Journal of Mathematical Modeling
Vol. -, No. -, 2026, pp. -. Research Article

JMM

Jointly subspace clustering and manifold learning for attributed graphs

Elham Khodaparast, Mina Jamshidi*

Department of Applied Mathematics, Graduate University of Advanced Technology

Email(s): m.jamshidi@kgut.ac.ir, mjamshidi78@yahoo.com, el-kh-85@yahoo.com

Abstract. Attributed graph clustering, or community detection, groups data with predefined graph similarity matrix based on both topological structure of the graph and node features. The features of nodes and their connections may not be fully consistent, leading to a potential mismatch between node attributes and graph structure in attributed graph clustering. Therefore, one of the main goals of attributed graph clustering is to learn similarity and feature representations that are as compatible as possible. Continuing this goal, in this paper we provide a method in which a new similarity matrix is constructed as close as possible to the predefined similarity matrix. Moreover, it simultaneously captures the subspace structure of data points and preserves the manifold structure induced by the feature matrix which causes more smoothness in the new graph structure. Results on real-world datasets demonstrate the effectiveness of our method in community detection.

Keywords: Attributed graph, geometrical structure, subspace learning.
AMS Subject Classification 2010: 68T05, 68T09

1 Introduction

Clustering is an approach of unsupervised data mining, which has widespread application in different domains [12, 16, 44]. While traditional partitioning methods like K-means and spectral clustering remain foundational, some research has aimed to improve their efficiency and clustering quality. This includes developing acceleration techniques for K-means [6], refining similarity measures through graph structures or local density [40, 50]. Some other researches utilize encoding methods to capture intrinsic structural patterns [10, 41]. The landscape of clustering has further expanded into specialized paradigms, such as fuzzy, neighborhood-based, multi-view, and graph-based clustering [1, 36, 39]. In particular, attributed graph clustering has emerged as a highly promising approach, as it simultaneously exploits topological connectivity and node-level attributes.

*Corresponding author
Received: 25 April 2026/ Revised: 15 September 2026/ Accepted: 16 September 2026
DOI: 10.22124/jmm.2026.33369.3050

Attributed graphs are data representation that combine topological structure with node features. The topological structure, often encoded as a similarity matrix, defines connections between points (e.g., in networks), while data attributes are represented as a feature matrix and provide additional information about the nodes. Social networks are the most famous examples of such data.

In a social network, the feature matrix includes user profile information–such as age, location, and interests–whereas the adjacency matrix captures the strength of interactions among users. Integrating both structural and attribute information provides deeper insights and more reliable results. By leveraging both, one can uncover latent patterns that remain hidden when considering either matrix in isolation. Attributed graph clustering has vast applications across diverse fields, including social networks (e.g., community detection, influencer analysis), bioinformatics (e.g., protein complex detection, disease subtype classification), telecommunications (e.g., anomaly detection, network optimization), computer vision (e.g., image segmentation, object recognition), and recommender systems (e.g., personalized content delivery).

In recent years, several methods and approaches have been proposed in this field, which fall into a number of key categories [35]. Some methods are based on generative models [8,22], where probabilistic techniques are used to uncover latent structures and vertex relationships. Other methods, such as [32,49], leverage random walk matrices to update the feature matrix. These methods utilize both graph structure and vertex information. Another major group of approaches, such as [11,43], focuses on learning vertex representations that preserve both feature similarity and graph geometry.

Non-negative matrix factorization-based methods have also been introduced, in which the feature matrix is decomposed into non-negative components, helping to improve the interpretability of the results [24, 33, 38]. These approaches have several drawbacks. Some are effective at capturing underlying topological patterns; however, they struggle with high-dimensional attributes. Local neighborhoods are well considered in some methods, but global graph properties are often ignored. To overcome this drawback authors proposed spectral subspace clustering for attributed graphs [15, 36]. However, one of the main challenges in is the misalignment between the feature and similarity matrices, i.e., the frequent mismatch between the graph structure encoded in the similarity matrix and the node features. This divergence occurs when topologically close nodes have dissimilar attributes (e.g., socially connected users with different interests), or vice versa.

To address this drawback, some methods treat the attribute matrix as a graph signal and exploit the fact that a signal is smooth when nearby nodes have similar feature values; accordingly, the feature matrix is updated using the similarity matrix [26, 56]. However, these methods ignore the overall structure of the data.

In this paper, with the aim of improving the interaction between graph connectivity and node attributes during the clustering process, we propose a novel method in which the hidden geometry of the graph, global structure, and feature smoothness are considered simultaneously to reconstruct a new similarity matrix. The main contributions of this work are summarized as follows:

- We propose a novel attributed graph clustering framework that learns an adaptive similarity matrix by jointly exploiting the self-expressive property of node attributes and the graph structure information. Unlike conventional graph-based clustering methods that directly rely on a predefined graph, the proposed approach refines the graph structure through similarity matrix learning.
- We introduce a unified objective function that simultaneously incorporates self-expressive graph learning, manifold structure preservation, and closeness to the original graph topology. As a result,

the learned similarity matrix captures both latent linear relationships and local geometric information while remaining consistent with the original similarity matrix.

- Different from many existing approaches that perform graph construction and spectral clustering in separate stages, the proposed method integrates graph learning and spectral clustering into a single optimization framework. This joint learning strategy allows the similarity matrix and clustering assignments to mutually enhance each other during the process.
- Extensive comparisons on four benchmark attributed graph datasets demonstrate that the proposed method achieves competitive and often superior clustering performance compared with several recent state-of-the-art attributed graph clustering methods.

The proposed method is illustrated in Figure 1. In this paper, uppercase and lowercase letters are used

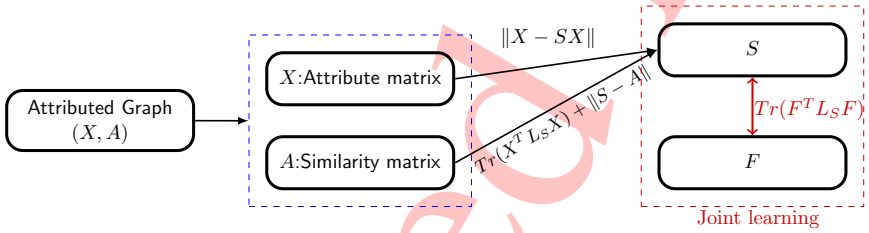


Figure 1: The illustration of our method. In our method we reconstruct a similarity matrix so that it is more compatible to the attribute matrix. This matrix contains hidden geometrical and subspace properties of the data.

to denote matrices and vectors, respectively. We consider $G = (V, E, A, X)$ as an attributed graph, where $V = \{v_1, v_2, \ldots, v_n\}$ and E denote the set of nodes and edges, respectively; $X = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^{n \times d}$ is the feature matrix, or attribute matrix, and A is the predefined similarity or adjacency matrix. We use the notation L_S for the Laplacian matrix of a similarity matrix S.

The remainder of the paper is organized as follows. In Section 2, the related work is described. Our method is proposed in Section 3. In this section, we first discuss the main approach and then solve its optimization problem and prove its convergence. Section 4 presents the evaluation of our method and the numerical results, and finally, in Section 5, we conclude the paper.

2 Related works

Attributed graph clustering is an important tool for discovering knowledge in data and has become very popular in recent years. In graph clustering, the main aim is to group nodes of a graph into clusters such that the nodes inside each cluster are highly similar to each other, while the edges between clusters indicate low similarity [25, 54]. In attributed graph clustering, both node similarity in the graph and node features are important, involving two levels: graph structural information and node feature information.

Early methods considered only graph structure information [17, 47] or only node feature information [21, 23], and did not consider them simultaneously. However, recent methods focus on both types of information, such as the co-clustering method [20]. To address the mismatch between similarity and feature matrices, several research directions have emerged:

Subspace-based methods: These methods focus on capturing the underlying linear structure of the data. For instance, [48] introduced a model in which the self-expressiveness of the data is learned using autoencoders and attention fusion modules. Similarly, [34] used subspace learning and structure fusion to capture multi-scale information and enhance attribute smoothness over the graph. Additionally, [14] suggested a method in which subspace structure, in addition to Laplacian smoothing, is employed, and [7] applied a subspace stochastic technique to the attribute matrix while utilizing a term for preserving graph structure.

Manifold-based methods: These approaches emphasize preserving the geometric manifold of the data. For example, [31] investigated a low-rank representation model in which dual manifold regularization is used to balance the attribute matrix and graph topology similarities. Furthermore, some studies focus on combining attribute similarity in subspaces with graph structure, such as the techniques proposed in [18, 19].

Graph refinement and Deep Learning methods: Other research focuses on updating the graph structure or using neural networks. In [5, 22, 55], both graph structure and nodal features were taken into account for clustering. In [2], the authors suggested embedding nodes into a continuous vector space to update the predefined graph structure based on attribute affinities. In the realm of deep learning, [54] used an autoencoder and a marginalized graph convolutional network for compact representation learning. Other neural-network-based approaches include the variational graph autoencoder in [28], the adversarial graph variational autoencoder in [45], and the structural deep clustering network in [4].

As mentioned before, the main problem in attributed graph clustering is the mismatch between the predefined similarity matrix and the attribute matrix of the data. While the aforementioned studies have made significant attempts to address this issue, they often focus on one aspect (subspace, manifold, or refinement) at the expense of others. In this work, we propose a novel method that integrates these perspectives into a unified framework.

3 Methodology

Research on clustering attributed graphs, as one of the advanced and efficient methods in data mining, can play a significant role in improving the quality and accuracy of data analysis. One of the main goals of recent clustering approaches is about having compatible relation between similarity matrix and nodal attributes. Suppose $G = (V, E, A, X)$ be an attributed graph, in which V is the set of nodes, E is the set of edges, A is the predefined similarity matrix and $X = [x_1, x_2, ..., x_n]^T \in \mathbb{R}^{n \times d}$ is the attribute matrix, where $x_i \in \mathbb{R}^d$ denotes the i^{th} data.

In the proposed method, we would like to construct a new similarity matrix S from the initial similarity matrix A which captures underlying linear relations in the data. Moreover, the similarity matrix is learned in such a way that it is not too far from the initial similarity matrix. It is desirable that this similarity matrix has exactly k connected components, and S can be directly used for the clustering task. In the following we describe the proposed method in detail.

3.1 Proposed method

As mentioned above, we aim to construct a new similarity matrix S which contains the linear structure of data. The main intuition behind our approach is that nodes belonging to the same cluster are often characterized not only by their graph connectivity but also by their ability to linearly represent each

other in the attribute space. Therefore, instead of directly using the predefined similarity matrix A, a new similarity matrix S is learned from the data. This matrix is expected to reveal latent relationships among nodes by assigning larger similarity values to nodes that can effectively reconstruct one another. As a result, nodes with similar attributes and similar underlying structures tend to form stronger connections, while unrelated nodes receive smaller similarity weights. This learned representation provides a better graph structure for clustering than the original similarity matrix alone.

The term linear structure denotes the extent to which the similarity matrix can represent the underlying subspace distribution of the data points. The self-expressive property is motivated by the observation that data points belonging to the same cluster often lie in or near a low-dimensional subspace in the attribute space. Therefore, each data point can often be represented as a linear combination of other data points from the same subspace. In an attributed graph, this property is useful because nodes within the same cluster tend to have similar attribute characteristics. Thus, by using the self-expressive property, the similarity matrix S can be learned such that the attribute information of each node can be represented by other nodes. In the proposed method, the similarity matrix S is learned by leveraging the self-expressive property to capture the linear structure of the data. It means that we consider S as

$$\|X - SX\|_F^2, \tag{1}$$

that is similar to subspace clustering. Moreover, we have

$$x_{i,r} \simeq s_{i,1}x_{1,r} + s_{i,2}x_{2,r} + \ldots + s_{i,n}x_{n,r}.$$

One main property from this relation is that the similarity weights s_{ij} are learnt such that strong connections (large s_{ij} values) between nodes lead to more similar feature representations and smooth feature transitions across the graph representation of S.

In addition to the self-expressive property, we incorporate a manifold regularization term that preserves the intrinsic geometric structure of the data X in the similarity matrix S. The manifold assumption states that nodes connected with larger weights should have more similar feature representations. This can be enforced by minimizing the following objective:

$$\frac{1}{2}\sum_{i,j}\|x_i - x_j\|_2^2 s_{i,j} = Tr(X^T L_S X). \tag{2}$$

Minimizing the above equation would cause that if s_{ij} is large, then $\|x_i - x_j\|_2^2$ should be small and vice-versa. This formulation directly penalizes large differences between features of strongly connected nodes, thereby it enforces smoothness across the manifold structure.
Summing up the above explanation, i.e. considering the formulations in Eq. (1) and Eq. (2), we have the following optimization problem:

$$\begin{aligned} &\min_S \|X - SX\|_F^2 + \alpha Tr(X^T L_S X) \\ &\text{s.t.} \quad X \in \mathbb{R}^{n\times d},\ S \in \mathbb{R}^{n\times n},\ S \geq 0. \end{aligned} \tag{3}$$

For the aim of having a restriction on S for not being far from the properties of the original similarity matrix A, we add the term $\|S - A\|_F^2$ to problem (3), as follows:

$$\min_S \|X - SX\|_F^2 + \alpha Tr(X^T L_S X) + \|S - A\|_F^2 \tag{4}$$

$$s.t. \quad X \in \mathbb{R}^{n\times d}, \ S \in \mathbb{R}^{n\times n}, \ S \geq 0.$$

The proposed objective function consists of three complementary components. The first term, ($\|X - SX\|_F^2$), is derived from the self-expressive property and aims to learn a similarity matrix capable of capturing the latent linear relationships among nodes. The second term, ($Tr(X^T L_S X)$), preserves the local geometric structure of the attribute space by encouraging strongly connected nodes to have similar feature representations. The third term, ($\|S-A\|_F^2$), acts as a fidelity constraint that prevents the learned similarity matrix from the original graph structure encoded in A. Together, these terms allow the learned similarity matrix to simultaneously exploit latent linear dependencies, manifold information, and prior graph connectivity.

The objective function in (4) is able to learn a similarity matrix that captures both the linear and geometric structures of the data. However, the learned graph is not yet explicitly optimized for clustering. In many graph-based clustering methods, graph construction and spectral clustering are performed sequentially. Such a two-stage strategy may propagate graph construction errors to the clustering stage. To overcome this limitation, we integrate graph learning and spectral clustering into a unified framework by introducing the term ($Tr(F^T L_S F)$), where F denotes the cluster indicator matrix. According to the spectral graph theory, minimizing this term under the constraint ($F^T F = I$) encourages nodes connected with large similarity values to be assigned to the same cluster. Consequently, the similarity matrix and clustering assignments are optimized simultaneously, allowing the learned graph to become more suitable for clustering. Hence, we have the following optimization problem:

$$\begin{aligned} L = \min_{S,F} \|X - SX\|_F^2 + \alpha Tr(X^T L_S X) + \beta \|S - A\|_F^2 + \gamma Tr(F^T L_S F) \\ s.t. \quad X \in \mathbb{R}^{n\times d}, \ S \in \mathbb{R}^{n\times n}, \ S \geq 0, \ F^T F = I, \end{aligned} \tag{5}$$

where α, β and γ are tuning parameters.

3.2 Solving the optimization problem

In this section, we obtain the optimal solutions of problem (5) in an alternative process as follows:

- Fix F. To calculate $\dfrac{\partial L}{\partial S}$, we rewrite the objective function L in the following way:

$$\begin{aligned} L &= \mathrm{Tr}\left((X - SX)^T (X - SX)\right) + \alpha \left(X^T D X - X^T S X\right) \\ &\quad + \beta |S - A|_F^2 + \gamma \mathrm{Tr}\left(F^T D F - F^T S F\right) \\ &= \mathrm{Tr}\left(X^T X - X^T S X - X^T S^T X + X^T S^T S X - \alpha X^T S X\right) \\ &\quad + \alpha \,\mathrm{Tr}\left(X^T D X\right) + \beta |S - A|_F^2 + \gamma \mathrm{Tr}\left(F^T D F\right) - \gamma \mathrm{Tr}\left(F^T S F\right). \end{aligned}$$

  Thus,

$$\begin{aligned} \frac{\partial L}{\partial S} &= -2XX^T + 2SXX^T - \alpha XX^T - \gamma FF^T \\ &\quad + \alpha Q + \gamma E + 2\beta (S - A), \end{aligned} \tag{6}$$

Algorithm 1: Iterative Graph Learning Algorithm

Require: Data matrix X; tuning parameters α, β, γ; max iterations $T_{\max}$; convergence threshold ε.

Initialize $S^{(0)} = A$, and let $F^{(0)}$ be the solution of $\min_F F^T L_{S^{(0)}} F$.
for $t = 1$ to $T_{\max}$ **do**
  Update $S^{(t)}$ using the closed-form solution in Eq. (7).
  $S^{(t)} \leftarrow \max(S^{(t)}, 0)$.
  Update $F^{(t)}$ by solving the spectral problem in Eq. (8).
  if $\|S^{(t)} - S^{(t-1)}\|_F / \|S^{(t-1)}\|_F < \varepsilon$ **then**
    break
  end if
end for
Retrieve cluster assignments via k-means on $F^{(t)}$.

Ensure: Clustering result.

where $Q, E \in \mathbb{R}^{n\times n}$ that $q_{ij} = \|X_{i,:}\|_2^2$, and $e_{ij} = \|F_{i,:}\|_2^2$, for $j = 1, \ldots, n$. Now, setting Eq. (6) equal to 0, gives

$$S = [\alpha(XX^T - Q) + \gamma(FF^T - E) + 2XX^T + 2\beta(A)](2\beta I + 2XX^T)^{-1} \tag{7}$$

By the first order approximation of Taylor expansion, $(I + aX)^{-1} \simeq (I - aX)$, thus the inverse calculation of large matrices can be avoided and, instead of $(2\beta I + 2XX^T)^{-1}$, we may use $\frac{1}{2\beta}(I - \frac{1}{\beta}XX^T)$.

To enforce the nonnegativity constraint on S, we perform the projection $S^{(t)} \leftarrow \max\left(S^{(t)}, 0\right)$.

- Suppose S is fixed, to get F, we have

$$\min_{F\in\mathbb{R}^{n\times k}} Tr(F^T L_S F) \qquad \text{s.}t. \qquad F^T F = I, \tag{8}$$

which is the problem of spectral clustering, and the optimal solution of F is obtained by the k eigenvectors of L_S corresponding to its k smallest eigenvalues.

We repeat the process until a predefined maximum number of iterations, $T_{\max} = 20$, is reached. This spectral-based initialization ensures that the iterative optimization begins with a configuration that captures the inherent structural information of the graph. Algorithm 1 summarizes the optimization procedure of (5). The updates of S and F correspond to Eqs. (7) and (8), respectively. The similarity matrix S is initialized using the predefined adjacency matrix A, i.e., $S^{(0)} = A$. The matrix $F \in \mathbb{R}^{n\times k}$ is initialized via spectral embedding, where its columns are set to the k eigenvectors corresponding to the k smallest eigenvalues of A. This ensures that the optimization process begins with a representation that captures the intrinsic structural topology of the graph. Also, the parameters are selected via grid search.

3.3 Learning mechanism

The proposed approach can be interpreted as an iterative graph learning algorithm. The variables S and F are treated as learnable matrices that are estimated directly from the data X. The algorithm starts from

initial similarity matrix S and the cluster indicator matrix F. Then an alternating learning process is done and S and F are iteratively refined. This process allows the graph structure and clustering assignments to mutually improve each other. Actually, in the first stage, the similarity matrix S is updated by minimizing the objective function by keeping F fixed. This update not only learns graph connections that capture the self-expressive relationships among nodes, but also preserves the manifold structure of the data and ensures consistency with the original graph topology. In the next stage, the cluster indicator matrix F is learned by using the obtained graph structure via spectral clustering technique. Then, this updated cluster indicator matrix again would influence the next learning of S through the clustering regularization term $Tr(F^T L_S F)$. The learning method is repeated until the maximum number of iterations is reached. In the learning process, the similarity matrix S and the cluster indicator matrix F can be considered as model parameters that are learned from the data. The objective function in (5) defines the optimization framework for learning the similarity matrix S and the cluster indicator matrix F from the data. Each term imposes a desirable property on the learned matrices. Minimizing the objective function enables the model to learn a similarity matrix that captures self-expressive relationships, preserves manifold information, remains close to the original graph structure, and simultaneously provides clustering. Therefore, the alternating optimization procedure provides the learning mechanism through which the graph structure and clustering assignments are estimated from the data. The overall learning procedure of the proposed method is illustrated in Figure 2.

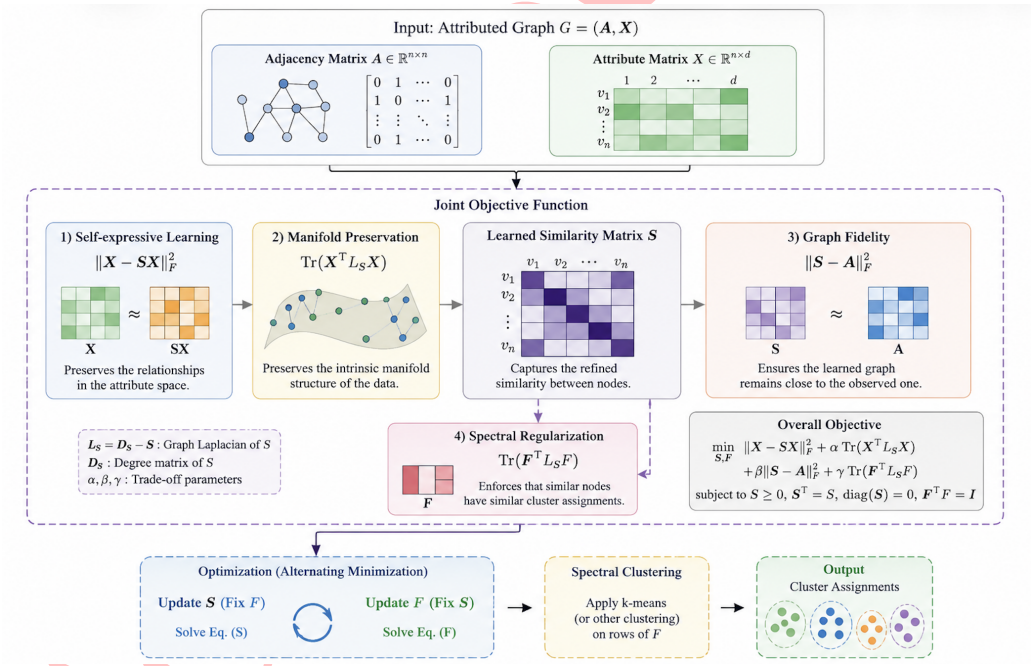


Figure 2: Overall learning framework of the proposed method. The similarity matrix is learned by jointly exploiting self-expressive learning, manifold preservation, graph fidelity, and spectral regularization. The matrices S and F are updated alternately until convergence to obtain the final clustering result.

3.4 Convergence proof

Algorithm 1 updates F and S in an alternative iteration scheme. If we show that each subproblem is convex then finding the optimal solution of each subproblem, would cause the iterative process be decreasing. It suffices to show that the objective function in (5) is monotonically non-increasing across epochs. Since the objective function is nonnegative and bounded below by zero, this monotonicity implies that the objective function converges to a finite limit. Hence in the following we show that each subproblem is convex with respect to its variable.

- Fix S: We show that the problem is convex with respect to F. To this aim, consider the following subproblem (8). The Hessian matrix of (8) is

$$\frac{\partial^2 Tr(F^T L_S F)}{\partial F \partial F^T} = L_S + L_S^T, \tag{9}$$

  where the Laplacian matrix L_S is positive semidefinite.Then, (8) is a convex function with respect to F in each iteration.

- Fix F: We show the convexity with respect to S. All terms of the objective function in (5) depend on S. The first and third terms in the norm are linear with respect to S, and the norm is also a convex function, so it is a convex function with respect to S.

  The proof that the second and fourth terms are convex is similar. Consider

$$\phi(S) = Tr(F^T L_S F) = \sum_{i,j=1}^{n} \|F_{i:} - F_{j:}\|_F^2 S_{ij}.$$

  For $\lambda \in [0,1]$ and $S, T \in \mathbb{R}^n$ we have

$$\begin{aligned}
\phi(\lambda S + (1-\lambda)T) &= \sum_{i,j} \|F_{i:} - F_{j:}\|_F^2 (\lambda S_{ij} + (1-\lambda) T_{ij}) \\
&= \sum_{i,j} \|F_{i:} - F_{j:}\|_F^2 \lambda S_{ij} + \sum_{i,j} \|F_{i:} - F_{j:}\|_F^2 (1-\lambda) T_{ij} \\
&= Tr(F^T L_{\lambda S + (1-\lambda)T} F) \\
&= Tr(F^T L_{\lambda S} F) + Tr(F^T L_{(1-\lambda)T} F) \\
&= \lambda Tr(F^T L_S F) + (1-\lambda) Tr(F^T L_T F).
\end{aligned}$$

  Consequently, all terms are convex with respect to S, and the sum of convex functions is also convex.

3.5 Computational complexity analysis

In this section, we provide the computational complexity of our method.

In each iteration, the similarity matrix S is updated according to Eq. (7). In this step, the main computational cost is related to the computation of the matrix product XX^T, which requires $O(n^2 d)$ operations. Since X is fixed during the optimization process, XX^T is computed once before starting the iterative process and then it is utilized in all iterations. To avoid computing the inverse of the large $n \times n$

matrix $(2\beta I + 2XX^T)$, we use the first-order Taylor approximation $(2\beta I + 2XX^T)^{-1} \approx \frac{1}{2\beta}\left(I - \frac{1}{\beta}XX^T\right)$. Therefore, the matrix inversion is avoided, and the remaining matrix operations for updating S have a complexity of $O(n^2)$. Thus, after the initial computation of XX^T, the computational complexity of updating S in each iteration is $O(n^2)$.

The cluster indicator matrix F is updated by solving the spectral clustering problem in Eq. (8), which requires computing the k eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix L_S. Hence its computational complexity is dominated by the eigenvalue decomposition and is approximately $O(n^3)$.

Therefore, the computational complexity of each iteration of the proposed algorithm is $O(n^2 + n^3) = O(n^3)$. In addition, the computation of XX^T requires a one-time preprocessing cost of $O(n^2 d)$. Although the first-order Taylor approximation avoids the explicit inversion of a large matrix and reduces the computational cost of updating S, the overall complexity of the proposed method remains $O(n^3)$ because the spectral decomposition is the dominant computational step.

It should be noted that the proposed method may face scalability limitations for very large graphs due to the $O(n^3)$ spectral decomposition. For very large-scale datasets, iterative eigensolvers or approximate spectral decomposition methods can be used to reduce the computational cost [52].

4 Results and discussion

In this section, we evaluate the performance of our proposed method by comparing it against several state-of-the-art approaches on various real-world datasets.

4.1 Datasets

Four famous benchmark datasets are often utilized to evaluate the performance of attributed graph clustering methods: Cora, Citeseer, Pubmed [28] and Wiki [30]. Cora, Citeseer, and Pubmed are citation networks and the last one, i.e. Wiki, is a webpage network. In the citation datasets, nodes determine the publications and edges stand for citations. In Wiki nodes denote webpages and two nodes are connected if they link each other. A brief introduction of the mentioned datasets is brought in Table 1.

Table 1: Datasets information

Dataset	Nodes	Edges	features	Classes
Cora	2708	5429	1433	7
Citeseer	3327	4732	3703	6
Wiki	2405	17981	4973	17
Pubmed	19717	44338	500	3

4.2 Evaluation metrics

The performance of the proposed method is evaluated via three popular metrics: Accuracy (ACC), Normalized Mutual Information (NMI) and F-score(F1). A brief introduction of them is brought in the following:

- ACC [27] is an evaluation metric that measures the number of data points which are contained in each cluster and are from the corresponding class. It is computed as follows:

$$ACC = \frac{1}{n} \max \left(\sum_{X_i Y_j} N(X_i, Y_j) \right),$$

where X_i and Y_j are the i-th cluster and j-th class, respectively, and $N(X_i, Y_j)$ counts points assigning to cluster i but are in j-th class.

- F1 [29] is a metric used to evaluate the performance of binary classification models. It combines two other famous metrics, i.e. precision and recall, into a single score. The F1 is defined as the harmonic mean of the mentioned metrics with the formulation as follows:

$$F1 = \frac{2 \times precision \times recall}{precision + recall}.$$

- NMI [27] is a widely popular evaluating measure which quantifies the statistical dependence of two variables. Consider $H(X)$ and $H(Y)$ to be the corresponding entropies of random variables X and Y. Then the NMI is defined as

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}},$$

where $I(X,Y)$ stands for mutual information between X and Y.

4.3 Comparing methods

Here we bring a brief introduction of comparing methods.

- The MGAE [54] uses auto-encoder techniques to learn compact representations of graphs and is based on a marginalization process that is specifically designed to reduce the effects of noise and improve the quality of graph representation.
- The VGAE (Variational Graph Auto-encoder) [28] is a neural network-based method in which the input graph is first transformed into a latent space via auto-encoder networks constructed by a graph convolutional network encoder and an inner product decoder are utilized. Then, the latent space is clustered based on k-Means or DBSCAN.
- The ARGA (Adversarial Graph Variational Auto-encoder) [45] is used to learn graph representations using adversarial training techniques. In this method, the latent representation is enforced to match a prior distribution via an adversarial training scheme.
- The AGC (Adaptive Graph Convolution method) [57] is capable to perform clustering in complex graphs with diverse features and complex relationships, and can adaptively use the graph structure to adjust the relationships of nodes.
- The SDCN (Structural Deep Clustering Network) [4] introduces a hybrid model of deep neural networks and clustering that has the ability to learn features and structure of data simultaneously.

- The AGE (Adaptive Graph Encoder for Attributed Graph Embedding) [9] dynamically learns the graph profiles based on the graph properties, and automatically adjusts weights and parameters based on the properties. This method has the ability to learn graph properties by combining the structural information of the graph and the properties of the nodes.
- The GALA [46] uses a deep graph-based neural network with a symmetric encoder and decoder designed to learn compact representations of graphs. The symmetry between the encoder and decoder allows for better learning of complex structures in graph data. It uses GCN layers to combine node information and graph structure into the final representations.
- The AMGC (Attribute-Missing Graph Clustering Network) [53] is a deep learning model for clustering graphs in which the missing attributes of nodes are automatically reconstructed by transferring information from neighboring nodes. Highly correlated updated features within the clusters would cause more intra-class compactness.
- The DGCLUSTER [3] is used for clustering attributed graphs through modularity maximization. The main goal of this method is to improve the clustering process of graphs with complex attributes, using a deep learning model that is able to identify optimal clusters.
- The GCOOL (Graph Communal Contrastive Learning) [37] is a framework to learn the community clusters and attribute matrix representation simultaneously via self-supervised contrastive train for community information.
- The GRACE [13] is a robust model for clustering attribute graphs that uses graph convolutional networks to learn graph features and cluster them. Based on building suitable graph Laplacians of graphs, this model obtains accurate clustering by using unsupervised learning and integrating structural and feature information.
- The NNMF (Non-Negative Matrix Factorization) [24] is a nonnegative matrix factorization based method in which additional information such as the graph structure, node neighbor relations, and node attributes are incorporated for clustering. In this method orthogonality constraints are set to for determining non-overlapping clusters.
- The KCAGC(K-means based Collaborative Attributed Graph Clustering) [58] is an attributed graph clustering method based on the k-means algorithm. Utilizing graph filters, it obtains filtered attribute sets from each local data and then it partitions data.
- The AOCD(Attention Overlapping Community Detection) [51] aggregates node attribute information via attention mechanisms which determine the the importance of neighboring nodes. More precisely, this method allows the model to dynamically weight the importance of neighboring nodes during the aggregation of node attribute information, thereby enhancing its ability to discover overlapping communities.

For a fair comparison, we divide the baseline methods into three categories: Traditional, Deep Graph Auto-encoders, and Advanced Learning, as shown in Table 2. Traditional methods have low computational complexity, deep learning methods have medium to high complexity, and most advanced approaches, have high computational complexity.

Table 2: Summary of the compared methods categorized by their architectural paradigms and learning objectives

Category	Method	Architecture Type	Learning Paradigm
Traditional	NNMF	Matrix Factorization	Linear/Traditional
	KCAGC	Graph Filter + K-means	Iterative/Filter-based
Deep GAE	VGAE	GCN Encoder-Decoder	Generative (Variational)
	MGAE	Auto-encoder	Marginalization-based
	ARGA	Adversarial GAE	Adversarial Learning
	GALA	Symmetric Encoder-Decoder	Deep Representation
Advanced	SDCN	Deep Hybrid Model	Joint Optimization
	AGC	Adaptive GCN	Adaptive Learning
	AGE	Adaptive Encoder	Dynamic Weighting
	DGCLUSTER	Deep Modularization	Modularity Maximization
	GCOOL	Contrastive GNN	Self-supervised Contrastive
	GRACE	GCN-based Laplacian	Unsupervised Learning
	AMGC	Deep Reconstruction	Attribute Reconstruction
	AOCD	Attention-based GNN	Attention Mechanism
Proposed	**Ours**	**Unified Iterative**	**Joint Optimization**

4.4 Comparative analysis

We evaluate the effectiveness of the proposed method on four widely-used benchmark datasets. Since the proposed method is fully unsupervised, no class labels are used during the learning process. The similarity matrix and clustering assignments are learned from the entire dataset. After convergence, the obtained clustering results are compared with the ground-truth labels only for evaluation using ACC, F1, and NMI. The comparative results are summarized in Tables 3, 4, and 5, respectively. To ensure a rigorous and unbiased comparison, the performance metrics of the baseline methods are adopted directly from their original published works. This approach ensures that our proposed method is compared against the optimized versions of these algorithms, as reported by their respective authors. A comparison with recent state-of-the-art methods demonstrates that our approach consistently outperforms existing baselines across most metrics. Based on the experimental results, we draw the following observations:

- Our method demonstrates significant advantages over various baseline algorithms, particularly in terms of ACC. The results indicates that our method outperforms all compared methods on datasets Cora, Citeseer and Wiki. The largest improvement is for Wiki which is 5.6% better than the second result achieved by AOCD. Also, for Cora and Citeseer datasets the improvements are 2.2% and 1.1% with respect to the second best methods AOCD and KCAGC, respectively. About Pubmed dataset, the proposed method gains the second score after NNMF method (around 2.7% less than it).

- The proposed method also has superior results considering the other methods with respect to F1 measure. Except Pubmed, it reaches the best results for all other datasets and for Pubmed it stands in the second-best place. Our method gained the largest improvement on Wiki with respect to the second best approach, , with 6.1%. It also had respectively 0.08% and 0.75% improvements for Cora and Citeseer with respect to the second results obtained by GCOOL method. Our new method is in the second place after KCAGC method for Pubmed dataset with 4.9% result decreasing.

Table 3: ACC clustering results

Method	Cora	Citeseer	Wiki	Pubmed
MGAE	63.43	63.56	50.14	43.88
VGAE	55.95	50.44	28.67	65.48
ARGA	64.00	57.30	41.40	59.12
AGC	53.25	41.26	17.33	64.08
SDCN	34.61	39.16	34.30	48.69
AGE	63.80	67.70	51.50	65.60
GALA	64.5	67.32	54.50	69.40
AMGC	66.65	60.92	51.7	64.56
DGCluster	58.54	55.16	47.30	55.63
GCOOL	67.40	66.98	53.64	68.97
GRACE	55.39	60.89	54.50	68.40
NNMF	56.98	52.78	55.02	**71.36**
KCAGC	67.51	68.40	46.50	69.36
AOCD	70.21	67.50	64.00	65.90
Our method	**71.75**	**69.12**	**67.58**	69.43

Table 4: F1 clustering results

Method	Cora	Citeseer	Wiki	Pubmed
MGAE	38.01	39.49	39.02	41.98
VGAE	41.50	31.88	20.49	50.95
ARGA	61.90	54.60	38.27	58.41
AGC	41.97	29.13	15.35	49.26
SDCN	21.78	33.14	21.80	38.78
AGE	37.30	43.30	31.70	31.50
GALA	53.20	44.60	38.90	32.17
AMGC	61.02	57.33	35.30	64.52
DGCluster	54.5	32.6	31.7	34.6
GCOOL	64.50	62.23	45.04	64.90
GRACE	50.50	60.17	36.19	64.68
NNMF	58.68	57.21	33.8	62.98
KCAGC	61.83	60.23	41.52	**68.51**
AOCD	63.90	60.40	61.00	60.80
Our method	**64.55**	**62.70**	**64.74**	65.17

- In terms of NMI, the new method obtains best results for Citeseer, Wiki and Pubmed datasets among all comparing methods. The most improvement is for Pubmed dataset which is 2.7% better than the result in KCAGC. NMI of our method has the improvement of 0.3% and 0.2% for Citeseer and Wiki datasets in AGE method which obtained the second results. For the Cora dataset, DGCluster achieved the best NMI result, outperforming the proposed method by 15.8%. However, the new method gained the second result and outperforms the other approaches.

The experimental results show different behaviors depending on the characteristics of each dataset. For the Citeseer and Wiki datasets, our method achieves the best performance. One possible reason is that, in these datasets, the node attributes and graph structure are more consistent with each other, al-

Table 5: NMI clustering results

Method	Cora	Citeseer	Wiki	Pubmed
MGAE	45.57	39.75	47.97	8.16
VGAE	38.45	22.71	30.28	25.09
ARGE	44.90	35.00	39.50	23.17
AGC	40.69	18.34	11.93	22.97
SDCN	12.56	16.19	14.20	4.99
AGE	49.30	41.09	52.24	30.00
GALA	50.70	40.10	50.40	32.70
AMGC	47.99	32.93	52.20	24.58
DGCluster	**62.10**	41.00	48.20	32.40
GCOOL	50.25	41.06	46.94	33.14
GRACE	34.65	31.14	31.43	32.60
NNMF	35.71	29.80	39.12	41.62
KCAGC	50.20	40.12	48.26	43.23
AOCD	50.00	40.70	35.00	29.00
Our method	52.29	**41.23**	**52.35**	**44.39**

lowing our method to effectively combine both sources of information. For the Cora dataset, although DGCLUSTER obtains a slightly higher NMI, our method achieves much better ACC and F1. This indicates that our method produces more accurate cluster assignments overall, while DGCLUSTER performs better only in terms of NMI. Therefore, our method provides a better balance among the different evaluation metrics. For the Pubmed dataset, different methods perform well on different metrics. NNMF achieves the highest ACC, while our method obtains the highest NMI. This suggests that linear models are more suitable for improving ACC on this dataset, whereas our method is more effective in capturing the underlying relationships in the data, leading to better NMI. Overall, our method achieves stable and competitive performance across all datasets. Unlike some competing methods whose performance changes considerably from one dataset to another, the proposed method maintains consistently good results, demonstrating its robustness on datasets with different characteristics.

It is noteworthy that the superior performance of our method remains consistent across multiple evaluation metrics (ACC, F1, and NMI) and diverse benchmark datasets. The stability of these results across datasets with varying structural characteristics suggests that the model learns robust representations. Unlike purely feature-based or structure-based approaches, our method jointly optimizes both components, which provides an implicit regularization effect and consequently reduces the risk of overfitting.

To evaluate the validity of the first-order Taylor approximation used in the optimization procedure, we first calculated its relative approximation error on the four benchmark datasets. Specifically, the approximation error is defined as

$$E = \left\|(I+aX)^{-1}\right\|_2 \left\|(I+aX)^{-1} - (I-aX)\right\|_2,$$

where $a = 1/\beta$. The obtained error values for Cora, Citeseer, Wiki, and Pubmed are 2.11×10^{-3}, 4.41×10^{-3}, 4.26×10^{-3}, and 3.12×10^{-3}, respectively. These values correspond to approximation errors of approximately 0.211%, 0.441%, 0.426%, and 0.312%. These results indicate that the first-order Taylor approximation provides a reasonable approximation to the inverse term under the parameter settings used in our experiments.

Table 6: Comparison of clustering performance between the exact inverse and the first-order Taylor approximation. The best results for each dataset and metric are shown in bold.

Dataset	Method	ACC (%)	F1 (%)	NMI (%)
Cora	Exact Inverse	69.33	63.02	**54.21**
	Taylor Approximation	**71.75**	**64.55**	52.29
Citeseer	Exact Inverse	68.12	60.14	39.91
	Taylor Approximation	**69.12**	**62.70**	**41.23**
Wiki	Exact Inverse	66.87	61.32	51.12
	Taylor Approximation	**67.58**	**64.74**	**52.35**
Pubmed	Exact Inverse	67.53	**65.31**	43.22
	Taylor Approximation	**69.43**	65.17	**44.39**

To examine the practical effect of the approximation, we compared the clustering performance obtained using the exact inverse and the first-order Taylor approximation. The results are shown in Table 6.

As can be seen in Table 6, Taylor-based update achieves higher ACC on all four datasets. It also achieves higher F1 scores on three datasets and higher NMI scores on three datasets. Overall, the Taylor approximation achieves better performance in 10 out of the 12 dataset-metric comparisons, while the exact inverse performs slightly better for Cora in terms of NMI and for Pubmed in terms of F1. These results indicate that the use of the Taylor approximation does not lead to a degradation in the final clustering performance in our experiments.

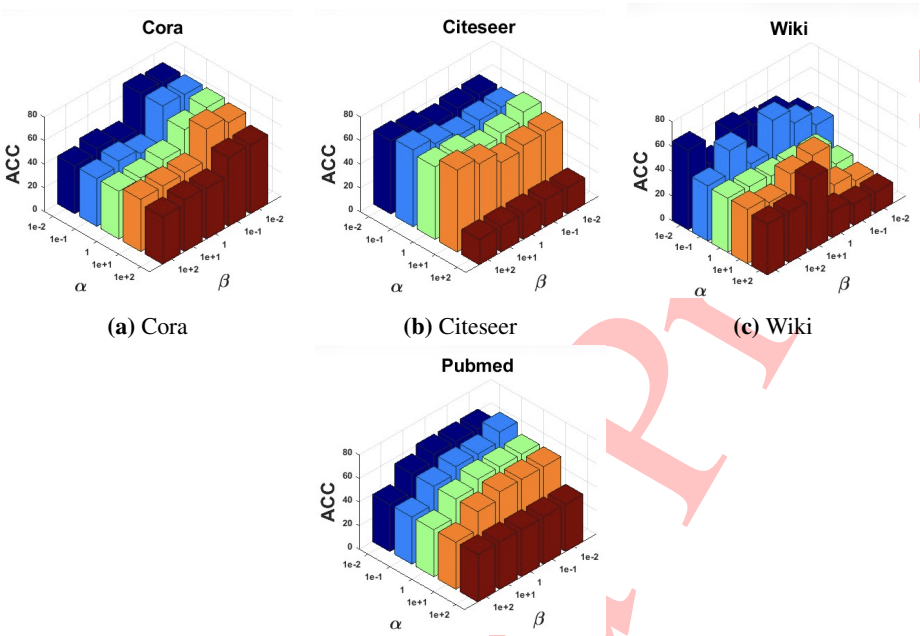
4.5 Parameter analysis

In our experiments, we use grid search, which has also been adopted in several related studies [1, 26, 36, 42, 56], to select the parameter values. Specifically, we consider $\alpha, \beta \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ and $\gamma \in \{10^{-1}, 10^{0}, 10^{1}\}$. Thus, the parameter values are selected from a predefined grid with five values for α, five values for β, and three values for γ.

For each dataset, we evaluate all parameter combinations in the specified grid and select the combination that gives the highest ACC. Therefore, the selected parameter values may be different for different datasets.

Here, we investigate the effects of changing α and β on the accuracy of the method, as shown in Figure 3. In these experiments, we set $\gamma = 1$ for Wiki and Pubmed, and $\gamma = 10$ for Core and Citeseer. The results show the following behavioral patterns:

- **Cora:** As demonstrated in Figure 3a, the performance is highly sensitive to β. Specifically, when β is held constant, variations in α (particularly for smaller values) result in negligible changes in accuracy, indicating that β dominates the learning process.
- **Citeseer:** Considering Figure 3b, the method shows distinct sensitivity to α at larger values (e.g., $\alpha = 1e^{+2}$), accuracy remains consistently low regardless of changes in β.
- **Wiki:** It is demonstrated in Figure 3c that while decreasing α and β generally leads to higher accuracy, the improvement does not follow a strictly monotonic trend compared to the other datasets.
- **Pubmed:** A clear, monotonic relationship is observed in Figure 3d in which decreasing both α and β systematically improves clustering accuracy.



(a) Cora **(b)** Citeseer **(c)** Wiki

(d) Pubmed

Figure 3: Parameter sensitivity analysis on (a) Cora, (b) Citeseer, (c) Wiki, and (d) Pubmed datasets. The plots show accuracy (ACC) as a function of α and β, with γ fixed. For Cora, the performance is more sensitive to β. For Citeseer, the changes are relatively smooth. For Wiki, the performance is sensitive to both α and β, with irregular changes. For Pubmed, the performance is sensitive to both α and β, with a more regular pattern.

In summary, reducing the values of α and β typically yields superior accuracy across the Cora, Citeseer, and Pubmed datasets. Although the Wiki dataset also benefits from smaller parameter values, the impact is less uniform, suggesting that the optimal balance between these parameters is dataset-dependent.

4.6 Ablation study

To evaluate the contribution of each component in our proposed objective function, we performed an ablation study by systematically removing each term—controlled by the parameters α, β, and γ. We examined the performance variations when each term is omitted. Table 7 summarizes the performance in terms of ACC, F1, and NMI. The results indicate that all terms are essential for achieving optimal performance. Specifically, removing the manifold regularization term (associated with α) leads to a performance drop. This confirms that the graph-based manifold regularization is critical to preserve the local geometric structure of the data in the learned similarity matrix. Removing the fidelity term, i.e. setting $\beta = 0$, we conclude from the results that it reduces the model's ability to stay consistent with the initial adjacency matrix A. This shows that this term is essential to guide the learning process using the prior structural information, for preventing the model from losing the original data connections. The γ term enables the model to jointly learn the similarity matrix S and the indicator matrix F. Without this term, F cannot be learned jointly with S and must instead be computed separately after optimizing S. Our results show that when $\gamma = 0$, performance drops and this shows that the spectral constraint is essential

for creating distinct clusters in the latent space. Consequently, the combination of these terms effectively balances structural information and node attributes which leads to efficient clustering performance.

Table 7: Ablation study results for different datasets

(a) Cora Dataset

(α, β, γ)	ACC	F1	NMI
(0.1, 10, 10)	71.75	64.55	52.29
(0, 10, 10)	61.13	48.14	48.32
(0.1, 0, 10)	67	61	50
(0.1, 10, 0)	69.01	62.11	51.13

(b) Citeseer Dataset

(α, β, γ)	ACC	F1	NMI
(0.001, 1, 100)	69.12	62.70	41.23
(0 , 1, 100)	65.17	51.12	38.01
(0.001, 0, 100)	67	62	41
(0.001, 1, 0)	64.13	51.21	37.14

(c) Wiki Dataset

(α, β, γ)	ACC	F1	NMI
(0.1, 100, 0.01)	67.58	64.74	52.35
(0, 100, 0.01)	58.10	51.22	31.24
(0.1, 0, 0.01)	43	34	41
(0.1, 100, 0)	50.65	50.18	29.00

(d) Pubmed Dataset

(α, β, γ)	ACC	F1	NMI
(0.1, 1, 0.0001)	69.43	65.17	44.39
(0 , 1, 0.0001)	58.70	51.22	31.24
(0.1, 0, 0.0001)	54	58	29.48
(0.1, 1, 0)	50.65	50.18	29.00

4.7 Robustness analysis with respect to clustering algorithms

To verify our method successfully improves the data representation, we conducted a robustness study. We compared our learned similarity matrix S against the initial adjacency matrix A using three different clustering approaches: K-means (centroid-based), Hierarchical Clustering (connectivity-based), and DBSCAN (density-based).

As shown in Table 8, the improvements gained from our proposed method are consistent across all tested paradigms. Specifically, the learned matrix S significantly outperforms the baseline A in terms of accuracy, NMI, and F1-score across all datasets. For instance, in the Cora dataset, K-means accuracy jumps from 29.95% with matrix A to 62.55% with matrix S. This widespread improvement demonstrates that our optimization process captures a more discriminative and robust latent structure that is universally beneficial, regardless of the clustering strategy used.

Interestingly, while all methods benefit from the learned matrix, spectral clustering consistently achieves the highest performance. This is because our objective function includes a spectral-based regularization term specifically designed to align the learned manifold with spectral properties. This term ensures that the optimized matrix S preserves complex, non-linear structures that are difficult for centroid-based or density-based methods to capture. Consequently, the inherent compatibility between the γ-term and the spectral decomposition process allows our framework to leverage the geometric potential of the learned affinity which leads to superior clustering of the datasets.

4.8 Convergence analysis

We showed the convergence of our method in Subsection 3.4. Here, we show the numerical results of convergence for the approach. The objective function values were obtained in each iteration for all

Table 8: Clustering study results for different datasets

(a) Cora Dataset

Method	Type	ACC	F1	NMI
spectral	S	71.75	64.55	52.29
	A	32.72	24.04	20.37
k-means	S	62.55	57.83	46.76
	A	29.95	8.13	1.71
Hier.	S	58.97	56.49	47.52
	A	37.74	21.16	14.71
DBSCAN	S	64.47	60.11	49.91
	A	25.23	10.32	3.51

(b) Citeseer Dataset

Method	Type	ACC	F1	NMI
spectral	S	69.12	62.70	41.23
	A	26.36	15.24	9.06
k-means	S	66.25	62.04	40.32
	A	21.55	7.06	1.95
Hier.	S	64.65	60.40	39.09
	A	24.52	11.15	6.51
DBSCAN	S	66.93	62.12	41.19
	A	21.12	8.61	2.82

(c) Wiki Dataset

Method	Type	ACC	F1	NMI
spectral	S	67.58	64.74	52.35
	A	25.36	16.23	22.20
k-means	S	50.21	36.56	45.05
	A	20.75	10.65	15.97
Hier.	S	49.03	36.19	41.51
	A	22.83	13.12	18.37
DBSCAN	S	39.00	31.03	42.29
	A	18.52	11.03	13.72

(d) Pubmed Dataset

Method	Type	ACC	F1	NMI
spectral	S	69.43	65.17	44.39
	A	13.88	11.22	6.85
k-means	S	41.14	37.42	25.85
	A	15.71	10.25	6.48
Hier.	S	43.63	38.44	25.42
	A	14.68	9.69	6.68
DBSCAN	S	43.67	39.35	42.63
	A	11.42	8.56	5.82

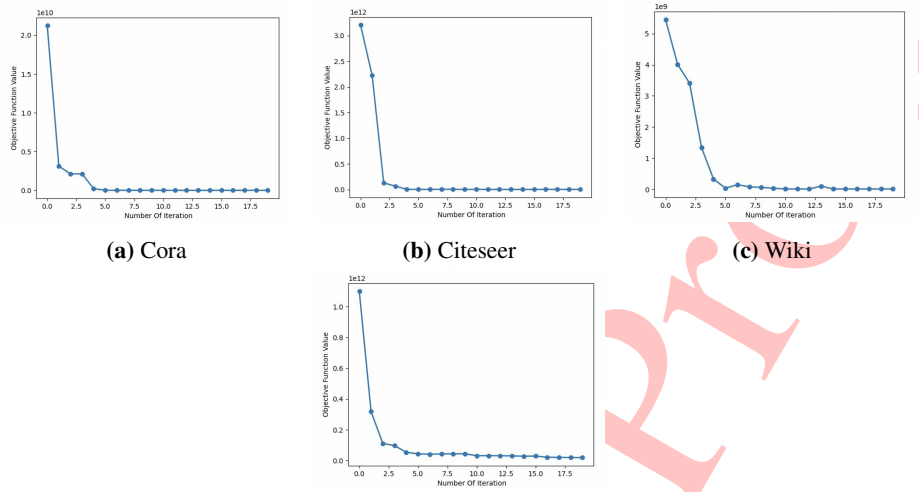
datasets. The results are shown in Figure 4. Based on this figure, it can be observed that the algorithm converges after 5 iterations for the Cora and Citeseer datasets. However, the convergence for Wiki and Pubmed datasets, is slightly slower and the algorithm converges in around iteration 10, respectively.

4.9 Limitations

Despite the promising performance of our method, it has several limitations. First, it involves three tuning parameters, α, β, and γ, whose values affects the clustering performance and appropriate selection is required via grid search. Second, the optimization process learns a similarity matrix of size $n \times n$, which increases the computational and memory requirements for very large-scale graphs. Third, the quality of the learned similarity matrix is partially influenced by the initial graph structure represented by A. If the original graph contains noisy or incomplete connections, the final clustering performance would be affected.

5 Conclusion

In this paper, we proposed a novel attributed graph clustering approach that jointly exploits the linear structure, manifold structure, and graph connectivity information of attributed data. The proposed



(a) Cora **(b)** Citeseer **(c)** Wiki

(d) Pubmed

Figure 4: The convergence behavior of the proposed method

method learns an adaptive similarity matrix through the self-expressive property. Moreover, the similarity matrix is learned in a way that preserves both the intrinsic geometric structure of the data and the consistency of the original graph topology, simultaneously. Unlike some conventional approaches that perform graph construction and clustering separately, our method integrates graph learning and spectral clustering into a unified optimization model which enables the learned graph structure to become more suitable for the clustering task. We applied an efficient alternating optimization algorithm to solve the proposed objective function. The convergence properties of the optimization procedure were also analyzed. Experimental evaluations conducted on four benchmark attributed graph datasets demonstrated the effectiveness of the proposed approach. The results showed that the proposed method achieved the best or second-best performance in most cases with respect to ACC, F1, and NMI, that outperforms several recent state-of-the-art attributed graph clustering methods. These results confirm that jointly modeling self-expressiveness, manifold preservation, and clustering information can significantly improve clustering results. Despite the good performance of our method, it still faces two main challenges when dealing with very large graphs: the sensitivity to parameter selection and the high computational cost. For future research, one may address these limitations in some ways. To improve efficiency, incorporating anchor node strategies can significantly reduce the computational complexity by approximating the full graph structure. Moreover, automated parameter tuning may make the framework more robust across different datasets. Also, One may extend this framework in two directions: adapting it to multi-view attributed graphs to better use diverse data sources, and exploring its potential for dynamic and heterogeneous graph structures.

Code and data availability

The implementation of the proposed method, together with the experimental scripts and a sample dataset Cora, is available at: https://github.com/miinajamshidi/Jointly-subspace-clustering-a

nd-manifold-learning-for-attributed-graphs.

Declarations

- No funding was received to assist with the preparation of this manuscript.
- The authors have no competing interests to declare that are relevant to the content of this article.
- We have used famous data which are available on repositories.

References

[1] M. Abousaeidi, M. Jamshidi, A. Armandnejad, *Multi-view spectral clustering based on considering co-proximity of views*, Pattern Recognit. (2025) 112919.

[2] E. Akbas, P. Zhao, *Graph clustering based on attribute-aware graph embedding*, From security to community detection in social networking platforms, (2019) 109–131.

[3] A. Bhowmick, M. Kosan, Z. Huang, A. Singh, S. Medya, *DGCLUSTER: A Neural Framework for Attributed Graph Clustering via Modularity Maximization*, Proc. AAAI Conf. Artif. Intell. **38** (2024) 11069–11077.

[4] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, P. Cui, *Structural deep clustering network*, Proc. Web Conf. (2020) 1400–1410.

[5] A. Bojchevski, S. Günnemann, *Bayesian robust attributed graph clustering: Joint learning of partial anomalies and group structure*, Proc. AAAI Conf. Artif. Intell. **32** (2018).

[6] I.D. Borlea, R.-E. Precup, F. Dragan, A.-B. Borlea, *Centroid update approach to K-means clustering*, Adv. Electr. Comput. Eng. **17** (2017) 3–10.

[7] H. Chen, Z. Yu, Q. Yang, J. Shao, *Attributed graph clustering with subspace stochastic block model*, Inf. Sci. **535** (2020) 130–141.

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

[9] G. Cui, J. Zhou, C. Yang, Z. Liu, *Adaptive graph encoder for attributed graph embedding*, Proc. 26th ACM SIGKDD Conf. Knowl. Discov. Data Min. (2020) 976–985.

[10] M.B. Dowlatshahi, H. Nezamabadi-Pour, *GGSA: a grouping gravitational search algorithm for data clustering*, Eng. Appl. Artif. Intell. **36** (2014) 114–121.

[11] X. Dong, D. Thanou, M. Rabbat, P. Frossard, *Learning graphs from data: A signal representation perspective*, IEEE Signal Process. Mag. **36** (2019) 44–63.

[12] U. E. Essiz, C. I. Aci, E. Sarac, M. Aci, *Deep learning-based prediction models for the detection of vitamin D deficiency and 25-hydroxyvitamin D levels using complete blood count tests*, Rom. J. Inf. Sci. Technol. **27** (2024) 295–309.

[13] B. Fanseu Kamhoua, L. Zhang, K. Ma, J. Cheng, B. Li, B. Han, *Grace: A general graph convolution framework for attributed graph clustering*, ACM Trans. Knowl. Discov. Data **17** (2023) 1–31.

[14] C. Fettal, L. Labiod, M. Nadif, *Scalable attributed-graph subspace clustering*, Proc. AAAI Conf. Artif. Intell. **37** (2023) 7559–7567.

[15] Y. Gan, Z. Gu, G. Xu, G. Zou, *Multi-view contrastive deep subspace clustering for attributed graph*, Expert Syst. Appl. **307** (2025) 130921.

[16] C. X. Gao, D. Dwyer, Y. Zhu, C. L. Smith, L. Du, K. M. Filia, J. Bayer et al., *An overview of clustering methods with guidelines for application in mental health research*, Psychiatry Res. **327** (2023) 115265.

[17] A. Grover, J. Leskovec, *node2vec: Scalable feature learning for networks*, Proc. 22nd ACM SIGKDD Conf. Knowl. Discov. Data Min. (2016) 855–864.

[18] S. Günnemann, I. Färber, B. Boden, T. Seidl, *Subspace clustering meets dense subgraph mining: A synthesis of two paradigms*, 2010 IEEE Int. Conf. Data Min. (2010) 845–850.

[19] S. Günnemann, B. Boden, T. Seidl, *DB-CSC: a density-based approach for subspace clustering in graphs with feature vectors*, Mach. Learn. Knowl. Discov. Databases (2011) 565–580.

[20] T. Guo, S. Pan, X. Zhu, C. Zhang, *CFOND: Consensus factorization for co-clustering networked data*, IEEE Trans. Knowl. Data Eng. **31** (2018) 706–719.

[21] X. Guo, L. Gao, X. Liu, J. Yin, *Improved deep embedded clustering with local structure preservation*, IJCAI'17: Proceedings of the 26th International Joint Conference on Artificial Intelligence, (2017) 1753–1759.

[22] D. He, Z. Feng, D. Jin, X. Wang, W. Zhang, *Joint identification of network communities and semantics via integrative modeling of network topologies and node contents*, Proc. AAAI Conf. Artif. Intell. (2017) 116–124.

[23] G. E. Hinton, R. R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*, Science **313** (2006) 504–507.

[24] V. Jannesari, M. Keshvari, K. Berahmand, *A novel nonnegative matrix factorization-based model for attributed graph clustering by incorporating complementary information*, Expert Syst. Appl. **242** (2024) 122799.

[25] Z. Kang, Z. Lin, X. Zhu, W. Xu, *Structured graph learning for scalable subspace clustering: From single view to multiview*, IEEE Trans. Cybern. **52** (2021) 8976–8986.

[26] Z. Kang, Z. Liu, S. Pan, L. Tian, *Fine-grained attributed graph clustering*, Proc. SIAM Int. Conf. Data Min. (SDM) (2022) 370–378.

[27] Z. Kang, G. Shi, S. Huang, W. Chen, X. Pu, J. T. Zhou, Z. Xu, *Multi-graph fusion for multi-view spectral clustering*, Knowl.-Based Syst. **189** (2020) 105102.

[28] T. N. Kipf, M. Welling, *Variational graph auto-encoders*, arXiv preprint arXiv:1611.07308 (2016).

[29] Z. Kong, Z. Fu, D. Chang, Y. Wang, Y. Zhao, *One for all: A novel dual-space Co-training baseline for Large-scale Multi-View Clustering*, arXiv preprint arXiv:2401.15691 (2024).

[30] Q. Li, X.M. Wu, H. Liu, X. Zhang, Z. Guan, *Label efficient semi-supervised learning via graph filtering*, Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (2019) 9582–9591.

[31] Y. Li, X. Chen, *Attributed graph subspace clustering with residual compensation guided by adaptive dual manifold regularization*, Expert Syst. Appl. **255** (2024) 124699.

[32] X. Li, D. Li, R. Jin, R. Ramnath, G. Agrawal, *Deep Graph Clustering with Random-walk based Scalable Learning*, Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Min. (ASONAM) (2022) 88–95.

[33] S. Li, L. Wan, Y. Zhang, L. Luo, *Weighted symmetric nonnegative matrix factorization and graph-boosting to improve the attributed graph clustering*, Eng. Appl. Artif. Intell. **142** (2025) 109914.

[34] W. Li, S. Wang, X. Guo, E. Zhu, *Deep graph clustering with multi-level subspace fusion*, Pattern Recognit. **134** (2023) 109077.

[35] M. Li, Z. Yang, X. Zhou, Y. Fang, K. Li, K. Li, *Clustering on attributed graphs: From single-view to multi-view*, ACM Comput. Surv. **57** (2025) 1–36.

[36] X. Lin, R. Yang, H. Zheng, X. Ke, *Spectral Subspace Clustering for Attributed Graphs*, Proc. 31st ACM SIGKDD Conf. Knowl. Discovery Data Min. **1** (2025) 789–799.

[37] Y. Liu, X. Gao, T. He, T. Zheng, J. Zhao, H. Yin, *Reliable node similarity matrix guided contrastive graph clustering*, IEEE Trans. Knowl. Data Eng. (2024).

[38] M. Mohammadi, K. Berahmand, S. Forouzandeh, X. Zhou, H. Khosravi, *Dual-view entropy-regularized nonnegative matrix factorization for attributed graph clustering*, Inf. Sci. **325** (2025) 122566.

[39] H. Motallebi, M. Jamshidi, *A distance scaling method to improve spectral clustering of data with different densities*, Int. J. Data Sci. **6** (2021) 328–347.

[40] H. Motallebi, R. Nasihatkon, M. Jamshidi, *A local mean-based distance measure for spectral clustering*, Pattern Anal. Appl. **25** (2022) 351–359.

[41] A. Moslemi, M. Jamshidi, *Unsupervised feature selection using sparse manifold learning: Auto-encoder approach*, Inf. Process. Manag. **62** (2025) 103923.

[42] M. Noroozi, M. Salahi, S. Eskandari, *Feature selection via mixed-integer program and supervised infinite feature selection method*, J. Math. Model. **13(2)** (2025) 341–356.

[43] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, P. Vandergheynst, *Graph signal processing: Overview, challenges, and applications*, Proc. IEEE **106** (2018) 808–828.

[44] G. J. Oyewole, G. A. Thopil, *Data clustering: application and trends*, Artif. Intell. Rev. **56(7)** (2023) 6439–6475.

[45] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, C. Zhang, *Adversarially regularized graph autoencoder for graph embedding*, arXiv preprint arXiv:1802.04407 (2018).

[46] J. Park, M. Lee, H. J. Chang, K. Lee, J. Y. Choi, *Symmetric graph convolutional autoencoder for unsupervised graph representation learning*, Proc. IEEE/CVF Int. Conf. Comput. Vis. (2019) 6519–6528.

[47] B. Perozzi, R. Al-Rfou, S. Skiena, *Deepwalk: Online learning of social representations*, Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (2014) 701–710.

[48] Z. Peng, H. Liu, Y. Jia, J. Hou, *Adaptive attribute and structure subspace clustering network*, IEEE Trans. Image Process. **31** (2022) 3430–3439.

[49] Z. Qin, H. Chen, Z. Yu, Q. Yang, J. Shao, *Attribute enhanced random walk for community detection in attributed networks*, Neurocomputing (2024) 128826.

[50] F. Sadjadi, V. Torra, M. Jamshidi, *Preprocessed Spectral Clustering with Higher Connectivity for Robustness in Real-World Applications*, Int. J. Comput. Intell. Syst. **17** (2024) 86.

[51] K. Sismanis, P. Potikas, D. Souliou, A. Pagourtzis, *Overlapping community detection using graph attention networks*, Future Gener. Comput. Syst. **163** (2025) 107529.

[52] P. Tang, E. Polizzi, *FEAST as a subspace iteration eigensolver accelerated by approximate spectral projection*, SIAM J. Matrix Anal. Appl. **35(2)** (2014) 354–390.

[53] W. Tu, R. Guan, S. Zhou, C. Ma, X. Peng, Z. Cai, Z. Liu, J. Cheng, X. Liu, *Attribute-missing graph clustering network*, Proc. AAAI Conf. Artif. Intell. **38(14)** (2024) 15392–15401.

[54] C. Wang, S. Pan, G. Long, X. Zhu, J. Jiang, *Mgae: Marginalized graph autoencoder for graph clustering*, Proc. ACM SIGIR Conf. Inf. Retr. (CIKM) (2017) 889–898.

[55] T. Yang, R. Jin, Y. Chi, S. Zhu, *Combining link and content for community detection: a discriminative approach*, Proc. KDD (2009) 927–936.

[56] X. Zhang, X. Xie, Z. Kang, *Graph Learning for Attributed Graph Clustering*, Mathematics **10(24)** (2022) 4834.

[57] X. Zhang, H. Liu, Q. Li, X.-M. Wu, *Attributed graph clustering via adaptive graph convolution*, arXiv preprint arXiv:1906.01210 (2019).

[58] R. Zhang, X. Hou, Z. Tian, Y. He, E. Gong, J. Liu, Q. Wu, K. Ren, *Attributed Graph Clustering in Collaborative Settings*, IEEE Trans. Dependable Secur. Comput. **22(3)** (2025) 3093–3104.