Clustering on Attributed Graphs: From Single-view to Multi-view

MENGYAO LI, Hunan University, Changsha, China
ZHIBANG YANG, Changsha University, Changsha, China
XU ZHOU, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
YIXIANG FANG, The Chinese University of Hong Kong, Shenzhen, Shenzhen, China
KENLI LI, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
KEQIN LI, Hunan University, Changsha, China and State University of New York, New York, USA

Attributed graphs with both topological information and node information have prevalent applications in the real world, including recommendation systems, biological networks, community analysis, and so on. Recently, with rapid development of information gathering and extraction technology, the sources of data become more extensive and multi-view data attracts growing attention. Consequently, attributed graphs can be divided into two categories: single-view attributed graphs and multi-view attributed graphs. Compared with single-view attributed graphs, multi-view attributed graphs can provide more complementary information but also pose challenges to fusing information of multi-views. Moreover, attributed graph clustering aims to reveal the inherent community structure of the graph, which is widely applied in fraud detection, crime recognition, and recommendation systems. Recently, numerous methods based on various ideas and techniques have appeared to cluster attributed graphs, thus there is an urgent need to summarize related methods. To this end, we make a timely and comprehensive review of recent methods. Furthermore, we provide a novel standard according to fusion results to classify related methods into three categories: fusion on adjacency matrix methods, fusion on embedding methods, and model-based methods. Moreover, to conduct a comprehensive evaluation of existing methods, this article evaluates these advanced methods with sufficient experimental results and theoretical analysis. Finally, we analyze the challenges and open opportunities to promote the future development of this field.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Mathematics of computing** → *Attributed graph*; • **Computing methodologies** → Machine learning;

Additional Key Words and Phrases: Attributed graph clustering, Multi-view attributed graph, Machine learning

This research was supported by the Programs of NSFC (Grant Nos. U23A20317, 62172146), the Provincial Key R&D Program of Hunan (2024AQ2025, 2023GK2002), the Yuelu Mountain Industrial Innovation Center Project (No. 2023YCII0118), and the Natural Science Foundation of Hunan Province (Grant Nos. 2023JJ10016, 2023JJ30083).

Authors' Contact Information: Mengyao Li, Hunan University, Changsha, China; e-mail: lmy835@hnu.edu.cn; Zhibang Yang (Corresponding author), Changsha University, Changsha, Hunan, China; e-mail: yangzb@ccsu.edu.cn; Xu Zhou, College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan, China; e-mail: zhxu@hnu.edu.cn; Yixiang Fang, The Chinese University of Hong Kong, Shenzhen, Shenzhen, China; e-mail: fangyixiang@cuhk.edu.cn; Kenli Li, College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan, China; e-mail: lkl@hnu.edu.cn; Keqin Li, Hunan University, Changsha, Hunan, China and State University of New York, NY, USA; e-mail: likq@hnu.edu.cn.

ACM 0360-0300/2025/02-ART163
https://doi.org/10.1145/3714407

ACM Reference Format:
Mengyao Li, Zhibang Yang, Xu Zhou, Yixiang Fang, Kenli Li, and Keqin Li. 2025. Clustering on Attributed Graphs: From Single-view to Multi-view. *ACM Comput. Surv.* 57, 7, Article 163 (February 2025), 36 pages. https://doi.org/10.1145/3714407

1 Introduction

As a common type of graph [8, 25, 97], attributed graphs are composed of a set of nodes with individual characteristics and edges that reflect connection relationships and features. Moreover, attributed graphs [71, 84] have prevalent applications in the real world due to their powerful ability to model complex relationships between different entities with various features, including social networks [22], citation networks [27], biological networks [3, 15, 39], and knowledge graphs [17, 41]. For instance, in social networks [11, 40, 57], each individual is represented as a node whose attributes are personal information, and edges connecting two nodes can reflect the communications between individuals. In a biological network of protein–protein-interaction [58, 94], a protein can be seen as a node whose attributes are protein characteristics, and edges can reveal biochemical interactions between two proteins. Recently, with the rapid development of information gathering and extraction technology, data sources have become more extensive, and multi-view data has attracted increasing attention. Compared with single-view attributed graphs, multi-view attributed graphs can model more complex networks with multiple relationships and contain more complementary information. Due to this attractive modeling ability, attributed graphs [71, 84, 96] are becoming ubiquitous and driving the emergence of various tasks in graph data mining.

Attributed graph clustering [14, 37, 93] aims to allocate given nodes into several disjoint clusters, which plays a critical role in recommendation system [54, 69], organization detection, and other applications, as shown in Figure 1. For example, in social media, such as Facebook, Twitter, and Weibo, users and their connections constitute an attributed graph. Node attributes contain the information of users, including place of residence, gender, education level, and other details. We can cluster nodes via attributes and connections to divide users into different groups, which contributes to friend recommendation and interest tribe construction. As a challenging problem, attributed graph clustering needs to effectively combine structural information and node attributes, which can be applied in many real-life applications. Here are some typical applications:

- **Recommendation System.** E-commerce platforms and short video platforms attempt to provide users with goods or videos that users are interested in. Attributed graph clustering can contribute to accurate and effective personalized recommendations [4], as users in the same cluster may have similar interests and preferences.
- **Organization Detection.** In the field of anti-fraud, to obtain huge profits in a short time, the crimes show obvious gang characteristics, while the terrorist organization also has obvious group characteristics in social networks. Thus, attributed graph clustering is beneficial to quickly discover groups for organization detection (i.e., fraud detection [77] and terrorist organization identification [70]).
- **Biomedical Science.** In the fields of biology and medicine, attributed graph clustering can not only identify complexes of new proteins and disease pathogenic factors but also help integrate proteome and genetic data to reveal key insights into the structure of mammalian lipid metabolism networks.
- **Social Network Analysis.** In social networks [57] such as WeChat, Facebook, and Twitter, platforms can gather users and recommend people with the same hobbies, creating a tribe of interests so users can communicate better. Online friends can also be connected in real life to expand the social sphere.

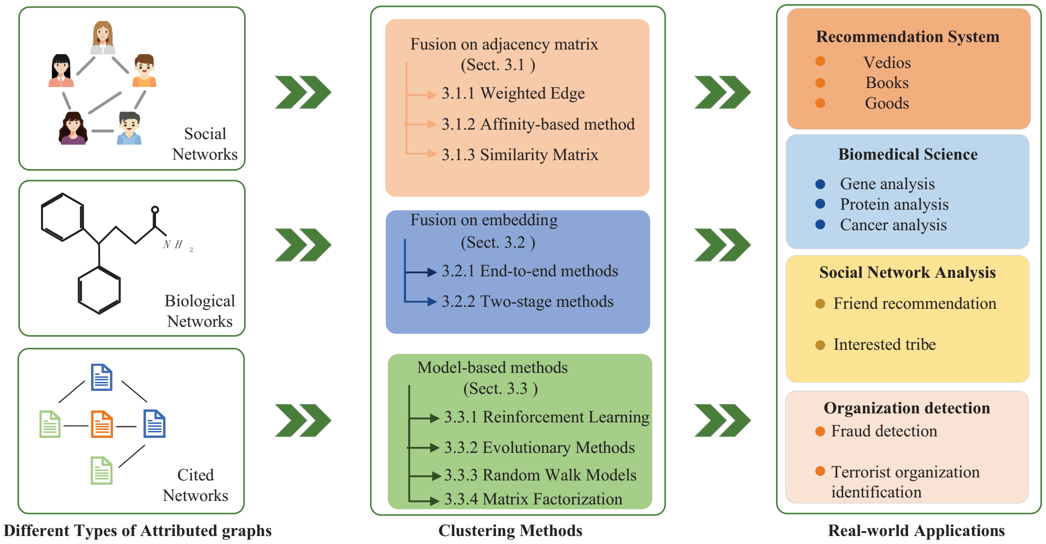


Fig. 1. Applications of attributed graph clustering.

Owing to the importance of attributed graph clustering, many advanced works have emerged. Since there are few works related to edge attributes, in this article, we mainly focus on attributed graphs with node attributes. In this part, we review existing works about clustering attributed graphs in the past decades and classify these works into several groups. Figure 2 shows the timeline of attributed graph clustering development.

Different from plain graphs only with structural information, attributed graphs have not only attributes but also structural information, both of which contribute to the clustering. Hence, it is crucial to make full use of attributes and structural information. However, earlier works often only consider attributes or structure information. For example, k-means methods only focus on node attributes and greatly reduce the clustering accuracy.

To fully utilize attributes and structural information, recent works focus on fusing these two pieces of information. Several works take attributes into structural information to update a new adjacency matrix, such as References [2, 59, 73, 74, 78, 88]. These works convert the attributed graph to the weighted graph by fusing attribute similarity between nodes and original adjacency relationships and then implement the label propagation algorithm or other methods to cluster. Moreover, there are some methods that adopt subspace learning to fuse attributes and structural information to obtain a new adjacency/similarity matrix. Furthermore, many works attempt to utilize spectral clustering methods and provide the needed affinity matrix via fusing attributes and topology.

In recent years, graph neural networks have served as an excellent embedding learning technology in many fields [63, 81, 92]. Due to their excellent ability to learn deep information, GNN-based methods achieve brilliant performance on clustering tasks by optimizing the embeddings. These methods aim to take structural information into attributes by blending attributes of adjacent nodes. However, as GNNs are not designed for clustering, some works presented in References [36, 99, 100] just implement GNNs as feature extractors and then utilize traditional clustering methods on node embeddings. Consequently, the processes of representation learning and clustering are executed separately [23, 44], which limits the accuracy of clustering. Many end-to-end clustering methods in References [83, 85, 89, 91] are proposed to build a clustering-oriented

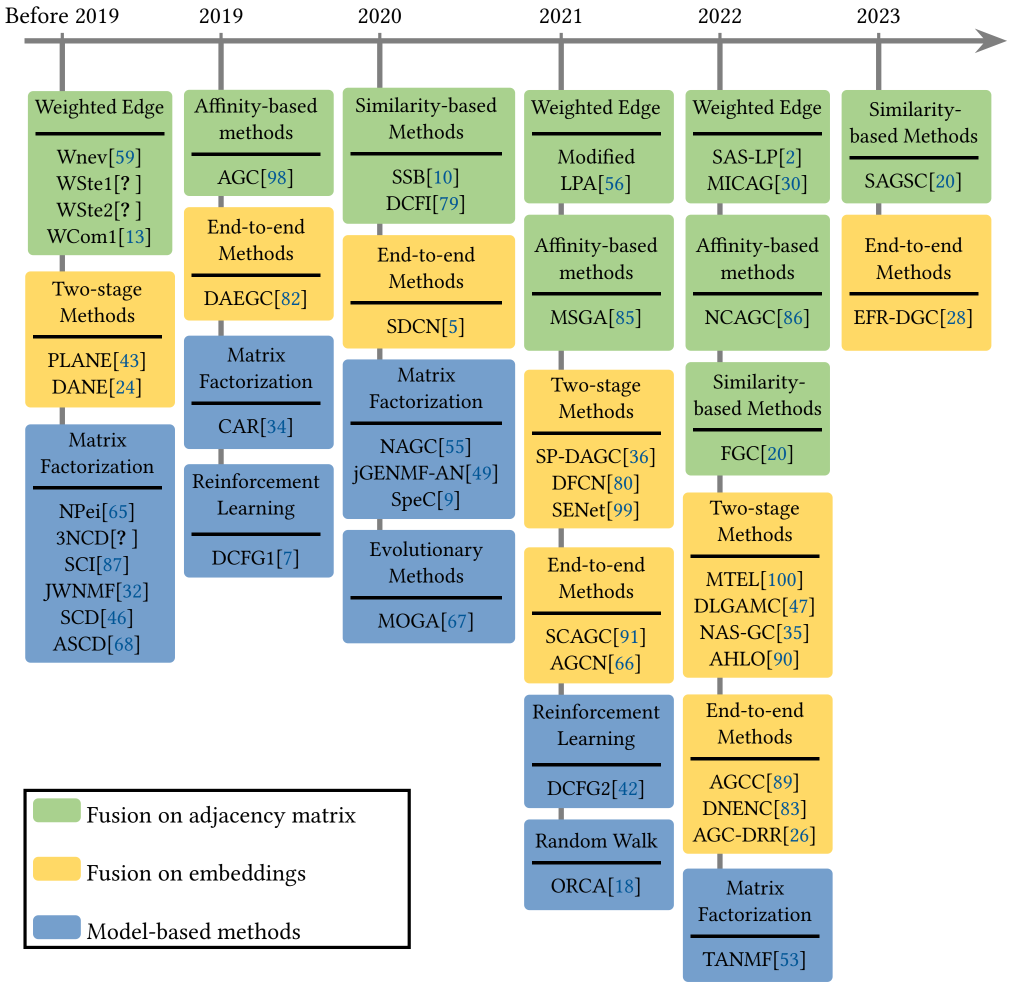


Fig. 2. A timeline of attributed graph clustering development.

framework. These methods integrate the node representation and clustering process into a unified framework.

Besides, some works in References [42, 67, 95] design special algorithms and fuse attribute information and structural information into the unique concepts within the algorithm, such as reinforcement learning, evolutionary methods, matrix factorization [32, 61, 65, 87], and so on.

Through adequate research and analysis of existing works, we divide the relevant works into the following three categories:

- **Fusion on adjacency matrix methods.** These methods always convert attributes to the adjacency matrix by calculating similarities between nodes. Then, clustering methods will be conducted on the updated adjacency matrix to realize clustering.
- **Fusion on embedding methods.** These methods integrate structural information into attribute information by combining attributes of adjacency nodes. The updated embeddings fusing the structural information with attributes can contribute to the accuracy of clustering.

Table 1. Compared with Existing Surveys

Survey	Year	Graph Type	Criterion of Division
[6]	2015	Attributed graphs	Edge or node-attributed graphs
[12]	2019	Attributed graphs	Fusion sequence
[76]	2021	All types of graphs	Deep learning technology
Ours	2023	Attributed graphs	Fusion results

- **Model-based methods.** Model-based methods usually design a specific algorithm to realize clustering, thus, these methods combine attributes and structural information into unique concepts, such as reinforcement methods that integrate attributes and structural information as a reward function to optimize the clustering results.

Comparison with existing reviews. In our investigation, there are three surveys [6, 12, 76] related to graph clustering in recent years. We compare these surveys with ours from different aspects, as shown in Table 1. The first survey [6] was published in 2015, and it focused on the attributed graphs with either edge or node attributes. Another survey [12] was published in 2019, and this article paid attention to node-attributed social networks. This survey classified methods according to when to fuse attributes and structural information. The survey presented in Reference [76] considered all types of graphs and paid attention to the deep-learning methods. In the past few years, many approaches based on different ideas and techniques related to attributed graphs have emerged in this field. Thus, we sort out recent methods and make a summary of various methods.

This article gives a comprehensive and up-to-date overview of attributed graph clustering and classifies recent articles according to a novel classification standard. Furthermore, we evaluate recent methods on attributed graph clustering by various experiments and explore the relationships between multi-view and single-view clustering tasks through the experiment results. This article is intended for general researchers who want to learn about attributed graph clustering, graph clustering researchers who want to track the latest advances in single-view or the nascent development of multi-view, and domain experts who want to extend existing clustering methods to multi-view graph clustering. The main contributions of this survey are summarized as follows:

- **Unified framework and systematic classification.** In this article, we propose a new systematic taxonomy for this survey. For each category, we overview related methods and summarize each category into a unified framework.
- **Related experimental evaluation.** We perform a thorough analysis and comparison of different methods through various experiments. Four single-view datasets and five multi-view datasets are selected to evaluate related methods.
- **Abundant resources.** We collect abundant resources related to attributed graph clustering, including datasets of single-view tasks and multi-view tasks, evaluation metrics, and so on.
- **Future direction.** We offer insightful suggestions for future studies. We discuss challenges and open opportunities for future directions.

The remainder of this article is organized as follows: Section 2 describes the related tasks in detail and provides some related concepts and definitions. Moreover, we classify and summarize advanced methods of clustering on single-view attributed graphs in Section 3. Section 4 reviews methods related to clustering on the multi-view attributed graphs. In Section 5, we summarize the challenges of the multi-view task and attempt to learn from the single-view task to bring inspiration to the multi-view task. Section 6 summarizes the relevant datasets. Section 7 evaluates advanced methods related to attributed graph clustering with extensive experimental results.

Table 2. Notations and Descriptions

Notations	Descriptions	Notations	Descriptions
N	The number of nodes	S	Similarity matrix
X	Attribute matrix	α	Weighting factor
A	Adjacency matrix	l	The probability distribution of clustering labels
G	Attributed graph	Q	The distribution of soft labels
R	The number of views	P	The target distribution derived from Q
x_i^r	The attribute vector of node v_i in the rth view	L_{con}	The contrastive loss function
$e_{i,j}^r$	The relationships between node v_i and v_j in the rth view	L_c	The clustering loss
C	The clustering partition of the graph	λ^r	The weight for the rth view
K	The number of clusters	$\overline{X}^r$	The smooth representation by graph filtering in the rth view
D	Self-expression coefficient matrix.	$f(A_r)$	The high-order adjacency relationship
z_i	The node representation of node v_i	$(*)^T$	The transpose of the matrix
$\|\| * \|\|_2$	The l_2 norm	r	The smooth parameter
$\|\| * \|\|_F$	Frobenius norm	L_e	The reconstruction loss

Section 8 summarizes the key contributions and findings of this survey and then analyzes the future directions.

2 Preliminaries

In this section, we mainly focus on two types of attributed graphs and introduce the definitions and applications of these graphs. The used notations are summarized in Table 2.

2.1 Attributed Graphs

In this section, we classify attributed graphs into two categories, namely, single-view attributed graphs and multi-view attributed graphs, which are formally defined as follows:

2.1.1 Single-view Attributed Graphs. A basic attributed graph can be referred to as a single-view attributed graph. This graph is usually generated from one relationship. To facilitate the study, we give a precise definition of the common attribute graph.

Definition 1 (Single-view Attributed Graph). We consider a common attributed graph $G = (V, E, X)$, where $V = \{v_1, v_2, \ldots, v_N\}$ is the node set, and N is the number of nodes. Furthermore, E is the edge set of a given attributed graph G, which contains edges $e_{i,j}$ connecting nodes v_i and v_j. $X = \{x_1, x_2, \ldots, x_n\}$ represents the attribute matrix, where x_i represents the attributed vector of node v_i. The adjacency matrix $A \in R^{n \times n}$ is a general data structure to store the connecting relationship between the vertices in the graph. If there exists an edge between nodes v_i and v_j, then $a_{i,j} = 1$, otherwise, $a_{i,j} = 0$.

Example 2.1. Figure 3(a) describes the interactions around Jack on the WeChat platform. Each node represents a user of the WeChat platform, whose attributes of nodes are personal information, such as hobbies, addresses, and so on. Edges can reveal the interaction between two users. We can

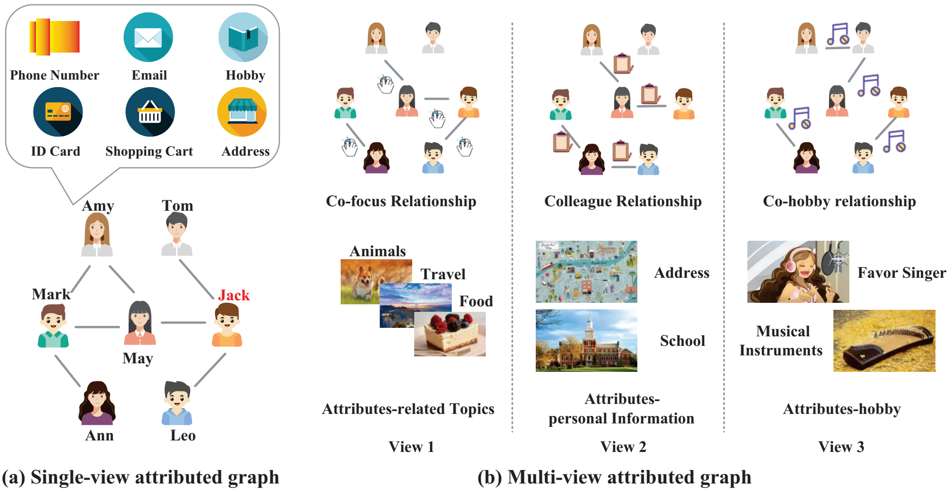


Fig. 3. An example of the single-view attributed graph and the multi-view attributed graph.

observe Jack and Leo have chatted with each other, which may indicate they are online friends, while Jack and Ann have no communication, hinting that they may not know each other.

2.1.2 Multi-view Attributed Graphs. Multi-view attributed graphs can provide different but complementary information from multiple views.

Definition 2 (Multi-view Attributed Graph). Define a multi-view attributed graph as $G = \{\Phi, E_1, E_2, \ldots, E_R, X_1, X_2, \ldots, X_R\}$, where V is the node set, $e^r_{i,j} \in E^r$ denotes the relationships between nodes v_i and v_j in the rth view $G_r(V, E_r, X_r)$, and $x^r_i \in X^r$ is the attribute matrix of node v_i in the rth view. Additionally, adjacency matrices $\{A^r\}^R_{r=1}$ for $a^r_{i,j} \in A^r$ are utilized to store the structure information of G. If there exists an edge between nodes v_i and v_j, then $a^r_{i,j} = 1$, otherwise, $a^r_{i,j} = 0$.

Example 2.2. Figure 3(b) shows a multi-view attributed graph representing different relationships in a social network. Nodes represent individuals in the social life that have different attributes and relationships in each view, providing different perspectives for us. In the co-focus relationship, attributes are composed of related topics that people care about; in the colleague relationship, attributes reflect personal information such as the member's academic background and home address; while in the co-hobby relationship, attributes include the interests and hobbies of different individuals.

2.2 Graph Clustering

Definition 3 (Single-view Attributed Graph Clustering). Given an attributed graph $G = (V, E, X)$, graph clustering aims to partition the nodes of the graph G into K clusters $\{C_1, C_2, ...C_K\}$.

Definition 4 (Multi-view Attributed Graph Clustering). Given a multi-view attributed graph $G = \left\{V, E^1, E^2, \ldots, E^R, X^1, X^2, \ldots, X^R\right\}$, the clustering on multi-view attributed graph aims to find a consensus partition fitting all views to partition the nodes into k clusters $\{C_1, C_2, ...C_K\}$.

Example 2.3. Taking Figure 3 as an example, Figure 3(a) is a single-view attributed graph of a social network, and by fusing the attributes containing personal information and structure indicating social relationships, we can segment the nodes representing people well. In the multi-view attributed graph of Figure 3 (b), we need to consider the information under three views, including co-focus, colleague, and co-hobby relationships with their related attributes, perform the fusion between the views, extract the consensus information, and the clustering result should satisfy the approval of multiple views at the same time.

2.3 Evaluation Metrics

To evaluate the quality of clustering results, we introduce the following evaluation indicators: **ACC (Accuracy)**, **NMI (Normalized Mutual Information)**, **ARI (Adjusted Rand Index)**, **F_1(F_1 − score)**, **P (Precision)**, and **R (Recall)**. There are different calculation methods and evaluation criteria for different evaluation indicators.

ACC: Accuracy [75] compares the obtained tags with the real tags that are provided by the data. Formally, it is defined as

$$ACC = \frac{\sum_{i=1}^{N} \sigma\left(s_i\ ,\ map(r_i)\right)}{N}, \tag{1}$$

where r_i and s_i represent the obtained tags and the actual tags corresponding to data x_i, N is the number of nodes, and σ indicates an indicator function:

$$\sigma(x,y) = \begin{cases} 1 & if \quad x = y\ ; \\ 0 & otherwise\ . \end{cases} \tag{2}$$

The mapping in the formula represents the recursive allocation of the best class criteria to ensure the accuracy of the statistics.

NMI: Normalized Mutual Information [75] is defined to measure the similarities between clustering results and real tags. Its value range is [0,1], and the larger the value, the more similar it is. Formally, NMI can be computed as

$$NMI\ (Y,C) = \frac{2 \times I\ (Y;C)}{H\ (Y) + H\ (C)}\ , \tag{3}$$

where Y represents the true category of the data, C is the clustering result. $H(.)$ represents cross entropy and is defined as

$$H(X) = -\sum_{i=1}^{|X|} P(i) \times log\, P(i)\ , \tag{4}$$

while $I(Y;C)$ is mutual information, $I(Y;C) = H(Y) - H(Y|C)$.

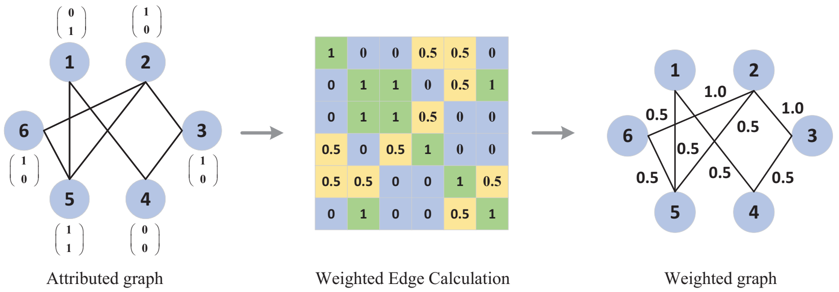
ARI: For any number of cluster centers and samples, the Adjusted Rand Index [33] of random clustering is very close to zero. Its value range is [-1,1], and a negative number represents a bad result. The closer it is to 1, the better the result is. Formally, it can be computed as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}, \tag{5}$$

where n_{ij} is the number of nodes that belong to the real cluster C_j and assignment cluster $C_i^{'}$. Respectively, a_i represents the number of nodes that belong to the assignment cluster $C_i^{'}$, and b_j is the number of nodes in real cluster C_j. Moreover, n represents the total number of nodes.

P: Precision [79] refers to the ratio of the number of correct nodes to the total number of nodes in the same cluster according to clustering results and can be computed as

$$P\ = \frac{TP}{TP + FP}, \tag{6}$$



Attributed graph Weighted Edge Calculation Weighted graph

Fig. 4. Weighted edge methods. These methods convert attributed graphs into weighted graphs by combining both attributive and adjacency relationships and adopting these combinations as the weights of edges.

where TP represents the number of nodes that belong to the same cluster according to real labels and assigned clusters, and FP is the number of nodes in different clusters according to real labels but in the same cluster in accordance with assigned results.

R: Recall [83] refers to the ratio of the number of correct answers given by the algorithm to the number of nodes that belong to the same cluster in accordance with real labels. Formally, it is defined as

$$R = \frac{TP}{TP + FN}, \tag{7}$$

where FN is the number of nodes that belong to the same cluster according to real labels but in different clusters in the light of assigned clusters.

F1-score: F1-score [62] represents the harmonic average of Precision and Recall and can be computed by

$$F_1 = 2 \times \frac{P \times R}{P + R}. \tag{8}$$

3 Single-view Attributed Graph Clustering Methods

In this article, we review related articles in recent years and classify these methods into three classes according to fusion results: fusion on adjacency matrix methods, fusion on embedding methods, and model-based methods. The fusion on adjacency matrix methods store attributed information of nodes inside the adjacency matrix. Differently, the second class stores the topology information inside the node embeddings. The third class combines attribute and structure information into particular concepts inside the proposed models. In this section, we display these three classes and representative works in detail.

3.1 Fusion on Adjacency Matrix Methods

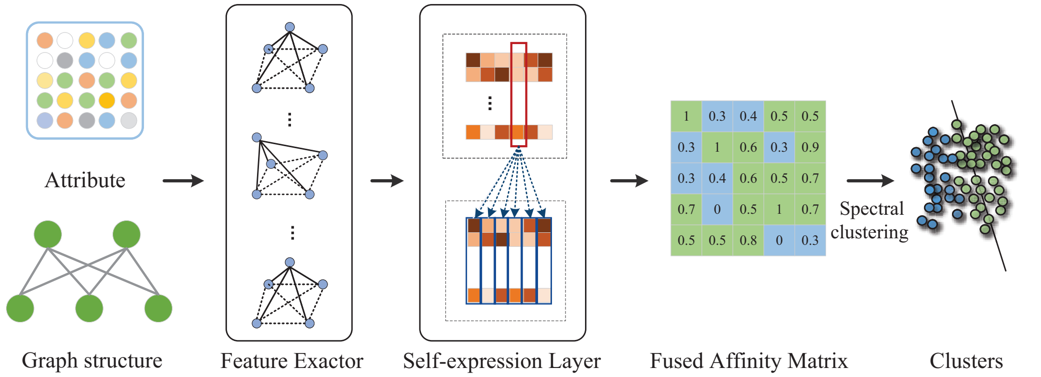
3.1.1 Weighted Edge. The works in References [2, 30, 56] convert the attributed graph $G(V, E, X)$ into a weighted graph $G(V, E)$ by combining both attributive and adjacency relationships and adopting these combinations as the weights of edges. Then, the clustering task is performed on the weighted graph instead of the attributed graph, as shown in Figure 4. Table 3 summarizes recent works related to weighted edge methods. In Reference [2], authors calculate weights of edges by combining attribute similarity and adjacency information. Here, the weights of edges can be concluded as

$$\hat{e}_{i,j} = sim(x_i\ , \ x_j)\ , \tag{9}$$

$$\tilde{e}_{i,j} = \lambda \times \hat{e}_{i,j} + (1 - \lambda) \times e_{i,j}\ , \tag{10}$$

Table 3. Recent Works Related to Weighted Edge Methods

Methods	Year	Edge Weights	Clustering Methods	Quality Measures
SAS-LP[2]	2022	The combination of attribute similarity and structural similarity	Label Propagation	NMI, ACC, F1
MICAG[30]	2022	The likelihood in the same cluster of adjacent nodes	Depth-first search	P, R, F-measure
Modified LPA [56]	2021	Edge Weighting Methods [31]	Label propagation	NMI, Modularity, Modularity, Entropy, Conductance, Density



1	0.3	0.4	0.5	0.5
0.3	1	0.6	0.3	0.9
0.3	0.4	0.6	0.5	0.7
0.7	0	0.5	1	0.7
0.5	0.5	0.8	0	0.3

Fig. 5. Affinity-based methods. They construct affinity matrix by combining attributes and structural information and then implement spectral clustering on the affinity matrix to acquire the final clusters.

where x_i denotes the attribute of node v_i and the function $sim(*,*)$ defined as Cosine Similarity or Matching Coefficient values aims to calculate the attributes similarities between nodes. Equation (10) describes the integration of both structural and attributive similarities. The hyperparameter λ contributes to the balance between attributes and structure. If $\lambda = 0$ in Equation (10), then the generated graph contains weights based on structural information; if $\lambda = 1$, then the generated graph is based on attributes. MICAG [30] obtains weights of edges by calculating the likelihood in the same cluster of adjacent nodes, and Modified LPA [56] expends the edge weighting methods [31] to continuous properties.

After we obtain the weighted graph, classical graph clustering algorithms can be utilized such as weighted Louvain, depth-first search [30], and label propagation [2, 56]. It is worth noting that attributive similarities are often determined by author preferences.

3.1.2 Affinity-based Methods. Several affinity-based methods in References [85, 86, 98] construct affinity matrix by combining attributes and structural information. After obtaining the affinity matrix (symmetric and non-negative), these methods implement spectral clustering method on the affinity matrix to acquire the final clusters, as shown in Figure 5. Table 4 summarizes recent works related to affinity-based methods. To obtain the affinity matrix, most methods presented in References [85, 86] utilize the self-expression module, while Reference [98] applies the linear kernel to learn pairwise similarity between nodes. The self-expression module is introduced by deep subspace clustering methods to learn the self-expressiveness between node representations. Specifically, given a node representation z_i, the self-expression module can represent this node

Table 4. Recent Works Related to Affinity-based Methods

Methods	Year	Number of Clusters	Affinity Matrix Construction Method	Quality Measures
NCAGC [86]	2022	Yes	Self-expression Layer	ACC, NMI, ARI
MSGA [85]	2021	Yes	Self-expression	ACC, NMI, ARI
AGC [98]	2019	Yes	Linear kernel	ACC, NMI, F1

Table 5. Recent Works Related to Similarity-based Methods

Methods	Year	Fusion Similarity Construction	Number of Clusters	Quality Measures
FGC [38]	2022	Self-expression learning	Yes	ACC, F1, NMI
SSB [10]	2020	Weighted fusion	Yes	ACC, Entropy, NMI
AGPFC [29]	2019	Weighted fusion	Yes	Density, Entropy
SAGSC [20]	2023	Self-expression learning	Yes	ACC, ARI, NMI

embedding via a linear combination of other node representations, where the reconstructed node representation z_i can be expressed as:

$$z'_i = \sum_{i \neq j} d_{ij} z_j, \tag{11}$$

where d_{ij} is the weight for node embedding z_j to represent node representations z_i. Thus, D is the self-expression coefficient matrix. Moreover, the coefficient matrix D can be obtained by minimizing the self-expression loss as follows:

$$L = \min ||Z - ZD||_F^2 + ||D||_2, \tag{12}$$

where D is the self-expression coefficient matrix consisting of each linear combination coefficient, $|| * ||_2$ represents the l_2 norm, and $||D||_2$ is a regularization term.

3.1.3 *Similarity-based Methods.*

We have introduced several methods that convert attributive information to structural information by constructing weighted edges and learning the adjacency matrix for spectral clustering in the above parts. Then, similarity-based methods are described that construct a new similarity matrix by combining the structural similarity and the attributed similarity. Table 5 summarizes recent works related to similarity-based methods. SSB [10] and AGPFC [29] combine attributive similarity and structure similarity by linear weighting. The combined similarity can be represented as follows:

$$S(v_i, v_j) = \alpha \times s_{str}(v_i, v_j) + (1 - \alpha) \times s_{att}(v_i, v_j), \tag{13}$$

where v_i represents the ith node, $s_{str}(v_i, v_j)$ is the structural similarities between nodes v_i and v_j, and $s_{att}(v_i, v_j)$ measures the attributed similarity. In the above formula, α is the weighting factor ranging from 0 to 1.

In SSB [10], $s_{str}(v_i, v_j)$ is obtained from the original adjacency matrix, and $s_{att}(v_i, v_j)$ is calculated by defining the similarities between node attributes. SAGSC [38] and FGC [20] utilize self-expression learning to update the new similarity matrix, whose details of self-expression methods are described in Equation (11).

3.2 Fusion on Embedding Methods

3.2.1 *Two-stage Methods.*

As deep learning methods develop, many researchers attempt to utilize the powerful ability to represent graphs and implement the clustering task on the features. Several

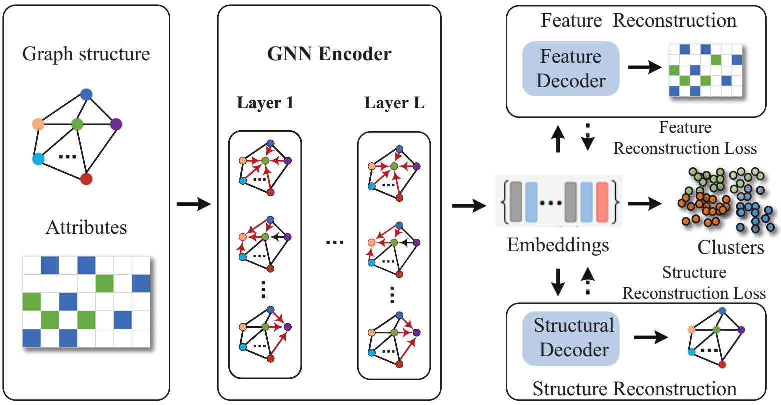


Fig. 6. A general framework for two-stage methods. These methods usually contain two stages: the embedding step and the clustering step. First, they obtain embeddings by optimizing the loss functions. Then, clustering methods are applied to optimized embeddings.

Table 6. Recent Works Related to Two-stage Methods

Title	Year	Loss	Embedding Technique	Number of Clusters	Quality Measures
MTEL [100]	2022	Multi-task prediction	GCN	Yes	ACC, ARI, NMI
DLGAMC [47]	2022	Self-supervised loss	GAT	Yes	ACC, F1, NMI, ARI
NAS-GC [35]	2022	Self-supervised loss	GCN	Yes	ACC, NMI, DM, F1
AHLO [90]	2022	Regularized harmony loss	GAT	Yes	NMI, ACC, ARI
SP-DAGC [36]	2021	Reconstruction, Self-separation regularization	Graph-AE	No	ACC, F1, ARI, NMI
DFCN[80]	2021	Reconstruction, Self-separation regularization	Graph-AE	Yes	ACC, NMI, ARI, F1
SENet[99]	2021	Spectral clustering loss	GCN	Yes	ACC, NMI, ARI

two-stage methods in References [35, 36, 47, 80, 100] usually contain two stages: the embedding step and the clustering step. At first, the topology information and attributes are encoded as embeddings by GNN or autoencoder. Second, the traditional clustering method like the k-means method is conducted on the embedded vectors to attain the final clusters, as shown in Figure 6. Significantly, the first step aims to extract useful information via the designed losses. The related training losses mainly include reconstruction loss, self-supervision loss, and so on. These losses will be described in detail as follows.

In Table 6, we summarize recent two-stage approaches. SP-DAGC [36] and DFCN [80] utilize the reconstruction losses that contain both feature reconstruction loss and structural reconstruction loss. The reconstruction loss of feature aims to learn node representations that can recover the original data. Differently, the structural reconstruction loss guarantees that the learned node representations can maintain the linkage relationships between each pair of nodes. The reconstruction loss can be defined as follows:

$$L = ||X - AE_1(X)||^2 + ||A - AE_2(A)||^2, \tag{14}$$

where X is the feature matrix, A represents the adjacency matrix, and $AE(*)$ denotes the process of autoencoder.

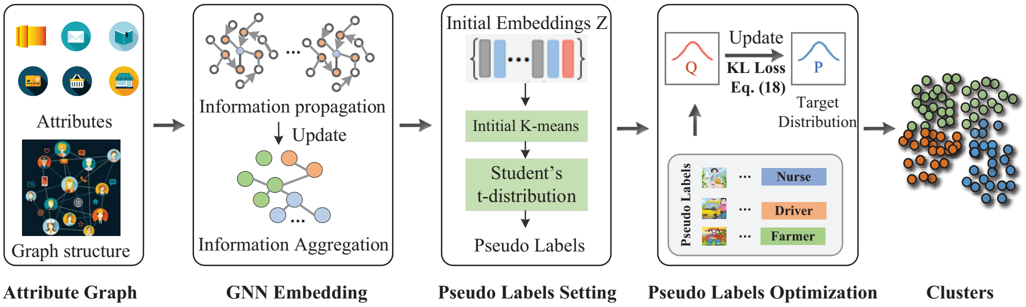


Fig. 7. A general framework for end-to-end methods. These methods optimize the node embeddings and the pseudo labels simultaneously through the objective loss function and output the finally optimized results of the pseudo labels as the clustering labels.

In References [47] and [35], the self-supervision loss aims to reduce the distances among nodes that belong to the same class and meanwhile enlarge the distances among nodes in the different classes. Furthermore, the initial partition is performed by calculating the similarities between the initial node embeddings, and then the node embeddings are continuously optimized through the training process. The self-supervision loss can be defined as:

$$Loss = \sum_{c \in C} \alpha \sum_{v_i \in c, v_j \notin c} ||z_i - z_j||_2 - \sum_{c \in C} \beta \sum_{v_i, v_j \in c, v_j \neq v_i} ||z_i - z_j||_2, \tag{15}$$

where C represents the cluster partition of all nodes, α, β are loss weights, and z_i, z_j are node representations of node v_i and v_j, respectively.

Unlike above works in References [35, 36, 47, 80], MTEL [100] constructs two prediction tasks to obtain embedding that considers node correlation and is capable of downstream clustering. AHLO [90] models node representations using a mixture of von Mises-Fisher distributions on a unit hypersphere. SENet [99] introduces the spectral clustering loss to obtain node embeddings more suitable for clustering. Although the loss functions adopted in these methods are different, they are all designed to obtain embeddings more suitable for clustering.

3.2.2 End-to-end Methods. The two-stage methods fail to obtain effective node representations for downstream node clustering task due to the separation of the clustering task and node embedding. As shown in Figure 7, end-to-end methods adopt a strategy that optimizes clustering allocation probability and node embedding simultaneously to solve this problem. The end-to-end methods utilize objective loss function to optimize the node embeddings and pseudo labels simultaneously and then consider the optimized pseudo labels as clustering results.

According to Table 7, some works presented in References [26, 89, 91, 101] utilize graph contrastive loss to optimize node embeddings and the pseudo labels.

The graph contrastive loss aims to maximize similarities among positive samples and meanwhile maximize distances among negative samples. Mutual information maximization can be employed to estimate the consistency between sample pairs, and the graph contrastive learning objective is expressed as:

$$I(Z^1, Z^2) = \frac{1}{N} \sum_{i=1}^{N} log \frac{exp(sim(z_i^1, z_i^2))}{\sum_{j=1, j \neq i}^{N} exp(sim(z_i^1, z_j^2))}, \tag{16}$$

where Z^1 and Z^2 represent the node representations of node v_i in view 1 and view 2, respectively. And $sim(*)$ is a function to calculate similarities between node representations.

Table 7. Recent Works Related to End-to-end Methods

Title	Year	Loss	Embedding Technique	Number of Clusters	Quality Measures
EFR-DGC [28]	2023	Self-optimizing loss Reconstruction loss	GAT	Yes	ACC, F1, NMI, ARI
AGCC [89]	2022	Contrastive Learning	GCN	Yes	ACC, F1, NMI, ARI
DNENC [83]	2022	Reconstruction loss, Self-optimizing loss	GAT	Yes	NMI, ACC, F, P, R, AE, ARI
AGC-DRR [26]	2022	Contrastive Learning	GCN	Yes	ACC, ARI, NMI, F1
Graph Debiased Contrastive Learning [101]	2021	Contrastive Learning	GCN	Yes	ACC, ARI, NMI
SCAGC [91]	2021	Contrastive Learning	GNN	Yes	ACC, NMI, ARI, F1
AGCN [66]	2021	Self-optimizing loss	GCN	Yes	ACC, ARI, NMI, F1
SDCN [5]	2020	Self-optimizing loss	GCN	Yes	ACC, ARI, NMI, F1
DAEGC [82]	2019	Self-optimizing loss	GAT	Yes	ACC, ARI, NMI, F1

AGC-DRR [26] and Graph Debiased Contrastive Learning [101] leverage two stochastic augmentation functions to generate two different views, defined as A, B. They transform the node representations Z^a, Z^b into probability distribution of clustering labels $l^a, l^b \in R^{n \times K}$ through clustering module, where $l^a_{i,k}$ represents the probability that the ith node belonging to kth class and K is the number of classes. And each column l^a_k of l^a is the representation of the kth cluster. Note that these methods will make representations of the same class more similar and increase the distances of representations in different classes. Thus, they randomly sample a cluster representation l^a_k in view A as an anchor, and the cluster representation l^b_k in view B is the positive example. On the contrary, the other cluster representations in the two views are negative examples. Then, the graph contrastive loss function is defined as follows according to Equation (8):

$$L_{con} = -log \frac{exp(sim(l^a_k, l^b_k)/\tau)}{\sum_{j=1}^{K} [exp(sim(l^a_k, l^a_j)/\tau) + exp(sim(l^a_k, l^b_j)/\tau)]}, \tag{17}$$

where τ is the instance temperature parameter and $sim(*)$ is the cosine similarity.

In Reference [91], unlike References [26] and [101], which adopt graph contrastive learning for the probability distribution of clustering labels, it utilizes graph contrastive learning for embedding representations and learns node embeddings by comparing positive and negative samples. Differently, AGCC [89] proposes instance contrastive loss and cluster contrastive loss to utilize graph contrastive learning for node representations and clustering assignment probability.

Several methods presented in References [5, 28, 66, 82, 83] adopt the self-optimizing loss to guide the process of clustering. They input the initial node embeddings into clustering module that utilizes the KL divergence to optimize the node embedding and obtain the estimated labels. The KL divergence in self-optimizing loss aims to calculate the similarity of distributions P and Q, where Q is the distribution of assigned labels measured through student's t-distribution and P is the target distribution originating from Q. With the target distribution P used as the ground-truth labels, the clustering loss constructed from the KL divergence is defined as:

$$L_c = KL(P||Q) = \sum_i \sum_u p_{iu} log \frac{p_{iu}}{q_{iu}}, \tag{18}$$

where i represents the ith node, u denotes the uth class, and q_{iu} measures the similarity of node embedding z_i and the center of the uth cluster μ_u. Moreover, q_{iu} is measured by student's

Table 8. Recent Works Related to Model-based Methods

Title	**Year**	**Model Type**	**Number of Clusters**	**Quality Measures**
DCFG2 [42]	2021	Game theory	Yes	Density, Entropy, F1
ACMin [95]	2021	Random walk model	Yes	AAMC, Modularity, CA, NMI
MOGA [67]	2020	Evolutionary Methods	Yes	NMI, CNMI
EMSC [16]	2020	Evolutionary Methods	Yes	Density, Mutation, ClustExp
PWMA\PLCA [1]	2020	Mathematical programming	No	RI, NMI
DCFG1 [7]	2019	Game theory	Yes	Density, Entropy, Average F1

We refer to "Evolutionary mining of skyline clusters of distributed graph data" as "EMSC" for short.

t-distribution:

$$q_{iu} = \frac{(1 + ||z_i - \mu_u||^2)^{-1}}{\sum_k (1 + ||z_i - \mu_k||^2)^{-1}}, \tag{19}$$

where z_i is the node representation of node v_i, and μ_u represents the center of the uth cluster. The student's t-distribution q_{iu} can be viewed as a distribution that assigns each node to different classes according to current node embeddings. The target distribution p_{iu} can be expressed as:

$$p_{i,u} = \frac{q_{iu}^2 / \sum_i q_{iu}}{\sum_k (q_{ik}^2 / \sum_i q_{ik})}. \tag{20}$$

It is worth noting that the clustering results for node v_i that are directly acquired from the optimized Q can be defined as:

$$l_i = \arg\max_u q_{iu}, \tag{21}$$

where q_{iu} is the possibility that node v_i belongs to the uth class.

Compared with DAEGC [82], which only employs a simple attention autoencoder to learn node embeddings, DNENC [83] proposes a neighbor-aware autoencoder to better integrate neighbor node attributes. Moreover, to better combine structural information and attributes, SDCN [5] adopts a basic autoencoder and GCN, which are unified by a dual self-supervised mechanism. AGCN [66] employs the dual self-supervised mechanism of SDCN [5] and further proposes a scale-wise fusion module that can accumulate multi-scale features. Differently, compared with SDCN [5], EFR-DGC [28] utilizes the reconstruction losses of two autoencoders (the **graph-autoencoder (GAE)** and a basic autoencoder) and the self-optimizing loss to efficiently guide model updates.

3.3 Model-based Methods

In the previous section, we introduce methods fusing attribute information and structural information to adjacency relationships and embeddings, respectively. However, there exist some methods that cannot fuse structural information and attributes to either adjacency relationships or embeddings. These methods provide a specific model or algorithm to realize the clustering task on attributed graph. Meanwhile, they integrate structural and attribute information into unique concepts within the model or algorithm, such as methods based on game theory, evolutionary methods, random walk models, and so on. Table 8 concludes recent model-based methods.

3.3.1 Game Theory. Game theory is widely applied in various fields. This is an ideal scenario for game theory analysis, where users interact and make rational decisions to ultimately realize their goals. Thus, to apply game theory to clustering problems, in References [42] and [7], nodes can usually be regarded as participants, strategy is defined as selecting from different clusters, and the cohesion and similarity obtained by transferring from one group to another are defined as rewards. The equilibrium of the game represents a stable clustering result. Moreover, these

Table 9. Recent Works Related to Matrix Factorization–based Methods

Title	Year	Compression Type	Number of Clusters	Quality Measures
TANMF [53]	2022	2-factor NNMF	Yes	NMI, Kappa
ORCA [18]	2021	2-factor NNMF	Yes	NMI, RI
NAGC [55]	2020	2-factor NNMF	Yes	ARI, AMI
jGENMF-AN [49]	2020	2-factor NNMF	Yes	NMI
SpEC [9]	2020	2-factor NNMF	Yes	F1, Jaccard, AC, NMI
CAR [34]	2019	2-factor NNMF	Yes	ACC

methods based on game theory usually define a special utility function, combining structural information with attributes for attributed graphs. The utility function is the particular definition in game theory–based methods. Each player has their own utility function and aims to maximize their utility value and quality function. Furthermore, Reference [7] considers that clusters are not necessarily disjoint compared with Reference [42].

3.3.2 Evolutionary Methods. Genetic algorithm is a biomimetic algorithm that attempts to find the optimum results via the theory of biological evolution. It compiles parameters into genes that are presented as binary or decimal code to simulate gene recombination and evolution in natural processes. The evolution of numerous chromosomes derived from multiple genomes resembles the process of biological evolution, which is conducted repeatedly until a final outcome is achieved. To utilize both structural and attribute information simultaneously, methods based on genetic algorithms have redefined the fitness function. In Reference [16], the fitness function is the special definition of genetic algorithm that can aggregate both structural and attribute information. Compared with Reference [16], MOGA [67] combines structure and attributes into the multiobjective functions.

3.3.3 Random Walk Model. The random walk–based model can effectively capture the multi-hop relationships. For example, as a common model, **random walk with restart (RWR)** utilizes the restart probability to capture the multi-hop relationships better. However, the random walk–based cannot process the attributed graph, since it only considers topology information. To solve this issue, ACMin [95] defines an attributed transition probability and a topological transition probability to take both attribute and structural information into account.

3.3.4 Matrix Factorization. **Non-negative matrix decomposition (NMF)** has developed rapidly in the field of graph clustering. In Table 9, several matrix factorization methods in References [9, 18, 34, 49, 53, 55] decompose the adjacency matrix or attribute matrix and attempt to get the optimal cluster membership matrix that conforms to both adjacency and attribute relationships, as shown in Figure 8. In Reference [53], the adjacency matrix A is decomposed into two nonnegative matrixes $F_1 \in R^{N \times K}$ and $B \in R^{K \times N}$, where F_1 represents the basis matrix and B represents the cluster membership matrix. Moreover, the attribute matrix X is also decomposed into two nonnegative matrixes $F_2 \in R^{N \times K}$ and B. The unified objective function can be expressed as:

$$Loss = ||A - F_1 B||_F^2 + ||X - F_2 B||_F^2. \tag{22}$$

The above objective function utilizes both attributes and topology information to obtain the cluster membership matrix that is consistent in the attribute relation and adjacency relation. The cluster label for node v_j can be acquired as:

$$q = \arg \max_{1 \le i \le K} B_{ij}, \tag{23}$$

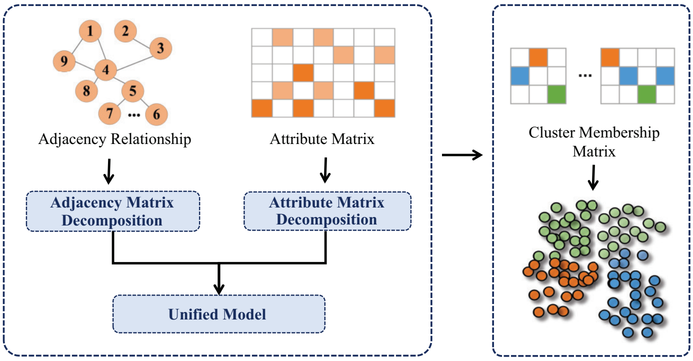

1
2
3
4
5
6
7
8
9
Adjacency Relationship
Attribute Matrix
Adjacency Matrix Decomposition
Attribute Matrix Decomposition
Unified Model
Cluster Membership Matrix

Fig. 8. A general framework for matrix factorization–based methods. These methods decompose the attribute matrix and adjacency matrix and attempt to get the optimal clusters by combining these two objective functions.

where K is the number of clusters and q is the obtained label. Compared with TANMF [53], ORCA [18] aims to deal with the problem that sparse and low-rank input can lead to an inordinate number of outliers. Moreover, jGENMF-AN [49] utilizes dimensionality reduction to extract attributes and adopts a smoothing process to reduce the heterogeneity of attributes and topology.

3.4 Discussions

In this section, we classify recent single-view attributed graph methods into three categories according to fusion results: fusion on adjacency matrix methods, fusion on embedding methods, and model-based methods. Unlike fusion on adjacency matrix methods pay more attention to storing attributed information in graph structure, fusion on embedding methods utilize GNN to integrate information from neighboring nodes that can store topology information into node embeddings. Moreover, model-based methods can store structural information and attributes into unique concepts.

Fusion on adjacency matrix methods. Three subcategories have different characteristics. Weighted edge methods convert the attribute graph into a weighted graph and implement label propagation or Louvain methods to obtain clusters, unlike similarity-based methods that just consider both structure and attribute information but still retain the property of the updated attribute graph to cluster via the classical distance-based clustering method. The definition of similarity between attributes is the key point and aporia of weighted edge methods. By fusing attribute similarity with existing adjacency information, edge weights can be effectively updated. Existing attribute similarity methods mostly rely on the cosine distance, and a more accurate and rapid method for measuring attribute similarity is still expected to be proposed. Furthermore, in existing methods, the number of clusters has a significant impact on clustering results, and most methods require the number of clusters as an input value. To our knowledge, there are few articles for clustering that do not require the given number of clusters. However, we observe that by updating label propagation methods or Louvain methods, weighted edge methods can effectively avoid the need for the number of clusters, which can contribute to unsupervised clustering.

The second subcategory (affinity-based method) is similar to similarity-based methods, which both construct a matrix indicating similarities among nodes. However, compared with similarity-based methods, the requirements for constructing an affinity matrix in the second subcategory are more stringent. Similarity-based methods construct a similarity matrix by calculating attribute similarities and add them to the original adjacency matrix, while the affinity matrix is usually obtained through self-expression modules and has both non-negative and symmetric properties. In addition, the affinity matrix is used for spectral clustering, while similarity-based methods use k-means and other custom clustering methods without these serious limitations. Moreover, these two methods have a smaller number of parameters and a simpler process compared to fusion on embedding methods based on deep learning and can be well extended to multi-view attributed graph clustering.

Fusion on embedding methods. In fusion on embedding methods, compared with two-stage methods that implement GNN as the feature exactor, end-to-end methods take node representation learning and clustering into a unified framework. However, these two subcategories are designed based on deep learning and have a large number of parameters. Faced with challenges of large graph and multi-view tasks, they may require model compression, pruning, and other operations to reduce costs and improve efficiency.

Model-based methods. In model-based methods, we observe that game theory methods can be extended to deal with fuzzy clustering, which can discover joint clusters. Non-negative matrix factorization can be applied to each single view separately and then obtain the consensus graph by fusing decomposition results of different views, ultimately achieving multi-view clustering.

4 Multi-view Attributed Graph Clustering Methods

This section focuses on the methods of clustering multi-view attributed graphs, which contain fusion on adjacency matrix methods and fusion on embedding methods. These methods aim to find a cluster partition that is consistent in all views. Thus, several methods attempt to construct a consensus graph from all views and then conduct a clustering task in the consensus graph. In this section, we will describe related works of the fusion on adjacency matrix methods and fusion on embedding methods.

4.1 Fusion on Adjacency Matrix Methods

To cluster multi-view attributed graphs, it is extremely important to obtain consistent information between different views and conduct the consensus graph. That is, all views share the same clustering results and further enjoy the same similarity matrix, whereas each view may play a different role in the clustering process.

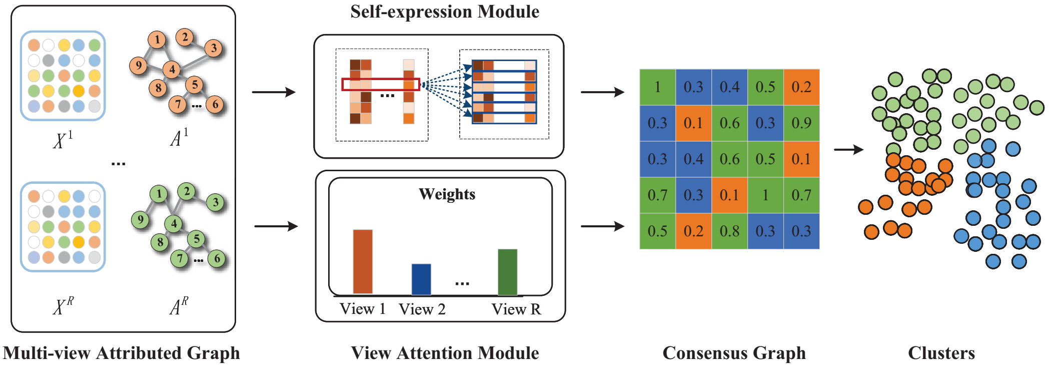
To obtain the same similarity matrix S in all views, Reference [51] utilizes the self-expressiveness property and optimizes the following objective function:

$$\min_{S,\lambda^r} \sum_{r=1}^{R} \lambda^r \left(|| \left(\overline{X}^r\right)^T - \left(\overline{X}^r\right)^T S||_F^2 + \alpha ||S - f(A_r)||_F^2 \right) + \sum_{r=1}^{R} (\lambda^r)^\alpha, \tag{24}$$

where λ^r is the weight for the rth view, $\overline{X}^r$ is the smooth representation by graph filtering in the rth view, $f(A_v)$ is the high-order adjacency relationship, and $\alpha < 0$ is the a smooth parameter. Through the optimization of the objective function, we can acquire the consensus similarity matrix and then implement a clustering module to obtain final results, as shown in Figure 9. Compared with MAGC [51], MCGC [64] adopts graph contrastive learning to learn a consensus graph, and MvAGC [50] chooses some anchors to reduce the computation complexity. Table 10 summarizes recent works related to the fusion on adjacency matrix methods.

4.2 Fusion on Embedding Methods

Fusion on embedding methods presented in References [19, 48, 52] usually train a series of encoders or auto-encoders via the reconstruction loss and, moreover, provide a module to obtain consensus embeddings through consistent loss. Then, we can conduct the traditional clustering algorithm



1	0.3	0.4	0.5	0.2
0.3	0.1	0.6	0.3	0.9
0.3	0.4	0.6	0.5	0.1
0.7	0.3	0.1	1	0.7
0.5	0.2	0.8	0.3	0.3

Fig. 9. The fusion on adjacency matrix method for multi-view attributed graphs.

Table 10. Recent Works Related to the Fusion on Adjacency Matrix Methods for Multi-view Attributed Graphs

Title	Year	Number of Clusters	Specialty	Quality Measures
MAGC [51]	2021	Yes	High-order adjacency information	ACC, F1, NMI, ARI
MCGC [64]	2021	Yes	Contrastive Clustering	ACC, NMI, ARI, F1
MvAGC [50]	2021	Yes	Anchor Selecting Strategy	ACC, F1, NMI, ARI

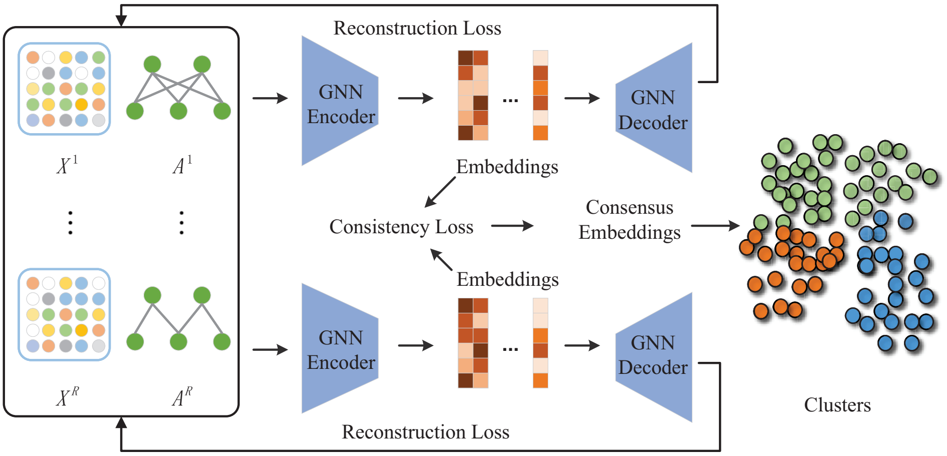


Fig. 10. Fusion on embedding methods for multi-view attributed graphs.

like k-means and others on the consistent embeddings to acquire clustering results, as shown in Figure 10. In particular, LMGEC [21] designs a linear graph autoencoder to reduce running time. Table 11 summarizes recent works related to fusion on embedding methods. Furthermore, we introduce frequently used losses for multi-view attributed graph clustering.

The reconstruction loss in single-view methods is defined as Equation (25), while in multi-view clustering methods presented in References [19, 48, 52], it can be considered as the sum of reconstruction loss in each view:

Table 11. Recent Works Related to Fusion on Embedding Methods for Multi-view Attributed Graphs

Methods	**Year**	**Loss**	**Embedding Technique**	**Number of Clusters**	**Quality Measures**
O2MAC [19]	2020	Reconstruction loss, Self-optimizing loss	Graph-AE	Yes	ACC, ARI, NMI, F1, Modularity
DIAGC [48]	2022	Reconstruction loss, Self-optimizing loss	GCN	Yes	ACC, ARI, NMI, F1
MGCCN [52]	2022	Reconstruction loss, Self-optimizing loss	GCN	Yes	ACC, ARI, NMI
LMGEC [21]	2023	Reconstruction loss	Linear Graph-AE	Yes	ACC, ARI, NMI, F1

$$L_e = \min_{\theta} \sum_{r=1}^{R} ||X_r - \widetilde{X_r}||_F^2, \tag{25}$$

where θ is the parameter of multi-view attributed graph encoders, R is the number of views, and $\widetilde{X_r}$ is the reconstruction of X_r.

The consistent embedding T usually represents node representations of the consensus view. T differs from S in Equation (24). S $\in R^{n\times n}$ reflects the similarity relationships among nodes in the consensus view, while T $\in R^{n\times d}$ serves as the representation of nodes in the consensus view. Consistency loss attempts to maximize the similarity between the embeddings of each view and the consistency embeddings. In Reference [48], authors utilize mutual information maximization to obtain the consistent embeddings:

$$I(Z^r, T) = \frac{1}{N} \sum_{j=1}^{N} log \frac{exp(sim(z_j^r, t_j))}{\sum_{j'=1, j' \neq j}^{N} exp(sim(z_j^r, t_{j'}))}, \tag{26}$$

where z_j^r represents the node representation of node v_j in the rth view, N represents the number of nodes, and T is the consistent embedding. To maximize similarities between the embeddings in each view and the consensus embedding, the consistency loss can be defined as:

$$L = \sum_{r=1}^{R} I(Z^r, T), \tag{27}$$

where R is the number of views, $I(Z^r, T)$ is the mutual information defined in Equation (26), and r is the rth view.

4.3 Discussion

In this section, we conclude methods clustering on multi-view attributed graphs and classify them into two classes: the fusion on adjacency matrix methods and the fusion on embedding methods. The proposed classification standard is also available for clustering multi-view attributed graphs. For example, MAGC and MCGC belong to the affinity-based method, one of the adjacency matrix fusion methods. O2MAC and MGCCN belong to the end-to-end methods, while DIAGC belongs to the two-stage methods, which are all fusion on embedding methods.

Multi-view graphs pose great challenges to existing methods, and with the development of multi-view large graphs, it is urgent to propose more efficient and low-cost clustering methods. However, through our investigation, there is currently only one article, presented in Reference [50], that involves reducing model complexity and running time. It will be interesting to fill this gap in the future study. To deal with this challenge, the fusion on embedding methods can adopt model pruning and model compression to improve efficiency. For the fusion on adjacency matrix methods, more effective anchor methods or graph sampling techniques can be utilized to accelerate clustering. Moreover, faced with the problem that GNN is designed for single-view tasks, we need to design a specific GNN more suitable for multi-view rather than the superposition of multiple encoders. Furthermore, existing fusion on adjacency matrix methods do not integrate node representation and clustering into one process, leading to ineffectively extracting information that is conducive to clustering. Therefore, it is expected to integrate multi-view information extraction and clustering objectives into a unified framework in the future study.

5 Challenges and Solutions: Learning from Single-view to Multi-view

5.1 Challenges for Clustering Single-view Attributed Graphs

- **Complex scenarios.** There exist overlapping phenomena in many real communities, where each node may be associated with multiple communities. Therefore, it is expected to extend existing methods to fit overlapping clustering. In addition, as multi-view data becomes increasingly widespread, it is urgent to develop existing methods for multi-view tasks. Moreover, homogeneity is difficult to guarantee in real organizational structures, so it is challenging to cluster in heterogeneous systems. Existing methods are supposed to be extended in these complex scenarios.
- **New trends.** With the growth of massive data and the increasing scale of graphs, it is needed to expand existing methods to large-scale scenarios. It is difficult to minimize the consumption of space and time while ensuring the accuracy of the model, which poses a significant challenge to existing research.

5.2 Challenges for Clustering Multi-view Attributed Graphs

The existing methods of clustering on multi-view attributed graphs can be divided into two classes. The first one incorporates multiple graphs into a consensus graph and employs a single-view algorithm on the consensus graph. The other methods extract node representations through GNN and then implement classical clustering methods on the learned node representation. Based on existing methods, we conclude the related challenges and problems.

- **Efficient consensus representations.** Several methods obtain clusters from the consensus graph, which is trained to reconstruct adjacent matrices in different views. However, multi-view clustering aims to obtain consensus clusters, whereas the reconstruction from the consensus graph focuses on the diversities between different views, leading to conflicts between the two tasks.
- **Specific GNNs.** GNN is designed mainly for single-view models. It is an extremely important problem to extend GNN to multiple views. Some methods employ a series of encoders and decoders to extract information from each view. However, these solutions lead to a large number of parameters and calculations or even introduce more noise to clustering. Therefore, this direct approach is not effective.
- **The end-to-end framework.** Most node embeddings aim to reconstruct the original graph, so they are not suitable for clustering. Therefore, we need to put the optimization of node embeddings and clustering in a unified framework.
- **The balance between consensus and diverse information.** Nowadays, most methods only utilize consensus information, but different views can also provide unique information as complementary information, so how to extract and utilize complementary information effectively becomes a problem.

5.3 Solutions Learning from Single-view to Multi-view

The past few decades have witnessed the vigorous development of methods related to clustering single-view attributed graphs. However, unlike single-view clustering methods, the clustering methods on multi-view attributed graphs are still at a nascent stage. Thus, it is essential for us to learn some solutions from single-view clustering methods that can be extended for multi-view clustering. The following solutions can be taken into consideration:

- **Matrix factorization–based methods.** Nowadays, there exist several methods dealing with multi-view data, which directly concatenate all features into a new feature and then construct a single graph from it. However, these simple approaches ignore the relationships between all views. From a global perspective, each view can contain specific information and mutual information, thus, there is an urgent need for us to better fuse the multi-view information. We consider that the matrix factorization–based methods can be expanded to multi-view clustering. We can perform matrix factorization on the attributed graph for each view. Furthermore, all views share a factorization matrix as the consensus graph or soft label assignment matrix. If the shared matrix is a consensus graph, then clustering methods can be performed on the consensus graph. When the shared matrix is a soft label assignment matrix, the cluster labels can be obtained directly.
- **Graph contrastive learning.** In the previous section, we describe the loss based on graph contrastive learning in the single-view methods, which aims to narrow the intra-cluster distance and alienate the distances between different clusters. In the multi-view task, we can find the node representations in the consensus graph through graph contrastive losses. The representations in different views of one node can be taken as positive sample groups, while the rest can be considered as negative samples. We can extend the graph contrastive losses to the multi-view clustering task and find consistent representations.
- **More suitable node representation for cluster tasks.** Obtaining node embeddings and clustering nodes are two separate tasks. Generally, node embeddings are optimized by reconstructing the input graph, which leads to the fact that the optimized node embeddings may not be fitted for clustering. Therefore, we ought to learn from the end-to-end models like the self-optimizing loss in the single-view methods and optimize the node embeddings and clustering in a unified way.
- **Attention mechanism.** The attention mechanism of GAT is extensively employed in single-view methods and, moreover, we can extend the attention mechanism to multi-view methods. For example, when embedding nodes in each view, the information of different nodes can be weighted and fused. In different views, some views have noise or incomplete information, so different views should enjoy different importance when integrating information.

6 Datasets

6.1 Attributed Graph Clustering Datasets

In this part, we summarize the commonly used single-view attributed graph datasets, as shown in Table 12.

- **Cora**[1,2]: This is a citation graph of papers composed of 2,708 nodes. The nodes represent scientific papers, and the edges show the citation relationship among papers. Each paper refers to at least one paper or is cited by other papers. Therefore, this is a connected graph with no isolated points.

[1] https://github.com/tkipf/gae/tree/master/gae/data
[2] https://linqs.soe.ucsc.edu/data

Table 12. Attributed Graph Clustering Datasets

Datasets	Nodes	Attribute Dimension	Edges	Classes	Type	Scale
Cora	2,708	1,433	5,429	7	Paper relationship	Small
Citeseer	3,327	3,703	4,732	6	Paper relationship	Small
PubMed	19,717	500	44,338	3	Paper relationship	Large
WIKI	2,405	4,973	17,981	17	Document relationship	Small
Amazon-Photo	7,650	745	119,081	8	Purchase relationship	Medium
BlogCatalog	5,196	8,189	171,743	6	Blogger relationship	Medium
WebKB	877	1,703	1,608	4	Web page relationship	Small

- **Citeseer**[1,2]: This dataset contains 3,312 scientific publications represented as nodes. The edges show the citation relationship among papers.
- **PubMed**[1,2]: This is a dataset composed of 19,717 scientific publications. These papers belong to three categories. The edges represent the cited relationships between papers.
- **WIKI**[3]: This is a network of documents. The nodes represent documents, and the edges represent hyperlinks to web pages between documents. The attributes of each node are a term frequency-inverse document frequency matrix of this document.
- **BlogCatalog [45]:** This is a blogger community network, in which nodes represent different bloggers and the edges represent the connections between bloggers. The attributes of nodes are composed of the keywords of the blogger's blog.
- **WebKB**[4]: It is a web page network of four universities. Each node represents a page, and the edges represent hyperlinks among pages. The words appearing on the page constitute the attributes of the node.

6.2 Multi-view Attributed Graph Clustering Datasets

In this part, we summarize the commonly used multi-view attributed graph datasets, as shown in Table 13.

- **ACM**[5]: This is a paper network, where nodes represent papers and can be classified into three classes. There exist two types of relationships between nodes, namely, co-author and co-subject.
- **DBLP**[6]: This is an author network, where nodes represent authors. Various relationships are contained among nodes, namely, co-author, co-conference, and co-term.
- **IMDB**[7]: This is a movie network, where nodes represent movies and can be classified into three classes. The edge among nodes indicates they have the same actor or director.
- **Amazon Photo and Amazon Computer [72]:** They are from the Amazon co-purchase network dataset. The node represents a product, and the edges between the nodes indicate whether the two products are usually purchased together. These datasets contain a co-purchase graph and a feature matrix separately.
- **WIKI-pro [72]**: The Wiki-pro dataset that contains multiple attribute matrices and multiple adjacency matrices is generated from the single-view attributed graph dataset (WIKI). In the WIKI-pro dataset, the additional views are created from the initial data, which contains a

[3] https://github.com/albertyang33/TADW/tree/master/wiki
[4] http://linqs.cs.umd.edu/projects/projects/lbc/index.html
[5] http://dl.acm.org
[6] https://dblp.uni-trier.de/
[7] https://www.imdb.com/

Table 13. Multi-view Attributed Graph Clustering Datasets

Dataset	Views	Nodes	Attributes in Each View	Edges in Each View	Clusters
ACM	2	3,025	1,830	Co-subject (29,281) Co-author (2,210,761)	3
DBLP	3	4,057	334	Co-author (11,113) Co-conference (5,000,495) Co-term (6,776,335)	4
IMDB	2	4,780	1,232	Co-actor (98,010) Co-director (21,018)	3
Amazon Photo	2	7,487	745 7,487	Co-purchase (119,043)	8
Amazon Computer	2	13,381	767 13,381	Co-purchase (245,778)	10
WIKI-pro	4	2,405	4,973 4,973	24,357 12,025	17

single graph structure and attribute matrix. We create the second topology by constructing the nearest neighbor graph via the cosine distance and generate the second attribute matrix using a log scale of the original ones.

7 Experiment and Analysis

To analyze the performance of advanced methods including single-view and multi-view methods, we evaluate these methods in different types of datasets. We select four popular single-view attributed datasets and three multi-view graph datasets to analyze nine single-view methods that are advanced in recent years. Meanwhile, we choose three datasets (ACM, DBLP, IMDB) that have the same attributes but different adjacent relationships, one dataset (Amazon Photos) that has the same adjacent relationship with different attributes in different views, and one dataset (WIKI-pro) contains both different adjacent relationships and attributes by artificial generation.

To evaluate the existing methods more detailedly and comprehensively, we will analyze them from the following four aspects: comparisons among single-view methods, comparisons among multi-view methods, comparisons between single-view methods and multi-view methods, and efficiency comparison.

7.1 Comparisons among Single-view Methods

7.1.1 Comparisons of Single-view Methods on Single-view Attributed Graph Datasets. To analyze the performances of single-view methods, we utilize four publicly available datasets via three widely employed metrics to evaluate them. The chosen nine methods belong to different types and were published in the past four years, which we consider important in this field. The results are shown in Table 14.

Overall, across four datasets, the chosen methods perform fairly evenly on the Cora and Citeseer datasets, while they perform variously in the PubMed and the Wiki datasets, particularly poor in the Wiki datasets. This above phenomenon has possibilities related to the characteristics of each dataset. We can observe that either the PubMed dataset, which has 19,717 nodes and is much larger than other datasets, or the Wiki dataset, which has 17 classes, both present a greater challenge to clustering than other datasets. Thus, in summary, various methods can perform differently in the same dataset due to the characteristics of their own methods, and due to the characteristics of the datasets, a method may perform differently across various datasets.

Table 14. Single Methods on Single-view Attributed Graph

Dataset	Citeseer			Cora		
	ACC	NMI	F1	ACC	NMI	F1
AGC	0.6720	0.4139	0.6266	0.6894	0.5374	0.6562
DAEGC	0.6720	0.3970	0.6360	0.7040	0.5280	0.6820
FGC	0.6901	0.4402	0.6443	0.7290	0.5612	0.6327
AGC-DRR	0.6832	0.4328	0.6482	0.2145	0.0243	0.1859
DFCN	0.6750	0.4280	0.6370	0.4177	0.1949	0.2773
SAGSC	0.6600	0.4000	0.2700	0.6200	0.4600	0.1600
SDCN	0.6596	0.3871	0.6362	0.6285	0.4121	0.5530
AGCN	0.6802	0.4067	0.6132	0.6326	0.4264	0.5522
EFR-DGC	0.7050	0.4450	0.6420	0.7220	0.5280	0.7050
Dataset	PubMed			Wiki		
	ACC	NMI	F1	ACC	NMI	F1
AGC	0.6982	0.3158	0.6876	0.4760	0.4503	0.4036
DAEGC	-	-	-	0.3314	0.3198	0.1907
FGC	0.6942	0.3115	0.6889	0.5089	0.4326	0.3460
AGC-DRR	0.5174	0.1280	0.5009	0.3306	0.2995	0.2937
DFCN	0.4180	0.0287	0.2366	0.2536	0.2087	0.1508
SAGSC	0.7100	0.3300	0.4400	0.4800	0.5000	0.0300
SDCN	0.4977	0.0810	0.3740	0.1672	0.0142	0.0179
AGCN	0.4771	0.0814	0.4619	0.1672	0.0142	0.0179
EFR-DGC	0.6611	0.2680	0.6557	0.4274	0.3317	0.2541

Specifically speaking, in the Citeseer and the Cora datasets, the methods EFR-DGC and FGC perform better according to the three metrics, while in the PubMed and Wiki datasets, the methods SAGSC and FGC perform well. As we can observe, the FGC method performs stably and effectively in all these methods. In fact, although FGC and the other methods (AGC and SAGSC) correspond to different method classes, they have similarities with each other. AGC proposes the k-oder graph convolution to smooth the features of nodes in each dimension. FGC not only utilizes the k-oder graph convolution but implements the self-expression and the high-order adjacency to improve the performance of clustering. SAGSC also adopts the self-expression and then decomposes the self-expression matrix further. Thus, their results on these four datasets are stable.

EFR-DGC performs well in the Citeseer and the Cora datasets, while it fails in the Wiki dataset. EFR-DGC has similarities with three other methods (DFCN, SDCN, AGCN), and they all belong to end-to-end methods except DFCN, which belongs to two-stage methods. All the above four methods integrate two deep neural structures for better embedding, with a brief that the graph neural network pays more attention to the topological structure and is inferior to extracting attribute information than other deep neural structures. We observed that EFR-DGC performs better among the four methods, which may be related to the attention mechanism in GAT, whereas the other methods all utilize GCN.

DAEGC and AGC-DRR belong to the end-to-end methods. DAEGC implements a special structure to store graphs, which requires more memory than other methods. As the number of nodes increases, it demands more memory, so it is out of memory during the PubMed dataset.

7.1.2 Results of Single-view Methods on Multi-view Attributed Graph Datasets. We utilize different single methods on each view of multi-view attributed graphs to analyze the relationships between different views and provide ideas for multi-view clustering.

Table 15. Single-view Methods on Multi-view Attributed Graph

Dataset	View	ACM			DBLP			IMDB		
		ACC	NMI	F1	ACC	NMI	F1	ACC	NMI	F1
AGC	0	0.7884	0.4849	0.7851	0.5495	0.2319	0.5444	0.5403	0.0009	0.2525
	1	0.7044	0.4881	0.6800	0.9006	0.7134	0.8949	0.5330	0.0060	0.3044
	2	\	\	\	0.6354	0.3330	0.6398	\	\	\
DAEGC	0	0.6390	0.3372	0.5928	0.4553	0.1778	0.3712	0.4059	0.0093	0.3440
	1	0.6432	0.3407	0.5986	0.8780	0.6828	0.8666	0.3736	0.0108	0.3523
	2	\	\	\	0.4368	0.1122	0.4218	\	\	\
FGC	0	0.8611	0.5793	0.8602	0.7298	0.4124	0.7260	0.5851	0.0660	0.4074
	1	0.791	0.4342	0.7851	0.9258	0.7710	0.9212	0.5715	0.0529	0.4328
	2	\	\	\	0.7328	0.4450	0.7311	\	\	\
AGC-DRR	0	0.3927	0.0189	0.0197	-	-	-	0.3661	0.0035	0.3455
	1	-	-	-	-	-	-	0.4467	0.0079	0.3180
	2	\	\	\	0.3350	0.0224	0.3295	\	\	\
DFCN	0	0.3947	0.0674	0.2634	0.3658	0.0642	0.3653	0.5052	0.0100	0.2641
	1	0.6483	0.4109	0.6616	0.8763	0.6720	0.8616	0.5471	0.0104	0.2493
	2	\	\	\	0.3333	0.0523	0.2164	\	\	\
SAGSC	0	0.8700	0.5900	0.2600	0.5900	0.2500	0.1500	0.5400	0	0.4700
	1	0.7000	0.4900	0.4500	0.8900	0.7000	0.3300	0.5400	0	0.4300
	2	\	\	\	0.5300	0.2400	0.2800	\	\	\
SDCN	0	0.9064	0.6939	0.9069	0.4491	0.1490	0.2927	0.5469	0.0288	0.3676
	1	0.6595	0.5256	0.5753	0.5891	0.3442	0.5133	0.5103	0.0484	0.4092
	2	\	\	\	0.5290	0.2668	0.4972	\	\	\
AGCN	0	0.8922	0.6682	0.8913	0.5290	0.2668	0.4972	0.5469	0.0026	0.2413
	1	0.6949	0.4539	0.6968	0.7649	0.6137	0.6387	0.5508	0.0079	0.2736
	2	\	\	\	0.5933	0.3654	0.5249	\	\	\
EFR-DGC	0	0.8760	0.6269	0.8725	0.3892	0.0835	0.3684	0.4467	0.0135	0.3856
	1	0.8079	0.5422	0.7999	0.7939	0.5624	0.7827	0.4452	0.0157	0.3956
	2	\	\	\	0.5751	0.2247	0.5383	\	\	\

We utilize "\" to indicate that this view does not exist and "–" represents the experiment is out of memory.

As we observe in Table 15, in the ACM dataset, most methods perform better in the 0th view than the 1th view, which implies there may exist more useful information in the 0th view in the ACM dataset. A few methods that do not align with this observation may result from their inability to effectively extract useful information from the views. In the DBLP dataset, it is evident that all methods perform better in the 1th view than other views. In the IMDB dataset, the performances in the two views are close to each other according to all methods. Overall, it is logical and essential to give different attention to different views in a specific task. However, the importance of views to a particular method is not always fixed but depends on its ability to abstract useful information from views.

In detail, in the ACM dataset, SDCN and EFR-DGC perform better in different views. Both in the DBLP and IMDB datasets, FGC performs best wherever in any view. In general, FGC utilizes high-order adjacency information and attributes effectively such that it can perform better in the chosen five datasets.

7.2 Comparisons among Multi-view Methods

Due to the limited number of multi-view methods, we select four existing well-known works and evaluate these methods on multi-view datasets. There are three datasets (ACM, DBLP, IMDB) that have the same attribute matrix and different adjacency matrixes in each view. Moreover, the

Table 16. The Results of Multi-view Methods

Dataset	**ACM**			**DBLP**			**IMDB**		
	ACC	**NMI**	**F1**	**ACC**	**NMI**	**F1**	**ACC**	**NMI**	**F1**
MAGC	0.8723	0.5976	0.8705	0.9307	0.7787	0.9262	0.6171	0.1181	0.4549
MvAGC	0.8882	0.6559	0.8893	0.8915	0.6836	0.8839	0.5435	0.0097	0.3071
O2MAC	0.8988	0.6785	0.9001	0.8904	0.6983	0.8789	0.4087	0.0042	0.3556
LMGEC	0.9298	0.7508	0.9304	0.9285	0.7740	0.9237	0.5846	0.0583	0.4246

Amazon Photos dataset has different attributed matrixes but shares the same adjacency matrix. Furthermore, the attributed matrix and adjacency matrix in WIKI-pro dataset for each view are different from the others. The chosen methods can deal with various cases of multi-view graph datasets except for O2MAC, which is designed for the multi-view graph containing the same attribute matrix and different adjacency matrixes. Thus, we only evaluate the other three methods on the Amazon Photos and WIKI-pro datasets.

Up to now, in multi-view attributed graph clustering, works belonging to the fusion on adjacency matrix methods are significantly more than those belonging to the fusion on embeddings methods, which may be caused by the fact that, compared with fusion on adjacency matrix methods, the fusion on embeddings methods have more parameters and longer training time. Thus, all of the selected methods are fusion on adjacency matrix methods except O2MAC, which is a fusion on embeddings method.

To demonstrate the performances of different methods more plainly, we use t-SNE to visualize the optimized node representation, as shown in Figures 12–15. For MAGC and MvAGC, we conduct SVD on the affinity matrix to obtain node representations. We can clearly see that in the ACM dataset, the clustering results of LEMGC are more clearly outlined and the categories are more differentiated than the other methods. Additionally, in the DBLP dataset, the clustering results of MAGC are more compact than other methods, with clear categories and significant category differences. Moreover, among these four methods, it is evident that the clustering results for the Photos and WIKI-pro datasets are notably inferior to those for ACM and DBLP, which can clearly distinguish different categories.

According to Table 16, we can observe that LMGEC performs well in the ACM and DBLP datasets, which demonstrates the effectiveness of the proposed linear graph filter based on one-hop neighborhood propagation. In the IMDB dataset, MAGC performs better than other methods. This method utilizes the self-expression property and the k-order graph convolution proposed by AGC. MvAGC is considered an improved version of MAGC, which greatly reduces time consumption by choosing anchor points. O2MAC implements a series of encoders and a decoder to extract consensus information that introduces a large number of parameters but is suitable for learning more complex models.

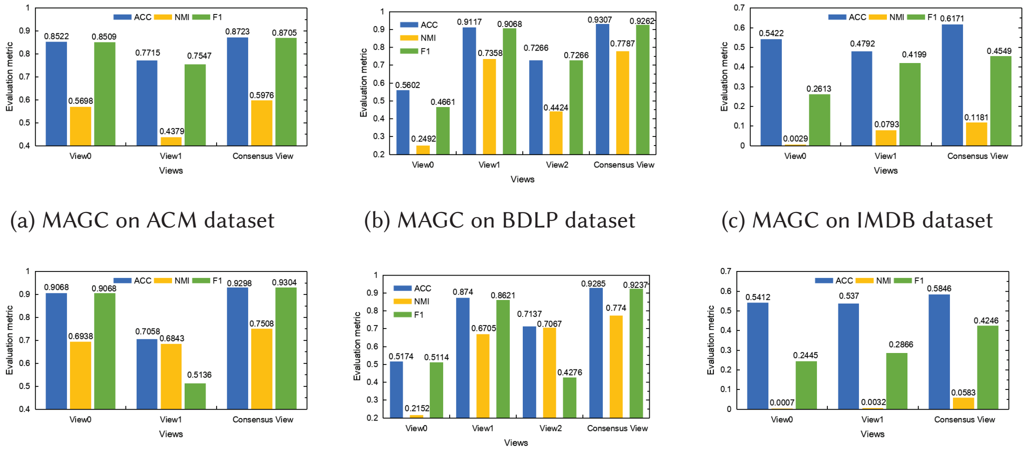
In the Amazon Photos and WIKI-pro datasets, according to Table 17, MAGC performs better than other methods, which can be observed that LMGEC is not good at dealing with the case with multiple attribute matrices.

7.3 Comparisons between Single-view Methods and Multi-view Methods

According to Tables 16 and 17, we find that LMGEC and MAGC perform well in the selected multi-view graph datasets. To further explore the relationships between single view and multiple views, we present the performances of the above two methods in each view shown in Figure 11. In Figure 11, we show the clustering results of each view and consensus view for comparison. We can observe that in existing methods, consensus views always perform better than individual views. In

Table 17. The Results of Multi-view Methods

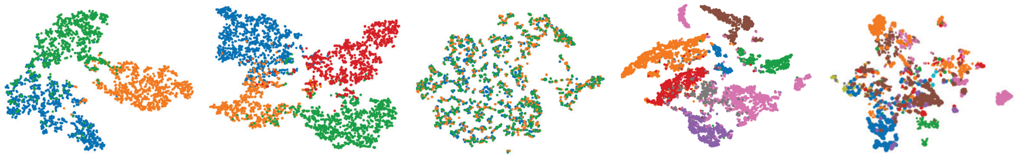
Dataset	Amazon Photos				WIKI-pro			
	ACC	NMI	F1	ARI	ACC	NMI	F1	ARI
MAGC	0.7841	0.7226	0.7291	0.6075	0.5625	0.4654	0.3851	0.3087
MvAGC	0.6630	0.5831	0.6365	0.4838	0.5018	0.4815	0.4220	0.2781
LMGEC	0.7093	0.6484	0.6087	0.5106	0.3587	0.1914	0.3779	0.2254



(a) MAGC on ACM dataset
(b) MAGC on BDLP dataset
(c) MAGC on IMDB dataset

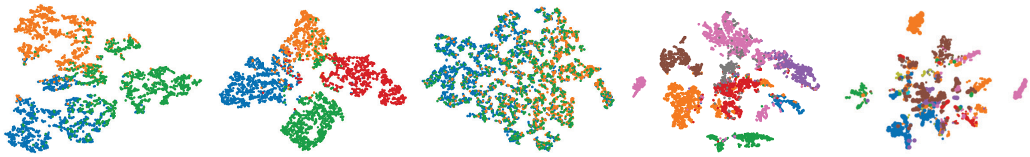
(d) LMEGC on ACM dataset
(e) LMEGC on DBLP dataset
(f) LMEGC on IMDB dataset

Fig. 11. Results with different views.



(a) ACM
(b) BDLP
(c) IMDB
(d) Photos
(e) WIKI-pro

Fig. 12. The node representations of the LMGEC method using t-SNE for two-dimensional projection and then colored based on real labels.



(a) ACM
(b) DBLP
(c) IMDB
(d) Photos
(e) WIKI-pro

Fig. 13. The node representations of the MAGC method using t-SNE for two-dimensional projection and then colored based on real labels.

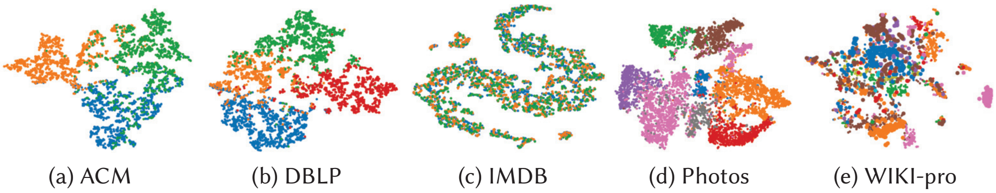


Fig. 14. The node representations of the MvGAC method using t-SNE for two-dimensional projection and then colored based on real labels.

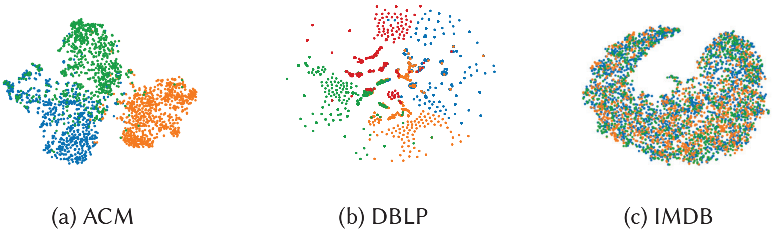


Fig. 15. The node representations of the O2MAC method using t-SNE for two-dimensional projection and then colored based on real labels.

summary, the performance of the single-view methods is closely bound up with the selected view, however, a good multi-view method can avoid the adverse information in the view and use the complementary information between the views to produce a multiplying effect on cooperation and achieve more excellent performance than the single view.

From Tables 15 and 16, we discover that there exist some single-view methods that perform better than multi-view methods in each view. In the ACM dataset, from the perspective of each view, the performances of EFR-DGC are better than MAGC in each view. In the IMDB dataset, the performances of FGC are better than LMEGC in each view. Thus, it is obvious that the integration of advanced single-view methods with the idea of multi-view may bring a leap forward in the growth of multi-view clustering on attributed graphs.

7.4 Efficiency Comparison

In this subsection, we evaluate these methods in terms of efficiency. Tables 18 and 19 show the running times of the single-view method and the multi-view method on the single-view and multi-view datasets, respectively.

Table 18 illustrates that SAGSC dedicates less time to clustering, as it employs the NJW spectral clustering approach [60] to accelerate the process. Furthermore, it can be observed that AGC-DRR requires a greater amount of time due to its utilization of graph contrastive learning, which generates two distinct views of the data. Additionally, the model is more intricate and comprises a considerable number of parameters. Moreover, we can find that, faced with the PubMed dataset, all methods need more time, for it contains tens of thousands of nodes, much larger than other datasets. Moreover, DAEGC is unable to allocate sufficient memory when confronted with the PubMed dataset, which suggests that it may require an excessive amount of space when clustering. Consequently, it is not a viable option for processing large graphs.

As illustrated in Table 19, LMGEC and MvAGC exhibit superior performance in terms of speed compared to other methods. LMGEC employs linear graph autoencoder to accelerate clustering, while MvAGC proposes the use of anchors instead of all nodes, which can notably increase the

Table 18. Running Time of Single-view Methods

Method	Citeseer	Cora	Pubmed	Wiki
AGC	83.10s	13.89s	137.62s	17.45s
DAEGC	6.84s	7.66s	out of memory	6.59s
FGC	152.20s	18.08s	335.07s	19.11s
AGC-DRR	188.25s	122.16s	4623.75s	124.72s
DFCN	31.71s	31.86s	121.71s	56.84s
SAGSC	4.42s	1.94s	4.33s	3.96s
SDCN	30.90s	21.85s	390.30s	23.98s
AGCN	35.82s	25.31s	461.66s	27.52s
EFR-DGC	96.88s	22.26s	946.15s	29.52s

Table 19. Running Time of Multi-view Methods

Method	ACM	DBLP	IMDB	Amazon Photos	WIKI-pro
O2MAC	328.64s	1116.67s	1111.19s	\	\
MAGC	83.87s	159.82s	235.48s	92.84s	103.48s
MvAGC	5.68s	9.70s	8.88s	6.80s	16.23s
LMGEC	2.86s	2.58s	2.72s	5.04s	7.24s

running time. Additionally, O2MAC requires a longer clustering time due to its design of a deep learning model with a greater number of parameters, which limits its effectiveness in processing larger graphs.

7.5 Summary

In the experimental section, we evaluate advanced single-view and multi-view methods on four single-view datasets and five multi-view datasets in Sections 7.1.1 and 7.2, respectively. Subsequently, as we examine the results of single-view methods across each view of the multi-view datasets in Section 7.1.2, we discover the clustering results vary in different views. Thus, it is logical and essential to allocate different levels of attention to different views. Moreover, we assess the performance of multi-view methods under individual views and a consensus view in Section 7.3. We find that good multi-view methods can harness complementary information between views to achieve significantly enhanced collaborative effects, surpassing the performance of single-view methods. Finally, we evaluate these methods from efficiency comparison: We find that SAGSC and LMGEC dedicate less time to clustering and may be suitable for large-scale scenarios, while DAEGC needs to allocate more memory, which is not available for large-scale scenarios.

8 Conclusion

In this section, we summarize and emphasize the key contributions and findings of this survey. Then, we suggest promising research directions related to attributed graph clustering in the future.

8.1 Summary of Key Contributions and Findings

This article conducts a comprehensive and up-to-date review of attributed graph clustering approaches. In this article, we provide a systematic taxonomy for advanced methods and summarize the methods of each class into a unified framework. Moreover, we perform a thorough experimental analysis of these advanced methods to explore relationships among single-view methods and multi-view methods. Below are our key contributions and findings:

Novel classification standard. In this article, we classify attributed graph clustering methods into three categories shown in Section 3 based on fusion results: fusion on adjacency matrix methods, fusion on embedding methods, and model-based methods. Fusion on adjacency matrix methods focus on storing attribute information within the graph structure, while fusion on embedding methods leverage GNNs to integrate information from neighboring nodes, storing topological information in node embeddings. Additionally, model-based methods can store structural and attribute information in distinct concepts. This classification approach is also applied to multi-view attributed graph clustering, as shown in Section 4.

Relationships between single-view and multi-view methods. Through combining information from different views, a good multi-view method can perform better than only using the individual view. The experiments in Section 7.3 present clustering results for both individual and consensus views in multi-view datasets. We find that the consensus view consistently outperforms individual views among existing methods. In summary, the performance of the individual view heavily relies on the chosen view, while an excellent multi-view method that combines information from different views can effectively leverage complementary information across views, leading to significant improvements to single-view outcomes. Currently, multi-view attributed graph clustering methods remain in a nascent stage of development. However, considering single-view techniques with multi-view concepts can significantly contribute to multi-view clustering on attributed graphs.

Different attentions for different views. Multi-view methods need to assign different attention to different views. The experimental results in Section 7.1.2 show that clustering outcomes vary across different views in multi-view datasets. Thus, it is both logical and necessary to give varying attention to different views in a specific task. Furthermore, the attention of each view is not fixed; it depends on the method's ability to extract useful information from each view.

8.2 Recommendations for Future Research

Although this survey conducts a comprehensive and up-to-date review of attributed graph clustering approaches, it mainly focuses on attributed graphs with node attributes, paying less attention to the cases with edge attributes, which is expected to be further investigated in future research. This subsection aims to shed light on future research directions to motivate readers and researchers in this field.

8.2.1 New Problem Setting. **Large-scale problem.** With the development of information mining technology, networks in real life have become more complex and large-scale. However, existing methods are not suitable for large-scale attributed graphs, since more running time and memory space are required by existing methods as the large scale increases. Therefore, there is an urgent need to provide a method that can reduce runtime and time complexity while improving the accuracy of large-scale graph clustering.

Multi-view problem. With the rapid expansion of collecting and mining information fields, data sources have become diversified, where different sources can provide different views, thus, multi-view data can contribute to information complementarity and improve the accuracy of clustering. However, existing single-view methods for clustering attributed graphs can not be implemented directly on multi-view attributed graphs. Hence, it is critical for us to design unique methods for multi-view graphs, which can derive from single-view methods.

Incomplete information problem. In real scenarios, not all relationships are available, which leads to the existence of incomplete information in attributed graphs. For example, due to the high cost of monitoring all protein interactions, **protein-protein interaction (PPI)** networks are often incomplete. In addition, for multi-view attributed graphs, missing and incomplete information often occurs. In fact, dealing with missing views is of great significance to us, as it can not only solve scenarios with incomplete information but also improve clustering performance.

8.2.2 High Efficiency. **Model acceleration.** With the increase in network scale, the number of views, and the number of attributes, numerous parameters will be contained in the fusion on embedding methods, which has significant impacts on runtime, footprint, and efficiency. Thus, some acceleration algorithms and related lightweight neural networks are needed to improve the efficiency of models and alleviate the construction consumption of models.

Graph Reduction. A large number of nodes and edges, as well as high dimensional attributes, can affect the running time of algorithms. Therefore, we can exploit graph pruning, subgraph sampling, and other techniques to reduce graphs and improve the efficiency of existing algorithms.

8.2.3 High Quality. **High-order adjacency.** Due to the insufficient information with only direct adjacency information, higher-order adjacency is gradually employed in recent research. There are some problems when adopting higher-order relationships to extract more topology information. Due to the fact that there are many high-order neighbors for one node, it may combine some unnecessary features and information, resulting in the probability of changing the relationships of nodes. Thus, there exists an urgent need for how to construct high-order adjacency.

The specific GNNs. For multi-view graph clustering, most deep models directly use GNN that is originally designed for the single view. Thus, it is an important issue how to design an appropriate model for multi-view tasks. Recently, some articles have adopted methods that assign an encoder and a decoder to each view or use multiple encoders but only one decoder for reconstruction. However, these encoders are not in intimate connections and with a large number of parameters, so there is an urgent need for a more efficient multi-view GNN model.

Information fusion. For attributed graphs, the fusion mechanism of attribute information and structure information has a great influence on clustering, which is always emphasized by researchers. Furthermore, for multi-view attributed graphs, the fusion of complementary and consensus information is a new challenge. Current methods mainly focus on the consensus information among different views. Existing methods extract a consensus graph from different views and then perform clustering on the consensus graph. However, there is not only consensus information but also unique information in each view. Existing methods overlook the unique information in each view, resulting in insufficient utilization of information from multiple views and reducing clustering performance.

Acknowledgments

We thank the reviewers for their constructive comments.

References

[1] Esmaeil Alinezhad, Babak Teimourpour, Mohammad Mehdi Sepehri, and Mehrdad Kargari. 2020. Community detection in attributed networks considering both structural and attribute similarities: Two mathematical programming approaches. *Neural Comput. Appl.* 32, 8 (2020), 3203–3220.

[2] Kamal Berahmand, Sogol Haghani, Mehrdad Rostami, and Yuefeng Li. 2022. A new attributed graph clustering by using label propagation in complex networks. *J. King Saud Univ. Comput. Inf. Sci.* 34, 5 (2022), 1869–1883.

[3] Kamal Berahmand, Elahe Nasiri, Rojiar Pir Mohammadiani, and Yuefeng Li. 2021. Spectral clustering on protein-protein interaction networks via constructing affinity matrix using attributed graph embedding. *Comput. Biol. Med.* 138 (2021), 104933.

[4] Lamia Berkani, Lylia Betit, and Louiza Belarif. 2021. BSO-MV: An optimized multiview clustering approach for items recommendation in social networks. *J. Univ. Comput. Sci.* 27, 7 (2021), 667–692.

[5] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. 2020. Structural deep clustering network. In *Proceedings of the Web Conference (WWW'20)*.

[6] Cécile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenková. 2015. Clustering attributed graphs: Models, measures and methods. *Netw. Sci.* 3, 3 (2015), 408–444.

[7] Zhan Bu, Huijia Li, Jie Cao, Zhen Wang, and Guangliang Gao. 2019. Dynamic cluster formation game for attributed graph clustering. *IEEE Trans. Cybern.* 49, 1 (2019), 328–341.

[8] Xin-Wei Cai, Xiangyu Ke, Kai Wang, Lu Chen, Tianming Zhang, Qing Liu, and Yunjun Gao. 2023. Efficient temporal butterfly counting and enumeration on temporal bipartite graphs. *Proc. VLDB Endow.* 17, 4 (2023), 657–670.

[9] Huidi Chen, Yun Xiong, Changdong Wang, Yangyong Zhu, and Wei Wang. 2020. SpEC: Sparse embedding-based community detection in attributed graphs. In *Proceedings of the 25th International Conference on Database Systems for Advanced Applications (DASFAA'20)*. Springer, 53–69.

[10] Haoran Chen, Zhongjing Yu, Qinli Yang, and Junming Shao. 2020. Attributed graph clustering with subspace stochastic block model. *Inf. Sci.* 535 (2020), 130–141.

[11] Xihui Chen, Sjouke Mauw, and Yunior Ramírez-Cruz. 2020. Publishing community-preserving attributed social graphs with a differential privacy guarantee. *Proc. Priv. Enhanc. Technol.* 2020, 4 (2020), 131–152.

[12] Petr Chunaev. 2020. Community detection in node-attributed social networks: A survey. *Comput. Sci. Rev.* 37 (2020), 100286.

[13] David Combe, Christine Largeron, Elöd Egyed-Zsigmond, and Mathias Géry. 2012. Combining relations and text in scientific network clustering. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM'12)*. IEEE Computer Society, 1248–1253.

[14] Pasqua D'Ambra, Panayot S. Vassilevski, and Luisa Cutillo. 2023. Extending bootstrap AMG for clustering of attributed graphs. *Appl. Math. Comput.* 447 (2023), 127904.

[15] Hasan Fehmi Danaci, Rengül Çetin-Atalay, and Volkan Atalay. 2018. EClerize: A customized force-directed graph drawing algorithm for biological graphs with EC attributes. *J. Bioinform. Comput. Biol.* 16, 4 (2018), 1850007:1–1850007:12.

[16] Wajdi Dhifli, Nour El Islem Karabadji, and Mohamed Elati. 2020. Evolutionary mining of skyline clusters of attributed graph data. *Inf. Sci.* 509 (2020), 501–514.

[17] Yuntao Du, Xinjun Zhu, Lu Chen, Ziquan Fang, and Yunjun Gao. 2023. MetaKG: Meta-learning on knowledge graph for cold-start recommendation. *IEEE Trans. Knowl. Data Eng.* 35, 10 (2023), 9850–9863.

[18] Srinivas Eswar, Ramakrishnan Kannan, Richard W. Vuduc, and Haesun Park. 2021. ORCA: Outlier detection and robust clustering for attributed graphs. *J. Glob. Optim.* 81, 4 (2021), 967–989.

[19] Shaohua Fan, Xiao Wang, Chuan Shi, Emiao Lu, Ken Lin, and Bai Wang. 2020. One2Multi graph autoencoder for multi-view graph clustering. In *Proceedings of the Web Conference (WWW'20)*. ACM/IW3C2, 3070–3076.

[20] Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. 2023. Scalable attributed-graph subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[21] Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. 2023. Simultaneous linear multi-view attributed graph representation learning and clustering. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM'23)*. ACM, 303–311.

[22] Linton C. Freeman. 2000. Visualizing social networks. *J. Soc. Struct.* 1 (2000), 4.

[23] Hongchang Gao and Heng Huang. 2018. Deep attributed network embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. International Joint Conferences on Artificial Intelligence Organization, 3364–3370.

[24] Hongchang Gao and Heng Huang. 2018. Deep attributed network embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. ijcai.org, 3364–3370.

[25] Yunjun Gao, Tianming Zhang, Linshan Qiu, Qingyuan Linghu, and Gang Chen. 2021. Time-respecting flow graph pattern matching on temporal graphs. *IEEE Trans. Knowl. Data Eng.* 33, 10 (2021), 3453–3467.

[26] Lei Gong, Sihang Zhou, Wenxuan Tu, and Xinwang Liu. 2022. Attributed graph clustering with dual redundancy reduction. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI'22)*. ijcai.org, 3015–3021.

[27] Steven Greenberg. 2009. How citation distortions create unfounded authority: Analysis of a citation network. *BMJ* 339, 7714 (2009), b2680.

[28] Jie Hao and William Zhu. 2023. Deep graph clustering with enhanced feature representations for community detection. *Appl. Intell.* 53, 2 (2023), 1336–1349.

[29] Chaobo He, Shuangyin Liu, Lei Zhang, and Jianhua Zheng. 2019. A fuzzy clustering based method for attributed graph partitioning. *J. Ambient Intell. Humaniz. Comput.* 10, 9 (2019), 3399–3407.

[30] Lun Hu, Shicheng Yang, Xin Luo, and MengChu Zhou. 2022. An algorithm of inductively identifying clusters from attributed graphs. *IEEE Trans. Big Data* 8, 2 (2022), 523–534.

[31] Bingyang Huang, Chaokun Wang, and Binbin Wang. 2019. NMLPA: Uncovering overlapping communities in attributed networks via a multi-label propagation approach. *Sensors* 19 (2019), 260.

[32] Zhichao Huang, Yunming Ye, Xutao Li, Feng Liu, and Huajie Chen. 2017. Joint weighted nonnegative matrix factorization for mining attributed graphs. In *Proceedings of the 21st Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'17).* 368–380.

[33] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *J. Classif* 2, 1 (1985), 193–218.

[34] Hiroyoshi Ito, Takahiro Komamizu, Toshiyuki Amagasa, and Hiroyuki Kitagawa. 2019. Detecting communities and correlated attribute clusters on multi-attributed graphs. *IEICE Trans. Inf. Syst.* 102-D, 4 (2019), 810–820.

[35] Chaojie Ji, Hongwei Chen, Ruxin Wang, Yunpeng Cai, and Hongyan Wu. 2022. Smoothness sensor: Adaptive smoothness-transition graph convolutions for attributed graph clustering. *IEEE Trans. Cybern.* 52, 12 (2022), 12771–12784.

[36] Junzhong Ji, Ye Liang, and Minglong Lei. 2021. Deep attributed graph clustering with self-separation regularization and parameter-free cluster estimation. *Neural Netw.* 142 (2021), 522–533.

[37] Barakeel Fanseu Kamhoua, Lin Zhang, Kaili Ma, James Cheng, Bo Li, and Bo Han. 2023. GRACE: A general graph convolution framework for attributed graph clustering. *ACM Trans. Knowl. Discov. Data* 17, 3 (2023), 31:1–31:31.

[38] Zhao Kang, Zhanyu Liu, Shirui Pan, and Ling Tian. 2022. Fine-grained attributed graph clustering. In *Proceedings of the SIAM International Conference on Data Mining (SDM'22).* SIAM, 370–378.

[39] Ali Golzadeh Kermani, Ali Kamandi, and Ali Moeini. 2022. Integrating graph structure information and node attributes to predict protein-protein interactions. *J. Comput. Sci.* 64 (2022), 101837.

[40] Abeer Khan, Lukasz Golab, Mehdi Kargar, Jaroslaw Szlichta, and Morteza Zihayat. 2020. Compact group discovery in attributed graphs and social networks. *Inf. Process. Manag.* 57, 2 (2020), 102054.

[41] Nasrullah Khan, Zongmin Ma, Aman Ullah, and Kemal Polat. 2022. Similarity attributed knowledge graph embedding enhancement for item recommendation. *Inf. Sci.* 613 (2022), 69–95.

[42] Mayank Kumar and Ruchir Gupta. 2021. Overlapping attributed graph clustering using mixed strategy games. *Appl. Intell.* 51, 8 (2021), 5299–5313.

[43] Tuan M. V. Le and Hady Wirawan Lauw. 2014. Probabilistic latent document network embedding. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'14).* IEEE Computer Society, 270–279.

[44] T. V. Le and H. W. Lauw. 2014. Probabilistic latent document network embedding. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'14).* IEEE Computer Society, 270–279.

[45] Jundong Li, Xia Hu, Jiliang Tang, and Huan Liu. 2015. Unsupervised streaming feature selection in social media. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15).* ACM, 1041–1050.

[46] Zhen Li, Zhisong Pan, Guyu Hu, Guopeng Li, and Xingyu Zhou. 2017. Detecting semantic communities in social networks. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* 100-A, 11 (2017), 2507–2512.

[47] Huifa Liao, Jie Hu, Tianrui Li, Shengdong Du, and Bo Peng. 2022. Deep linear graph attention model for attributed graph clustering. *Knowl. Based Syst.* 246 (2022), 108665.

[48] Jia-Qi Lin, Mansheng Chen, Xi-Ran Zhu, Chang-Dong Wang, and Haizhang Zhang. 2022. Dual information enhanced multi-view attributed graph clustering. *CoRR* abs/2211.14987 (2022).

[49] Qiang Lin, Yong Lin, Qiang Yu, and Xiaoke Ma. 2020. Clustering of cancer attributed networks via integration of graph embedding and matrix factorization. *IEEE Access* 8 (2020), 197463–197472.

[50] Zhiping Lin and Zhao Kang. 2021. Graph filter-based multi-view attributed graph clustering. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21).* ijcai.org, 2723–2729.

[51] Zhiping Lin, Zhao Kang, Lizong Zhang, and Ling Tian. 2023. Multi-view attributed graph clustering. *IEEE Trans. Knowl. Data Eng.* 35, 2 (2023), 1872–1880.

[52] Liang Liu, Zhao Kang, Jiajia Ruan, and Xixu He. 2022. Multilayer graph contrastive clustering network. *Inf. Sci.* 613 (2022), 256–267.

[53] Dan-Dan Lu, Ji Qi, Jie Yan, and Zhong-Yuan Zhang. 2022. Community detection combining topology and attribute information. *Knowl. Inf. Syst.* 64, 2 (2022), 537–558.

[54] Hua Ma, Yuqi Tang, Xuxiang Zhang, Haibin Zhu, Peiji Huang, and Hongyu Zhang. 2023. Learning resource recommendation via knowledge graphs and learning style clustering. *J. Intell. Fuzzy Syst.* 44, 5 (2023), 8053–8069.

[55] Seiji Maekawa, Koh Takeuchi, and Makoto Onizuka. 2020. New attributed graph clustering by bridging attribute and topology spaces. *J. Inf. Process.* 28 (2020), 427–435.

[56] Deepanshu Malhotra and Anuradha Chug. 2021. A modified label propagation algorithm for community detection in attributed networks. *Int. J. Inf. Manag. Data Insights* 1, 2 (2021), 100030.

[57] Debasis Mohapatra and Manas Ranjan Patra. 2019. Anonymization of attributed social graph using anatomy based clustering. *Multim. Tools Appl.* 78, 18 (2019), 25455–25486.

[58] Elahe Nasiri, Kamal Berahmand, Mehrdad Rostami, and Mohammad Dabiri. 2021. A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Comput. Biol. Med.* 137 (2021), 104772.

[59] Jennifer Neville, Micah Adler, and David Jensen. 2003. Clustering relational data using attribute and link information. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence: Text Mining and Link Analysis Workshop.*

[60] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *Proceedings of the Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01).* MIT Press, 849–856.

[61] Hung T. Nguyen and Thang N. Dinh. 2015. Unveiling the structure of multi-attributed networks via joint non-negative matrix factorization. In *Proceedings of the IEEE Military Communications Conference (MILCOM'15).* IEEE, 1379–1384.

[62] Charles Otto, Dayong Wang, and Anil K. Jain. 2018. Clustering millions of faces by identity. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 2 (2018), 289–303.

[63] Guangjin Ou, Guoxian Yu, Carlotta Domeniconi, Xuequan Lu, and Xiangliang Zhang. 2020. Multi-label zero-shot learning with graph convolutional networks. *Neural Netw.* 132 (2020), 333–341.

[64] Erlin Pan and Zhao Kang. 2021. Multi-view contrastive graph clustering. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS'21).*

[65] Yulong Pei, Nilanjan Chakraborty, and Katia P. Sycara. 2015. Nonnegative matrix tri-factorization with graph regularization for community detection in social networks. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15).* AAAI Press, 2083–2089.

[66] Zhihao Peng, Hui Liu, Yuheng Jia, and Junhui Hou. 2021. Attention-driven graph clustering network. In *Proceedings of the 29th ACM International Conference on Multimedia (MM'21).*

[67] Clara Pizzuti and Annalisa Socievole. 2020. Multiobjective optimization and local merge for clustering attributed graphs. *IEEE Trans. Cybern.* 50, 12 (2020), 4997–5009.

[68] Meng Qin, Di Jin, Kai Lei, Bogdan Gabrys, and Katarzyna Musial-Gabrys. 2018. Adaptive community detection incorporating topology and content in social networks. *Knowl. Based Syst.* 161 (2018), 342–356.

[69] Mehrdad Rostami, Mourad Oussalah, and Vahid Farrahi. 2022. A novel time-aware food recommender-system based on deep learning and graph clustering. *IEEE Access* 10 (2022), 52508–52524.

[70] Firas Saidi, Zouheir Trabelsi, and Henda Ben Ghézala. 2018. An evidential clustering based framework for cyber terrorist cells topology identification. In *Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE'18).* IEEE.

[71] Asif Salim, S. S. Shiju, and S. Sumitra. 2023. Neighborhood preserving kernels for attributed graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* 45 (2023), 828–840.

[72] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *CoRR* abs/1811.05868 (2018).

[73] Karsten Steinhaeuser and Nitesh V. Chawla. 2008. Community detection in a large real-world social network. In *Social Computing, Behavioral Modeling, and Prediction*, Huan Liu, John J. Salerno, and Michael J. Young (Eds.). Springer US, Boston, MA, 168–175.

[74] Karsten Steinhaeuser and Nitesh V. Chawla. 2010. Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters* 31, 5 (2010), 413–421.

[75] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* 3 (2002), 583–617.

[76] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cécile Paris, Surya Nepal, Di Jin, Quan Z. Sheng, and Philip S. Yu. 2021. A comprehensive survey on community detection with deep learning. *CoRR* abs/2105.12584, (2021).

[77] Sharmila Subudhi and Suvasini Panigrahi. 2020. Two-stage automobile insurance fraud detection by using optimized fuzzy C-Means clustering and supervised learning. *Int. J. Inf. Secur. Priv.* 14, 3 (2020), 18–37.

[78] A. Dang and E. Viennet. 2012. Community detection based on structural and attribute similarities. In *Proceedings of the 6th International Conference on Digital Society (ICDS'12).*

[79] Fengqin Tang, Chunning Wang, Jinxia Su, and Yuanyuan Wang. 2020. Spectral clustering-based community detection using graph distance and node attributes. *Comput. Stat.* 35, 1 (2020), 69–94.

[80] Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, En Zhu, and Jieren Cheng. 2021. Deep fusion clustering network. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI'21), 33rd Conference on Innovative Applications of Artificial Intelligence (IAAI'21), 11th Symposium on Educational Advances in Artificial Intelligence (EAAI'21).* AAAI Press, 9978–9987.

[81] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18).* OpenReview.net.

[82] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Attributed graph clustering: A deep attentional embedding approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19).* ijcai.org, 3670–3676.

[83] Chun Wang, Shirui Pan, Celina Ping Yu, Ruiqi Hu, Guodong Long, and Chengqi Zhang. 2022. Deep neighbor-aware embedding for node clustering in attributed graphs. *Pattern Recog.* 122 (2022), 108230.

[84] Jialin Wang, Xiaoru Qu, Jinze Bai, Zhao Li, Ji Zhang, and Jun Gao. 2023. SAGES: Scalable attributed graph embedding with sampling for unsupervised learning. *IEEE Trans. Knowl. Data Eng.* 35, 5 (2023), 5216–5229.

[85] Tong Wang, Junhua Wu, Zhenquan Zhang, Wen Zhou, Guang Chen, and Shasha Liu. 2021. Multi-scale graph attention subspace clustering network. *Neurocomputing* 459 (2021), 302–314.

[86] Tong Wang, Guanyu Yang, Qijia He, Zhenquan Zhang, and Junhua Wu. 2022. NCAGC: A neighborhood contrast framework for attributed graph clustering. arXiv:2206.07897

[87] Xiao Wang, Di Jin, Xiaochun Cao, Liang Yang, and Weixiong Zhang. 2016. Semantic community identification in large attribute networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence.* AAAI Press, 265–271.

[88] Xufei Wang, Lei Tang, Huiji Gao, and Huan Liu. 2010. Discovering overlapping groups in social media. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10).* IEEE Computer Society, 569–578.

[89] Yuxiang Wang, Cangqi Zhou, Jing Zhang, and Qianmu Li. 2022. Attributed graph clustering with double contrastive projector. In *Proceedings of the International Joint Conference on Neural Networks.* IEEE, 1–8.

[90] Daqing Wu, Xiangyang Guo, Xiao Luo, Ziyue Qiao, and Jinwen Ma. 2022. Adaptive harmony learning and optimization for attributed graph clustering. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'22).* IEEE, 1–8.

[91] Wei Xia, Quanxue Gao, Ming Yang, and Xinbo Gao. 2021. Self-supervised contrastive attributed graph clustering. *CoRR* abs/2110.08264 (2021).

[92] Yu Xie, Yuanqiao Zhang, Maoguo Gong, Zedong Tang, and Chao Han. 2020. MGAT: Multi-view graph attention networks. *Neural Netw.* 132 (2020), 180–189.

[93] Xiaoqiang Yan, Xiangyu Yu, Shizhe Hu, and Yangdong Ye. 2023. Mutual boost network for attributed graph clustering. *Expert Syst. Appl.* 229, Part A (2023), 120479.

[94] Fang Yang, Kunjie Fan, Dandan Song, and Huakang Lin. 2020. Graph-based prediction of Protein-protein interactions with attributed signed graph embedding. *BMC Bioinform.* 21, 1 (2020), 323.

[95] Renchi Yang, Jieming Shi, Yin Yang, Keke Huang, Shiqi Zhang, and Xiaokui Xiao. 2021. Effective and scalable clustering on massive attributed graphs. In *Proceedings of the Web Conference (WWW'21).* ACM/IW3C2, 3675–3687.

[96] Zelin Zang, Siyuan Li, Di Wu, Jianzhu Guo, Yongjie Xu, and Stan Z. Li. 2022. Deep manifold embedding of attributed graphs. *Neurocomputing* 514 (2022), 83–93.

[97] Tianming Zhang, Yunjun Gao, Lu Chen, Wei Guo, Shiliang Pu, Baihua Zheng, and Christian S. Jensen. 2019. Efficient distributed reachability querying of massive temporal graphs. *VLDB J.* 28, 6 (2019), 871–896.

[98] Xiaotong Zhang, Han Liu, Qimai Li, and Xiao-Ming Wu. 2019. Attributed graph clustering via adaptive graph convolution. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19).* ijcai.org, 4327–4333.

[99] Xiaotong Zhang, Han Liu, Xiao-Ming Wu, Xianchao Zhang, and Xinyue Liu. 2021. Spectral embedding network for attributed graph clustering. *Neural Netw.* 142 (2021), 388–396.

[100] Xiaotong Zhang, Han Liu, Xianchao Zhang, and Xinyue Liu. 2022. Attributed graph clustering with multi-task embedding learning. *Neural Netw.* 152 (2022), 224–233.

[101] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. 2021. Graph debiased contrastive learning with joint representation clustering. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21).* ijcai.org, 3434–3440.

Received 20 September 2023; revised 18 December 2024; accepted 6 January 2025