta(a, b) \in \{0, 1\}$ is a Boolean function, where $\delta(a, b) = 1$ if both a and b are true; and $\delta(a, b) = 0$ otherwise.

Graph Generation from Path Sampling. Denote the sampled path set as $\mathbb{S}_{path} = \{\mathrm{p}_i \doteq u_1^{(i)} u_2^{(i)} \cdots u_{t_i}^{(i)}\}_{i=1}^{s}$, where s is the number of paths and t_i is the length of the path p_i. Accordingly, we aim to learn a graph adjacency matrix $\mathbf{A}_g$ from the given path set $\mathbb{S}_{path}$. Likewise, it is expected that the joint probability would be maximized, i.e,

$$\mathcal{P}(\mathbb{S}_{path}; \mathbf{A}_g) = \mathcal{P}\left(\mathbb{X}_1 = \mathrm{p}_1, \ldots, \mathbb{X}_s = \mathrm{p}_s; \mathbf{A}_g\right), \tag{9}$$

where $\mathbb{X}_i$ is a random variable obeying the distribution $\mathbf{A}_g$, i.e., $\mathbb{X}_i \sim \mathbf{A}_g$ for any $i \in [s]$. Furthermore, we assume that the random variable set $\{\mathbb{X}_i\}_{i=1}^{s}$ a sequence of independent, identically distributed (IID) random data points, then $\mathcal{P}(\mathbb{S}_{path}; \mathbf{A}_g)$ could be the product of univariate density functions, i.e.,

$$\mathcal{P}(\mathbb{S}_{path}; \mathbf{A}_g) = \prod_{i=1}^{s} \mathcal{P}\left(\mathbb{X}_i = \mathrm{p}_i; \mathbf{A}_g\right). \tag{10}$$

Similar to graph generation from edge sampling, the optimal graph adjacency matrix $\mathbf{A}_g$ can be evaluated by solving the following optimization problem

$$\max_{\mathbf{A}_g \in \mathbb{R}^{N \times N}} \mathcal{P}(\mathbb{S}_{path}; \mathbf{A}_g) \text{ s.t. } \mathbf{A}_g \mathbf{1} = \mathbf{1}, \mathbf{A}_g \geq \mathbf{0}. \tag{11}$$

Due to the high complicacy of the derivative of Eq. (11), we minimize its negative log-likelihood function and define the Lagrange function $\mathcal{L}(\mathbf{A}_g, \lambda)$ as

$$\mathcal{L}(\mathbf{A}_g, \lambda) = -\sum_{i=1}^{s} \ln\left(\mathcal{P}\left(\mathbb{X}_i = u_1^{(i)} u_2^{(i)} \cdots u_{t_i}^{(i)}; \mathbf{A}_g\right)\right) + \lambda^{\top}(\mathbf{A}_g \mathbf{1} - \mathbf{1}), \tag{12}$$

where $\lambda = [\lambda_1; \cdots; \lambda_N] \in \mathbb{R}^{N \times 1}$ is a Lagrange multiplier. We take derivatives of $\mathcal{L}(\mathbf{A}_g, \lambda)$ with respect to $\mathbf{A}_g$ and λ, knowing

$$\begin{cases} \frac{\partial \mathcal{L}(\mathbf{A}_g, \lambda)}{\partial [\mathbf{A}_g]_{jk}} = -\sum_{i=1}^{s} \sum_{h=1}^{t_i - 1} \sum_{j=u_h^{(i)}, k=u_{h+1}^{(i)}} \frac{1}{[\mathbf{A}_g]_{jk}} + \lambda_j, \\ \frac{\partial \mathcal{L}(\mathbf{A}_g, \lambda)}{\partial \lambda} = \mathbf{A}_g \mathbf{1} - \mathbf{1}. \end{cases} \tag{13}$$

Setting $\frac{\partial \mathcal{L}(\mathbf{A}_g, \lambda)}{\partial \mathbf{A}_g} = \mathbf{0}$ and $\frac{\partial \mathcal{L}(\mathbf{A}_g, \lambda)}{\partial \lambda} = \mathbf{0}$, we obtain the optimal $\mathbf{A}_g^*$ and λ^* as

$$\begin{cases} [\mathbf{A}_g^*]_{jk} = \frac{\sum_{i=1}^{s} \sum_{h=1}^{t_i - 1} \delta(u_h^{(i)} = j, u_{h+1}^{(i)} = k)}{\sum_{k=1}^{N} \sum_{i=1}^{e} \sum_{h=1}^{t_i - 1} \delta(u_h^{(i)} = j, u_{h+1}^{(i)} = k)}, \\ [\lambda^*]_j = \sum_{k=1}^{N} \sum_{i=1}^{s} \sum_{h=1}^{t_i - 1} \delta(u_h^{(i)} = j, u_{h+1}^{(i)} = k). \end{cases} \tag{14}$$

3.3. Training loss and optimization algorithm

Aligned to most GCN models, the cross-entropy function is employed as an objective function to obtain the effective information flow in our proposed model,

$$\mathcal{L} = -\sum_{l\in\mathcal{Y}_L}\sum_{c=1}^{C}\mathbf{Y}_{lc}\ln\widehat{\mathbf{Y}}_{lc}, \tag{15}$$

where Y_{Lc} is the cth label of labeled space and $\widehat{\mathbf{Y}} = \mathrm{softmax}(\mathbf{Z})$ represents the predicted class labels. Additionally, $\mathbf{Y}$ is the matrix generated from the real label space of $\mathcal{Y}_L$, and C stands for the total number of labels in the dataset.

The time complexity of the proposed method mainly consists of two modules: graph generation and graph convolution construction. On one hand, when given sampling results, the computational complexity of the former graph generation consumes $\mathcal{O}(N^2)$ on edge sampling and $\mathcal{O}(pN^2)$ on path sampling, where N and p are the sample number and the maximum length of path. On the other hand, that of the latter graph convolution for each view costs $\mathcal{O}(|\mathcal{E}|NDH)$, where D, H, and $|\mathcal{E}|$ represents the numbers of input features, hidden channels, and graph edges, respectively. Considering that the number of hidden features is often much smaller than that of input features, view number, and path length are much smaller than sample number, i.e., $H \ll D$, $V \ll N$, $p \ll N$, therefore, the total time complexity requires $\mathcal{O}(N^2 + ND|\mathcal{E}|)$.

Gathering the multi-scale graph generation module and the label propagation module, the procedures are summarized in Algorithm 1.

Algorithm 1 Multi-Scale Graph Generation based multi-view semi-supervised classification (MSGG)

Require: $\{\mathbf{X}^{(v)}, \mathbf{Y}_\Omega\}_{v=1}^{V}$: Multi-view data with label index set Ω and V views;
$maxIter$: The maximum number of sampling iterations.
Ensure: $\{\hat{\mathbf{y}}_i\}_{i\notin\Omega}$: Predictive labels of test samples.
// Discrete-time Markov chain Construction
for $v = 1$ to V **do**
  Compute the v-th Markov chain $\mathcal{M}^{(v)} \doteq (S^{(v)}, \mathbf{P}^{(v)})$ with the state space $S^{(v)}$ and transition probability $\mathbf{P}^{(v)}$;
end for
// Multi-Scale Graph Generation
Initialize the set of sampling edges as $\mathbb{S}_{edge} = \emptyset$ and the set of sampling paths $\mathbb{S}_{path} = \emptyset$;
for $Iter = 1$ to $maxIter$ **do**
  if Edge sampling **then**
    Randomly sampling an edge (u, v) from V Markov chains and add it to $\mathbb{S}_{edge}$ with Eq. (3);
  else if path sampling **then**
    Randomly sampling a path $u_1u_2\cdots u_m$ from V Markov chains and add it to $\mathbb{S}_{path}$with Eq. (9);
  end if
end for
Evaluate the generative graph using maximum likelihood estimation, and denote the generated transition probability matrix as $\mathbf{A}_g$ with Eq. (5) or Eq. (11);
// Label Propagation on Graphs
for $v = 1$ to V **do**
  Calculate the v-th feature representation $\mathbf{H}^{(v)}$ by a multi-layer perceptron, given as $\mathbf{H}^{(v)} = \mathrm{MLP}_{\boldsymbol{\Theta}^{(v)}}(\mathbf{X}^{(v)})$ with Eq.(15);
end for
Propagate the labels of test samples using some graph backbone network on the graph $\mathcal{G}_g = (\mathbf{H}, \mathbf{A}_g)$ where $\mathbf{H} = [\mathbf{H}^{(1)}, \cdots, \mathbf{H}^{(V)}]$, denoted as $\widehat{\mathbf{Y}} = \mathrm{GNN}(\mathcal{G}_g, \mathbf{Y}_\Omega)$;
Return The predicted label set $\{\hat{\mathbf{y}}_i\}_{i\notin\Omega}$ with $\hat{\mathbf{y}}_i = \arg\max_{j\in[C]}\widehat{\mathbf{Y}}_{ij}$.

Table 2
A brief introduction to all tested multi-view dataset.

Datasets	# Samples	# Views	Feature distributions	# Classes
ALOI	1079	4	64/64/77/13	10
BBCsports	544	2	3,183/3,203	5
GRAZ02	1476	6	512/32/256/500/500/680	4
NGs	500	3	2000/2000/2000	5
NoisyMNIST	15,000	2	784/784	10
Wisconsin	265	2	1703/221	5

4. Experiment

In this section, the proposed framework is compared with state-of-the-art methods on six real-world benchmark datasets. At first, the setting of the experiment is described. The demonstrated performance of the proposed method is exemplified. Additionally, comprehensive experiments are conducted to analyze its parameter sensitivity. Finally, the performance visualization is carried out to demonstrate the performance and effectiveness of the proposed framework.

4.1. Experimental settings

The selected real-world datasets which have different labels obtained from different views are commonly used in the field of multi-view. The detail information of these datasets is listed:

- ALOI[1] is an image dataset which contains objects that are taken under varied light conditions or rotation angles. Its features include 64-D RGB color histograms, 64-D HSV color histograms, 77-D color similarities, and 13-D Haralick features.
- BBCsports[2] is a collection of 5 distinct types of sports websites sourced from BBC. This dataset includes two unique perspectives within its classification framework.
- GRAZ02[3] is an image dataset with different four classes: bicycles, people, cars, and a class that does not contain these objects. Its feature space is extracted from 6 types, i.e., SIFT, SURF, GIST, LBP, PHOG, and WT.
- NGs[4] is a Newsgroup dataset which is a collection of approximately 20,000 newsgroup documents. NGs consists of 500 newsgroup documents in 5 classes. Each document is pre-processed with three different methods to create feature space from three views.
- NoisyMNIST[5] is generated by adding the white Gaussian noise, the motion blur, and a combination of additive white Gaussian noise and reduced contrast to MNIST dataset.
- Wisconsin[6] serves as a good benchmark for testing the effectiveness of graph-based algorithms in classifying nodes. Wisconsin is part of the "WebKB" collection.

The detailed information of these datasets about a statistical summary is presented in Table 2, including the numbers of views, features, and classes.

4.2. Compared methods

The performance of the framework is proved with the six state-of-the-art methods. The detailed information about these algorithms is listed as follows:

[1] http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView
[2] http://mlg.ucd.ie/datasets/segment.html.
[3] https://github.com/EricWang-CS/Dataset.
[4] http://lig-membres.imag.fr/grimal/data.html.
[5] http://yann.lecun.com/exdb/mnist/.
[6] https://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/.

Table 3
Accuracy and F1-score (mean% and standard deviation%) of all compared semi-supervised classification methods with 10% labeled samples. Hyphen (–) denotes the out-of-memory error. The best performance is highlighted in bold and the second best is underlined.

Metric	Method\Dataset	ALOI	BBCsports	GRAZ02	NGs	NoisyMNIST	Wisconsin
ACC	Co-GCN	79.97 ± 1.98	86.31 ± 2.54	40.53 ± 2.56	87.31 ± 3.48	88.90 ± 0.12	69.41 ± 6.10
	GCN-Fusion	86.40 ± 1.51	91.12 ± 1.65	50.76 ± 0.74	82.18 ± 2.07	–	58.32 ± 4.40
	ERL-MVSC	87.88 ± 1.30	90.41 ± 1.09	54.09 ± 1.29	82.18 ± 2.20	–	54.06 ± 8.08
	DSRL	60.65 ± 5.07	90.91 ± 0.85	48.11 ± 1.04	74.49 ± 2.35	–	59.67 ± 9.95
	IMvGCN	64.79 ± 0.80	83.53 ± 0.84	37.52 ± 0.68	85.82 ± 0.70	80.84 ± 0.09	54.46 ± 1.25
	IHGCN	69.27 ± 6.40	92.99 ± 0.21	46.60 ± 0.54	94.44 ± 0.14	90.09 ± 0.02	69.08 ± 0.57
	MSGG-ES	94.84 ± 0.00	**94.91 ± 0.00**	**55.00 ± 0.00**	92.67 ± 0.00	90.26 ± 0.00	**73.95 ± 0.00**
	MSGG-PS	**95.48 ± 0.00**	90.59 ± 0.00	51.89 ± 0.00	**95.24 ± 0.00**	**90.45 ± 0.00**	68.40 ± 0.16
F1	Co-GCN	79.13 ± 2.44	84.99 ± 3.39	38.94 ± 1.50	87.23 ± 3.57	88.38 ± 0.26	41.77 ± 6.73
	GCN-Fusion	86.51 ± 1.54	91.24 ± 1.88	50.48 ± 0.73	81.82 ± 2.17	–	39.66 ± 4.71
	ERL-MVSC	88.42 ± 1.06	90.28 ± 1.58	**54.39 ± 0.00**	82.35 ± 2.20	–	46.97 ± 23.33
	DSRL	91.00 ± 0.79	93.75 ± 0.43	48.64 ± 1.05	74.08 ± 2.90	–	27.60 ± 9.37
	IMvGCN	63.86 ± 1.28	80.39 ± 0.88	36.21 ± 0.72	85.84 ± 0.71	79.96 ± 0.09	43.24 ± 0.71
	IHGCN	64.44 ± 8.19	92.54 ± 0.18	42.21 ± 1.04	94.43 ± 0.14	89.97 ± 0.02	36.37 ± 0.87
	MSGG-ES	94.89 ± 0.00	**94.65 ± 0.00**	54.22 ± 0.00	92.53 ± 0.00	90.11 ± 0.00	**52.40 ± 0.00**
	MSGG-PS	**95.47 ± 0.00**	91.69 ± 0.00	51.10 ± 0.00	**95.25 ± 0.00**	**90.30 ± 0.00**	48.60 ± 0.17

- Co-GCN (Li et al., 2020): This method is a GCN-based method whose strategy employs the graph information from heterogeneous views with an adaptive combined graph Laplacian matrix.
- GCN-Fusion: This is a traditional graph convolutional network that deals with semi-supervised node classification tasks. In the experiment, the average adjacency matrix is acquired during graph convolutions.
- ERL-MVSC (Huang et al., 2021): This approach constructs a linear regression model to deduce view-specific embedding regularizers and automatically train weights of various perspectives.
- DSRL (Wang et al., 2021): This model implements adaptive learning of data-driven sparse regularizers under deep learning models.
- IMvGCN (Wu et al., 2023): This method combines the reconstruction error and Laplacian embedding to formulate multi-view learning.
- IHGCN (Zou, Fang et al., 2024): This approach proposes an implicit heterogeneous graph convolutional network to explore heterogeneity among views.

These methods are competent for semi-supervised classification tasks. All compared algorithms are conducted with the default parameters. Notably, MVAR, WREG, ERL-MVSC, DSRL, IMvGCN, and IHGCN are specifically designed for multi-view learning environments.

4.3. Experimental result

The proposed framework is compared against six models on six multi-view benchmark datasets, utilizing 10% labeled samples for supervision and the number of iterations for the training model was set to 500. Two evaluation metrics are employed to demonstrate the classification performance. The first evaluation metric is accuracy (ACC), which is widely used to calculate the performance of the model. Additionally, the F1-score, which combines precision and recall, is introduced to highlight the differences among models. The means and standard deviations are recorded for all models. There is one more point, each experiment will be repeated six times under the same settings. The numerical results on the ACC and F1-score of all compared models on six datasets are shown in Table 3.

By and large, the MSGG-PS can outperform most comparison algorithms in three datasets, while the MSGG-ES is less capable than it. Specifically speaking, MSGG-PS achieves the best performance that has a 60% probability of getting the best achievement. MSGG-ES can obtain the best or second ranking achievement than other algorithms in 83.3% datasets in terms of ACC and F1-score. Besides, MSGG-ES has the most stable performance than MSGG-PS, achieving decent performance on all datasets. Although, MSGG-PS has a weaker performance than other algorithms except in ALOI, HGs, and NoisyMNIST. Random walk is probably the main reason for its poor effectiveness, and using other path sampling methods may improve the performance of this framework. As a result, the framework of MSGG can defeat other GCN-based models.

4.4. Parameter sensitivity analysis

The effectiveness of hyper-parameter K is investigated in this subsection. The detailed results and analysis of the experiment are shown as follows. The hyper-parameter K is constrained within the range of 1 to 15. The parameter analysis of MSGG on six datasets is shown in Fig. 2. Specifically, the performance of edge sampling is shown in Fig. 2.(a) and .(b), while the others demonstrate the performance of path sampling. Based on the results from the aforementioned figure, it can be visualized that the accuracy and F1-score of each dataset exhibit similar trends both in MSGG-ES and MSGG-PS. With the two most representative datasets, 3Sources and NGs, the performance of the framework improves as the hyper-parameter values increase until reaching an optimal point. Subsequently, the performance begins to decline and fluctuate slightly as the value of the hyper-parameter continues to increase.

The improvement in evaluation metrics can be ascribed to the incorporation of a larger number of neighbors, facilitating the exploration of varied and enhanced structural and feature information. Nevertheless, the subsequent decrease in evaluation metrics is caused by over-smoothing. Exceeding the optimal threshold of neighbors leads to a homogenization of information gathered across all nodes, thereby diminishing the discriminative ability of the model and negatively impacting node classification. However, there are also cases where the impact of changes in hyper-parameter values on the performance of the algorithm on the NoisyMNIST dataset appears to be rather limited.

4.5. Performance visualization

A popular visualization tool, t-SNE (Van der Maaten & Hinton, 2008), is employed to showcase the effectiveness of various representation methods in the BBCSports dataset. The feature representations are projected into 2D space with t-SNE, and the visualization results are depicted in Fig. 3. Here we illustrate the visualization results of Co-GCN, GCN-Fusion, ERL-MVSC, DSRL, IMvGCN, IHGCN, and MSGG, where points with different colors indicate different classes. Particularly noteworthy is that, the distribution of node representations within the same cluster is more concentrated, whereas different clusters are more distinctly separated. The visualization of MSGG-ES and MSGG-PS are better compared to other algorithms. While a few outliers still persist, their presence is substantially reduced, leading to tighter clustering compared to other algorithms. Collectively, the framework MSGG has better performance for feature representation.

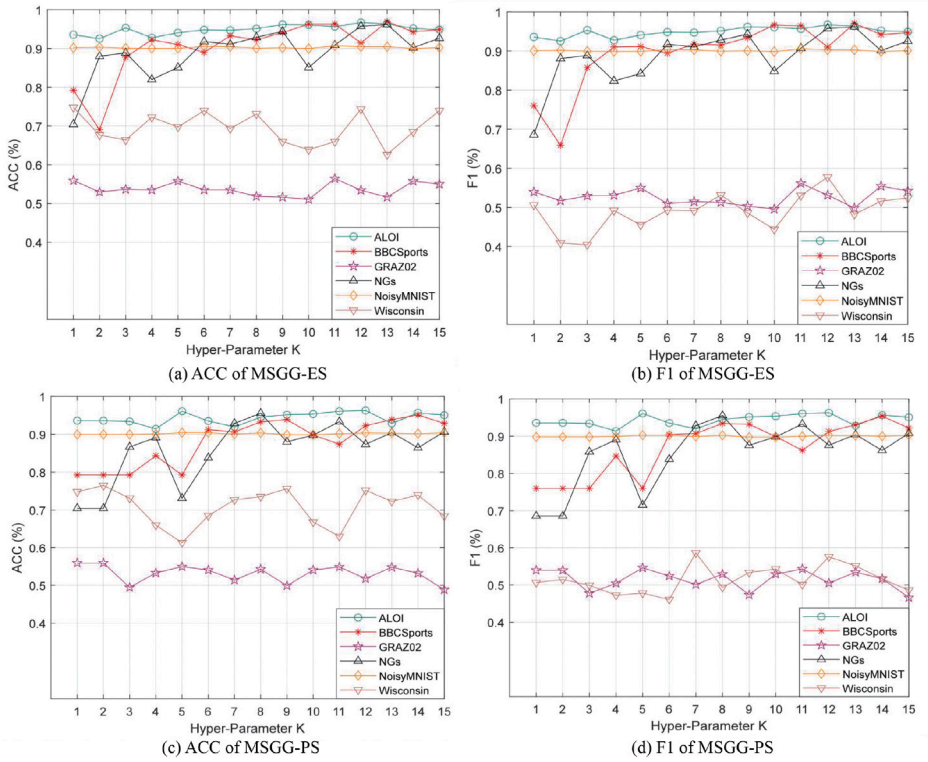


Fig. 2. Parameter analysis of MSGG with hyper-parameter K on six datasets.

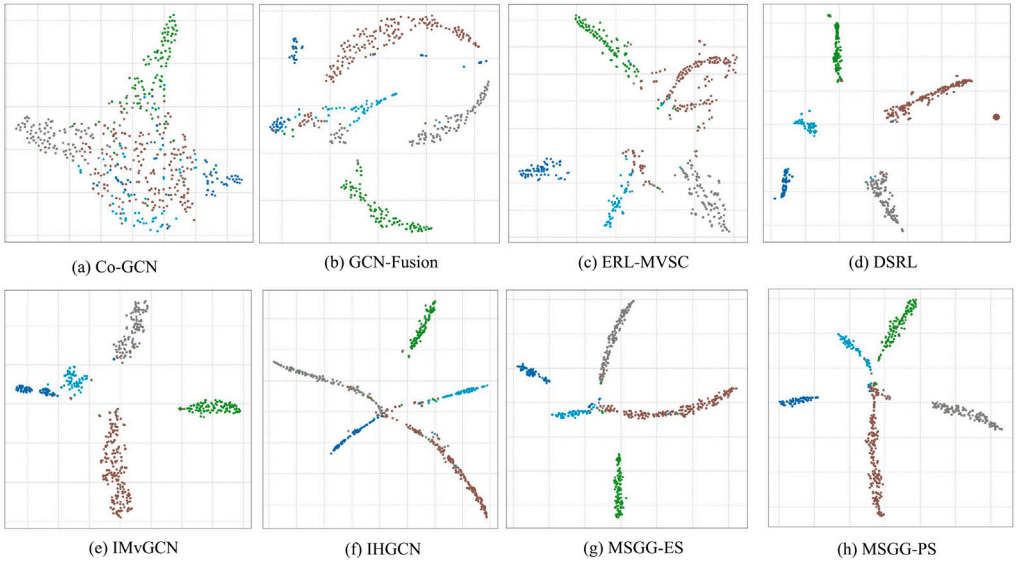


Fig. 3. Visualization results of the learned graph representations of compared algorithms on the BBCSports dataset.

5. Conclusion

In this paper, we designed a graph generation framework, which utilized maximum likelihood in multi-view data, and deployed it in a semi-supervised classification task. Existing graph generation models failed to fully leverage the wealth of multi-view information available and obtain a high quality graph. Therefore, the proposed model incorporated maximum likelihood to better explore the information present in multi-view data. First, this framework can generate graphs through two sampling methods, namely edge sampling and path sampling. Then, in the edge sampling step, the basic adjacency matrix is considered as the output of the sampling process, while the random walk is performed during the path sampling step in the proposed model. Finally, the proposed framework generated a graph by maximum likelihood estimation and utilized GCN to train the results.

Therefore, our work can be further improved in these directions. First, it will be a remarkable endeavor that integrates the proposed framework with other works. Then, graph generation rarely has an effective framework work to address multi-view learning. Effectively leveraging information from various perspectives for graph generation can significantly enhance model performance. Finally, in the proposed model, the adjacency matrix fails to change once it is constructed. It would maintain a more efficient transmission of sample information. Therefore, it would be interesting to investigate more efficient

graph generation methods without external information in multi-view learning.

CRediT authorship contribution statement

Yilin Wu: Conceptualization, Methodology, Code. **Zhaoliang Chen:** Writing – original draft, Data Processing. **Ying Zou:** Visualization, Investigation. **Shiping Wang:** Writing – review & editing. **Wenzhong Guo:** Supervision.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Acknowledgments

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

Data availability

Data will be made available on request.

References

Borlea, Ioan-Daniel, Precup, Radu-Emil, Dragan, Florin, & Borlea, Alexandra-Bianca (2017). Centroid update approach to K-means clustering. *Advances in Electrical & Computer Engineering*, *17*(4).

Bronstein, Michael M., Bruna, Joan, LeCun, Yann, Szlam, Arthur, & Vandergheynst, Pierre (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, *34*(4), 18–42.

Cai, Yiran, Che, Hangjun, Pan, Baicheng, Leung, Man-Fai, Liu, Cheng, & Wen, Shiping (2024). Projected cross-view learning for unbalanced incomplete multi-view clustering. *Information Fusion*, *105*, Article 102245.

Chen, Jie, Ma, Tengfei, & Xiao, Cao (2018). FastGCN: Fast learning with graph convolutional networks via importance sampling. In *International conference on learning representations* (pp. 1–15).

Chen, Jie, Mao, Hua, Peng, Dezhong, Zhang, Changqing, & Peng, Xi (2023). Multi-view clustering by consensus spectral rotation fusion. *IEEE Transactions on Image Processing*, *32*, 5153–5166.

Chen, Ming, Wei, Zhewei, Huang, Zengfeng, Ding, Bolin, & Li, Yaliang (2020). Simple and deep graph convolutional networks. In *International conference on machine learning* (pp. 1725–1735).

Cheng, Jiafeng, Wang, Qianqian, Tao, Zhiqiang, Xie, Deyan, & Gao, Quanxue (2021). Multi-view attribute graph convolution networks for clustering. In *Proceedings of the international joint conferences on artificial intelligence* (pp. 2973–2979).

Chiang, Hsiu-Sen, Shih, Dong-Her, Lin, Binshan, & Shih, Ming-Hung (2014). An APN model for Arrhythmic beat classification. *Bioinformatics*, *30*(12), 1739–1746.

Gomez-Bombarelli, Rafael, Wei, Jennifer N., Duvenaud, David, Hernández-Lobato, José Miguel, Sánchez-Lengeling, Benjamín, Sheberla, Dennis, Aguilera-Iparraguirre, Jorge, Hirzel, Timothy D., Adams, Ryan P., & Aspuru-Guzik, Al'an (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, *4*(2), 268–276.

Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., & Bengio, Yoshua (2014). Generative adversarial nets. In *Neural information processing: 21st international conference* (pp. 2672–2680).

He, Wei, Farrahi, Katayoun, Chen, Bin, Peng, Bohua, & Villavicencio, Aline (2024). Representation transfer and data cleaning in multi-views for text simplification. *Pattern Recognition Letters*, *177*, 40–46.

Huang, Aiping, Lu, Jielong, Wu, Zhihao, Chen, Zhaoliang, Chen, Yuhong, Wang, Shiping, & Zhang, Hehong (2024). Geometric localized graph vonvolutional network for multi-view semi-supervised classification. *Information Sciences*, Article 120769.

Huang, Aiping, Wang, Zheng, Zheng, Yannan, Zhao, Tiesong, & Lin, Chia-Wen (2021). Embedding regularizer learning for multi-view semi-supervised classification. *IEEE Transactions on Image Processing*, *30*, 6997–7011.

Jia, Xiaodong, Jing, Xiao-Yuan, Zhu, Xiaoke, Chen, Songcan, Du, Bo, Cai, Ziyun, He, Zhenyu, & Yue, Dong (2020). Semi-supervised multi-view deep discriminant representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(7), 2496–2509.

Jiang, Xinke, Qin, Zidi, Xu, Jiarong, & Ao, Xiang (2024). Incomplete graph learning via attribute-structure decoupled variational auto-encoder. In *Proceedings of the 17th ACM international conference on web search and data mining* (pp. 304–312).

Khan, Muhammad Raza, & Blumenstock, Joshua E. (2019). Multi-gcn: Graph convolutional networks for multi-view networks, with applications to global poverty. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 606–613).

Kilic, Umit, Essiz, Esra Sarac, & Keles, M. Kaya (2023). Binary anarchic society optimization for feature selection. *Romanian Journal of Information Science and Technology*, *26*(3–4), 351–364.

Kipf, Thomas N., & Welling, Max (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations* (pp. 1–14).

Labbé, Yann, Carpentier, Justin, Aubry, Mathieu, & Sivic, Josef (2020). Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer vision–ECCV 2020: 16th European conference* (pp. 574–591).

Li, Qimai, Han, Zhichao, & Wu, Xiao-Ming (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 3538–3545).

Li, Shu, Li, Wen-Tao, & Wang, Wei (2020). Co-gcn for multi-view semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 4691–4698).

Li, Zhenglai, Tang, Chang, Liu, Xinwang, Zheng, Xiao, Zhang, Wei, & Zhu, En (2021). Consensus graph learning for multi-view clustering. *IEEE Transactions on Multimedia*, *24*, 2461–2472.

Li, Hongsheng, Zhu, Guangming, Zhang, Liang, Jiang, Youliang, Dang, Yixuan, Hou, Haoran, Shen, Peiyi, Zhao, Xia, Shah, Syed Afaq Ali, & Bennamoun, Mohammed (2024). Scene graph generation: A comprehensive survey. *Neurocomputing*, *566*, Article 127052.

Liu, Qi, Allamanis, Miltiadis, Brockschmidt, Marc, & Gaunt, Alexander (2018). Constrained graph variational autoencoders for molecule design. *Advances in Neural Information Processing Systems*, 7806–7815.

Pozna, Claudiu, & Precup, Radu-Emil (2012). Aspects concerning the observation process modelling in the framework of cognition processes. *Acta Polytechnica Hungarica*, *9*(1), 203–223.

Qu, Leigang, Liu, Meng, Cao, Da, Nie, Liqiang, & Tian, Qi (2020). Context-aware multi-view summarization network for image-text matching. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 1047–1055).

Seo, Youngjoo, Defferrard, Michaël, Vandergheynst, Pierre, & Bresson, Xavier (2018). Structured sequence modeling with graph convolutional recurrent networks. In *Neural information processing: 25th international conference* (pp. 362–373).

Tan, Garry Wei-Han, Ooi, Keng-Boon, Leong, Lai-Ying, & Lin, Binshan (2014). Predicting the drivers of behavioral intention to use mobile learning: A hybrid SEM-Neural Networks approach. *Computers in Human Behavior*, *36*, 198–213.

Van der Maaten, Laurens, & Hinton, Geoffrey (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(11), 2579–2605.

Wang, Shiping, Chen, Zhaoliang, Du, Shide, & Lin, Zhouchen (2021). Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5042–5055.

Wang, Feng, Tan, Sinan, Li, Xinghang, Tian, Zeyue, Song, Yafei, & Liu, Huaping (2023). Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 19706–19716).

Wang, Zicheng, Zhao, Zhen, Xing, Xiaoxia, Xu, Dong, Kong, Xiangyu, & Zhou, Luping (2023). Conflict-based cross-view consistency for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 19585–19595).

Wu, Zhihao, Chen, Zhaoliang, Du, Shide, Huang, Sujia, & Wang, Shiping (2024). Graph convolutional network with elastic topology. *Pattern Recognition*, *151*, Article 110364.

Wu, Wenhao, He, Dongliang, Lin, Tianwei, Li, Fu, Gan, Chuang, & Ding, Errui (2021). Mvfnet: Multi-view fusion network for efficient video recognition. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 2943–2951).

Wu, Zhihao, Lin, Xincan, Lin, Zhenghong, Chen, Zhaoliang, Bai, Yang, & Wang, Shiping (2023). Interpretable graph convolutional network for multi-view semi-supervised learning. *IEEE Transactions on Multimedia*, 8593–8606.

Wu, Felix, Souza, Amauri, Zhang, Tianyi, Fifty, Christopher, Yu, Tao, & Weinberger, Kilian (2019). Simplifying graph convolutional networks. In *International conference on machine learning* (pp. 6861–6871).

Wu, Jianlong, Xie, Xingyu, Nie, Liqiang, Lin, Zhouchen, & Zha, Hongbin (2020). *Proceedings of the AAAI conference on artificial intelligence* (pp. 6388–6395).

Xu, Chenfeng, Wu, Bichen, Hou, Ji, Tsai, Sam S., Li, Ruilong, Wang, Jialiang, Zhan, Wei, He, Zijian, Vajda, Peter, Keutzer, Kurt, & Tomizuka, Masayoshi (2023). NeRF-Det: Learning geometry-aware volumetric representation for multi-view 3D object detection. In *IEEE/CVF international conference on computer vision* (pp. 23263–23273).

Yan, Shen, Xiong, Xuehan, Arnab, Anurag, Lu, Zhichao, Zhang, Mi, Sun, Chen, & Schmid, Cordelia (2022). Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3333–3343).

Yan, Lu, Zhao, Tao, Xie, Xiangpeng, & Precup, Radu-Emil (2024). OSSEFS: An online semi-supervised ensemble fuzzy system for data streams learning with missing values. *Expert Systems with Applications*, *255*, Article 124695.

Yang, Xiaocui, Feng, Shi, Wang, Daling, & Zhang, Yifei (2020). Image-text multimodal emotion classification via multi-view attentional network. *IEEE Transactions on Multimedia*, *23*, 4014–4026.

Yao, Kaixuan, Liang, Jiye, Liang, Jianqing, Li, Ming, & Cao, Feilong (2022). Multi-view graph convolutional networks with attention mechanism. *Artificial Intelligence*, *307*, Article 103708.

You, Jiaxuan (2021). *Empowering deep learning with graphs*. Stanford University.

Zou, Zixin, Cheng, Weihao, Cao, Yan-Pei, Huang, Shi-Sheng, Shan, Ying, & Zhang, Song-Hai (2024). Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 7900–7908).

Zou, Ying, Fang, Zihan, Wu, Zhihao, Zheng, Chenghui, & Wang, Shiping (2024). Revisiting multi-view learning: A perspective of implicitly heterogeneous Graph Convolutional Network. *Neural Networks*, *169*, 496–505.