Contextual Correlation Preserving Multi-View Featured Graph Clustering

Tiantian He, Yang Liu, *Senior Member, IEEE,* Tobey H. Ko, Keith C.C. Chan, and Yew-Soon Ong, *Fellow, IEEE*

***Abstract*—Graph clustering, which aims at discovering sets of related vertices in graph structured data, plays a crucial role in various applications such as social community detection and biological module discovery. With the huge increase in the volume of data in recent years, graph clustering is used in an increasing number of real-life scenarios. However, the classical and state-of-the-art methods, which consider only single-view features or a single vector concatenating features from different views and neglect the contextual correlation between pairwise features, are insufficient for the task, as features that characterize vertices in a graph are usually from multiple views and the contextual correlation between pairwise features may influence the cluster preference for vertices. To address this challenging problem, we introduce in this study a novel graph clustering model, dubbed Contextual Correlation Preserving Multi-View Featured Graph Clustering (CCPMVFGC), for discovering clusters in graphs with multi-view vertex features. Unlike most of the aforementioned approaches, CCPMVFGC is capable of learning a shared latent space from multi-view features as the cluster preference for each vertex and making use of this latent space to model the interrelationship between pairwise vertices. CCPMVFGC uses an effective method to compute the degree of contextual correlation between pairwise vertex features, and utilizes view-wise latent space representing the feature-cluster preference to model the computed correlation. Thus, the cluster preference learned by CCPMVFGC is jointly inferred by multi-view features, view-wise correlations of pairwise features, and the graph topology. Accordingly, we propose a unified objective function for CCPMVFGC and develop an iterative strategy to solve the formulated optimization problem. We also provide the theoretical analysis of the proposed model, including convergence proof and computational complexity analysis. In our experiments we extensively compare the proposed CCPMVFGC with both classical and state-of-the-art graph clustering methods on eight standard graph datasets (six multi-view and two single-view datasets). The results show that CCPMVFGC achieves competitive performance on all eight datasets, which validates the effectiveness of the proposed model.**

***Index Terms*—Attributed graph, community detection, complex network, graph clustering, contextual correlation, multi-view features, contextual correlation preserving multi-view featured graph clustering (CCPMVFGC)**

I. Introduction

Many real-world data can be represented as graphs in which the data samples and their interrelationships can be readily represented by terminologies such as vertices (nodes) and edges (links). For example, a social graph can be used to model the users (vertices/nodes) and their friendship (edges/links) in a social networking site. With the huge increase in data in recent years, knowledge discovery from graph-structured data has attracted much interest from the learning community. Unlike graphs that are generated randomly, real-world graphs possess latent structures with cohesive latent features shared among the vertices. Graph cluster (community) is an important latent feature of the graphs, as it reveals information that could be useful in various applications. Thus, an increasing number of studies have recently aimed at discovering graph clusters [15], [29], and many analytical applications in areas such as biology [1], neural sciences [20], and social networks [31], [58] have emerged that use the technique.

Graph clustering aims at discovering sets of related vertices in graphs, and has long been a challenging problem. To unfold the set of subgraphs with vertices that share similar latent features, both model-based and heuristic-based representative approaches attempt to maximize the topological similarity shared by the vertices within each cluster using different types of topological information, which can be either the edges in the graph, or edge weights that reveal the topological similarity between the connected vertices. For example, the Clauset-Newman-Moore algorithm (CNM) [10] and fast unfolding [7] are two classical heuristic approaches based on modularity optimization [37]. Model-based methods, such as the Stochastic Block Models (SBM) [1], [41], Communities through Directed Affiliations (CoDA) [59], and semi-supervised matrix factorizations [60] are also able to discover graph clusters using edge information. Other typical approaches, including Normalized Cut (NCut) [43], Affinity Propagation [17], Hidden Community Detection [23], and Lambda Correlation Clustering (LambdaCC) [47], are able to perform graph clustering using edge weights.

In addition to graph topology, the content data in graphs, e.g., profiles describing social network users in social graphs or fingerprints describing functions of biological units in biological graphs, can be considered as vertex features that characterize the vertices from a semantic perspective. Attempts to discover clusters in the graphs have been made by utilizing vertex features rather than the topological information carried by graph data. For example, k-means clustering [35] is capable of discovering clusters in the graph utilizing vertex features. Multi-assignment clustering (Mac) [16] is a typical

T. He and Y.S. Ong are with School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: {tiantian.he,asysong}@ntu.edu.sg). Y. Liu is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China and also with the HKBU Institute of Research and Continuing Education (IRACE), Shenzhen, China (e-mail: csygliu@comp.hkbu.edu.hk). T.H. Ko is with the Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong SAR, China (e-mail: tobeyko@hku.hk). K.C.C. Chan is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: cskcchan@comp.polyu.edu.hk).

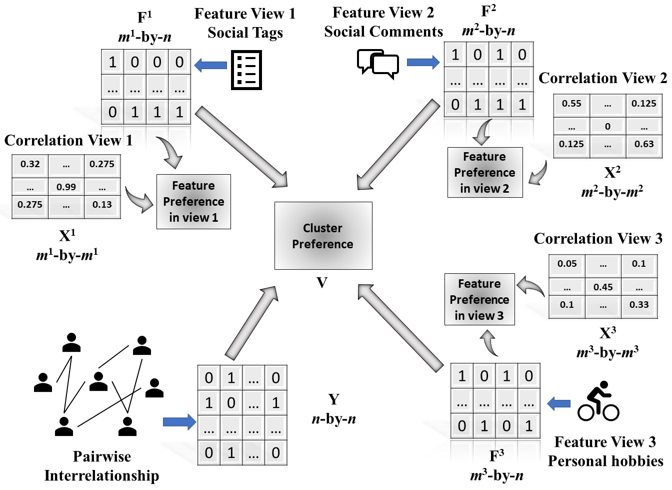

Fig. 1. Schematic illustration of the idea behind the proposed Contextual Correlation Preserving Multi-View Featured Graph Clustering (CCPMVFGC).

probabilistic generative model for discovering graph clusters where vertex features are binary values.

Rather than solely using graph topology or vertex features, many recent approaches consider both when performing graph clustering tasks. Here, a set of features are involved to characterize the vertices alongside the topological information, so that Attributed Graphs/Networks (AGs) [8], [25] are defined, to distinguish from those graphs containing only the topological information. In AGs, clusters with vertices sharing similar vertex features and topological information are of great interest.

AG clustering, which has edges and vertex features as two separate data views, can be considered as a special case of co-learning, and is typically solved through the following two methodologies. First, a new graph can be constructed where edges are weighted using an appropriate similarity measure, considering both topology and features between pairwise vertices, and then a common clustering model, such as spectral clustering [21], Expectation-Maximization Clustering [5], or Markov clustering [32], [62], [63], is used to unfold the clusters in the newly constructed graph. Second, a co-learning model can be built to infer a low dimensional latent space shared by the edge structure and vertex features. Many machine learning models have been adapted so they can discover such shared latent spaces. Inspired by topic modeling [6], several model-based approaches, such as Relational Topic Models (RTM) [9], iTopic Models [44], and Block Latent Dirichlet Allocation (BlockLDA) [4], have been proposed to learn a low dimensional latent space hidden in both graph topology and vertex features. Communities from Edge Structure and Node Attributes (CESNA) [58], General Bayesian framework for Attributed Graph Clustering (GBAGC) [57], and Circles [36] are three probabilistic generative models that are able to learn a latent space compactly representing the cluster preference for each vertex, regarding feature and structural similarity. In addition to probabilistic models, other techniques such as matrix factorization and evolutionary optimization have also been used for AG clustering. For example, Mining Interesting Sub-Graphs (MISAGA) [25] and Fuzzy Structural Pattern discovery for Graph Analytics (FSPGA) [24] are two approaches that can discover clusters in AG using edge and content similarity between vertices. More recently, multiobjective genetic algorithm for attribute graphs (MOGA-@net) [38] was proposed which utilizes evolutionary approach to uncover clusters in AG through maximizing connectivity and similarity of node attributes in clusters.

In today's hyper-connected networked economy, data with multi-view vertex features are easily accessible. Using the online social network as an example, as depicted in Figure 1, a graph can be modeled using the social network with the users modeled as vertices and their friendship modeled as edges. In addition to these basic elements, additional information such as social tags, comments, and personal hobbies of each user can all be utilized to characterize the vertices. Under this circumstance, an AG can be further redefined as a multi-view featured graph (MVFG), where the vertex features used to characterize the vertices are derived from more than one view.

Despite the availability of data, there is a lack of effective approaches for discovering clusters in real graphs with multi-view features or modeling graph edges using multi-view features, given that features from different views may impact both the edge and cluster preferences of each vertex in the graph. While the problem of graph clustering with multi-view vertex features has not been extensively investigated, multi-view learning in general has recently been the focus of much attention [56] and has been widely studied in many tasks such as image segmentation and clustering [2], [49]–[52], image retrieval [53], [54], object recognition [18], [19], traffic trajectory analysis [61], and social media analytics [33], [34]. Moreover, the role of vertex features in graph clustering has not been fully explored. Most previous approaches to AG clustering consider the similarity of features associated with pairwise vertices, but neglect the contextual concurrency between different vertex features, which may potentially indicate an intimate relationship between a pair of vertices whose features are disparate. Thus, to take advantage of the growing volume of multi-view featured data in today's networked economy, properly defining graph clustering problems in a multi-view featured setting, and developing effective approaches to further explore the role of feature correlations in graph clustering, are of great benefit.

In this paper, we propose a novel graph clustering model called Contextual Correlation Preserving Multi-View Featured Graph Clustering (CCPMVFGC) as an effective solution to the multi-view featured graph clustering problem. Figure 1 illustrates how CCPMVFGC works to discover clusters in a social graph with multi-view vertex features. Representing the multi-view vertex features and edge as independent data matrices, i.e., $\mathbf{F}^i$ $(i = 1, 2, 3)$ and $\mathbf{Y}$ as shown in the figure, CCPMVFGC assumes the edge structure $\mathbf{Y}$ is generated by the vertex-cluster preference $\mathbf{V}$, which is a shared latent space learned from $\mathbf{F}^i$s. In addition, CCPMVFGC constructs another matrix in each view, i.e. $\mathbf{X}^i$ $(i = 1, 2, 3)$, representing the contextual correlation between pairwise features and assumes such correlations are generated by the view-wise latent space representing the feature-cluster contributions.

Given such assumptions, the learning of cluster preference $\mathbf{V}$ may effectively acquire the view-wise effect of vertex features, and be regularized by contextual correlations between pairwise features, so that more meaningful clusters are expected to be uncovered. Note that the problem tackled by CCPMVFGC in this paper is fundamentally different from the problem of clustering in heterogeneous graphs [45], which aims at discovering clusters in a set of graphs having multi-type vertices and edges. The graph data dealt by CCPMVFGC, however, have single-type vertices and edges but multi-view vertex features.

The contributions of this paper are summarized as follows:

- We propose CCPMVFGC, which is a novel and generic model that combines graph clustering with multi-view learning, to perform the clustering task in the multi-view featured graph. Unlike previous methods, CCPMVFGC attempts to learn the feature-cluster preference in each view from both vertex features and their contextual correlations, so that such preference can well capture the contextual interdependency of features in each cluster, and induce the model to infer similar cluster preferences for those vertices sharing more correlated features.
- We design a unified objective function for CCPMVFGC and develop an iterative strategy to solve the formulated optimization problem. In addition, we provide the theoretical analysis of the proposed model, including convergence proof and computational complexity analysis.
- We extensively compare the proposed CCPMVFGC with both classical and state-of-the-art graph clustering methods on eight standard graph datasets. The results show that CCPMVFGC performs the best on most datasets, which validates the effectiveness of the proposed method.

The remainder of this paper is organized as follows. In Section II, we present the details of the proposed method, including the notations, the formulation, the proposed objective function, and the optimization procedure. In Section III, we analyze the algorithmic properties of the proposed method, including the algorithm convergence and computational complexity. Section IV provides a series of experiments on eight standard graph datasets to demonstrate the behavior of the proposed method and validate its effectiveness. In Section **??**, an in-depth discussion is conducted to distinguish CCPMVFGC from approaches that learn another latent feature in AG, known as vertex embedding. Finally, we conclude the paper and discuss the future work in Section V.

II. Multi-View Featured Graph Clustering Preserving Contextual Correlation

In this section, we elaborate the proposed Contextual Correlation Preserving Multi-View Featured Graph Clustering (CCPMVFGC) model. First we introduce the mathematical notations used in this paper. We then present two important components of the proposed model, i.e., the view-wise contextual correlation preserving feature clustering and interrelationship modeling, and formulate the objective function of CCPMVFGC. Finally, we develop an iterative strategy to solve the formulated optimization problem.

TABLE I
Notations used by CCPMVFGC

Notation	Meaning
n	Number of vertices
m	Number of features
m^i	Number of features in view i
d	Number of views of vertex features
k	Number of clusters to be discovered
$\mathbf{Y}$	Vertex adjacency/interrelationship matrix
$\mathbf{F}^i$	Node-feature matrix in view i
$\mathbf{X}^i$	Feature correlations in view i
$\mathbf{V}$	$n \times k$ latent space representing vertex-cluster preference
$\mathbf{U}^i$	$m^i \times k$ latent space representing feature-cluster preference in feature view i
α, λ	Model parameters

A. Notations

Given a graph composed of n vertices, |E| edges, d views of vertex features, with view i containing m^i features, and $\sum_{i=1}^{d} m^i = m$, we use two binary matrices $\mathbf{Y} \in \{0,1\}^{n \times n}$ and $\mathbf{F}^i \in \{0,1\}^{m^i \times n}$ to represent whether two vertices are connected and whether a vertex has a corresponding feature in view i, respectively. In addition, we use a non-negative real matrix $\mathbf{X}^i$ with the size $m^i \times m^i$ to represent the contextual correlations between pairwise features in view i. To model the graph clustering problem, CCPMVFGC uses the $n \times k$ latent space $\mathbf{V}$, and the $m^i \times k$ latent space $\mathbf{U}^i$ to represent the shared vertex-cluster preference, and the feature-cluster contributions in view i, respectively. The (i,j)-th element of a matrix $\mathbf{Y}$ is denoted as $\mathbf{Y}_{ij}$. The Frobenius norm is denoted as $\|\cdot\|_F$ and the trace operation is denoted as $tr(\cdot)$. Table I summarizes the notations used in this paper.

B. Contextual Correlation Preserving View-wise Feature Clustering

Vertex features from different views and their correlations may have view-wise effects influencing both the interrelationship modeling and the cluster preferences. To effectively acquire such latent effects from multi-view vertex features, for each view, CCPMVFGC first constructs an additional matrix representing the contextual correlations between these features using a word-embedding approach, and then attempts to infer two latent spaces ($\mathbf{V}$ and $\mathbf{U}^i$), representing the vertex-cluster preference and feature-cluster contributions in view i, respectively.

1) Identifying the correlations between pairwise features: As the correlations between pairwise features may reveal whether a pair of features is coinstantaneously observed with a high frequency, which may affect the latent feature preference for each cluster, CCPMVFGC considers the correlations to uncover more meaningful cluster structure in the attributed graph. Specifically, CCPMVFGC uses the shifted point-wise mutual information (SPMI) method [30] to compute the contextual correlations between each pair of features. Then, the obtained contextual correlations between each pair of vertex features can be normalized using the negative entropy of the corresponding feature pairs. Given two features, say f_j and

f_k in view i, the contextual correlation between them can be calculated as:

$$\begin{aligned}
&\mathbf{X}^i_{jk} = \begin{cases} \frac{PMI(f_j,f_k)}{H(f_j,f_k)} & \text{if } PMI(f_j,f_k) \geq log\delta, \\ 0 & \text{otherwise,} \end{cases} \\
&PMI(f_j,f_k) = log\frac{obs(f_j,f_k)\cdot|obs|}{obs(f_j,+)\cdot obs(f_k,+)}, \\
&H(f_j,f_k) = -log\frac{obs(f_j,f_k)}{|obs|},
\end{aligned} \tag{1}$$

where $obs(f_j,f_k)$, $|obs|$, and $obs(f_j,+)$ denote the number of observations in which f_j and f_k are associated with the same vertex, the total number of observations in which two features are both associated with the same vertex, and the number of observations that f_j correlates to other features, respectively. Note that the contextual correlation $\mathbf{X}^i_{jk}$, represented by the quotient of mutual information between f_j and f_k by the negative entropy of $p(f_j,f_k)$, can be simplified by canceling out the same factor $p(f_j,f_k)$ from both the numerator and the denominator for a concise representation.

By varying the value of δ, the obtained $\mathbf{X}^i_{jk}$ is equivalent to the value inferred via word2vec with a negative sampling value of δ [30]. Accordingly, CCPMVFGC is able to model the feature-cluster preference $\mathbf{U}^i$ using both view-wise vertex features $\mathbf{F}^i$ and their contextual correlations $\mathbf{X}^i$, so that a $\mathbf{U}^i$ with more feature interdependence is expected to be learned. In this study, δ is simply set to 1 to allow CCPMVFGC to capture more correlations between vertex features in the clustering process.

2) View-wise feature clustering: Given $\mathbf{F}^i$ and $\mathbf{X}^i$ in each feature view, CCPMVFGC assumes that $\mathbf{X}^i$ is generated by the product of $\mathbf{U}^i$ and $\mathbf{U}^{iT}$, and that $\mathbf{F}^i$ is generated by the product of $\mathbf{U}^i$ and $\mathbf{V}^T$. Thus we have $\mathbf{X}^i_{jk}$ = $[\mathbf{U}^i\mathbf{U}^{iT}]_{jk}$+$\phi_{jk}$ and $\mathbf{F}^i_{jk}$ = $[\mathbf{U}^i\mathbf{V}^T]_{jk}$+$\xi_{jk}$, where ϕ_{jk} and ξ_{jk} denote the error terms. Accordingly, the view-wise feature clustering problem can be formulated as minimizing the following cost function:

$$O_1 = \sum_i [\left\|\mathbf{F}^i - \mathbf{U}^i\mathbf{V}^T\right\|_F^2 + \left\|\mathbf{X}^i - \mathbf{U}^i\mathbf{U}^{iT}\right\|_F^2 + \Omega(\mathbf{U}^i), \tag{2}$$

where $\Omega(\mathbf{U}^i) = \sum_i\sum_j(\sum_k \mathbf{U}^i_{jk})^2$ is the regularization term for controlling the sparseness of $\mathbf{U}^i$. Compared with the traditional L_1 norm regularizer, $\Omega(\mathbf{U}^i)$ could further improve the sparsity of each row in $\mathbf{U}^i$ [26], so that the learned $\mathbf{U}^i_{jk}$ for features unrelated to certain clusters will approach zero.

By minimizing the cost function in Eq. (2), CCPMVFGC can obtain a shared optimal preference between n vertices and k clusters, and a view-wise preference between m^i features in view i and k clusters, respectively.

C. Interrelationship Modeling

Interrelationship modeling is an essential component in graph clustering, as the edge structure is the cornerstone of graph data. An appropriate modeling method may faithfully reveal the relationship between the latent cluster preference and the original edge structure.

Typically, interrelationship modeling is conducted by assuming the edge structure is generated by the latent space. Although such approach is effective to some extent, it is vulnerable to noisy edges, e.g., edges that connect vertices whose local structures are different. Inspired by diffusion theory [12], we propose the following re-weighting method to make the graph structure robust to noisy data:

$$\mathbf{D}_{ij} = \begin{cases} \frac{[\mathbf{Y}^T\mathbf{Y}+\mathbf{Y}]_{ij}}{2d_i} + \frac{[\mathbf{Y}^T\mathbf{Y}+\mathbf{Y}]_{ij}}{2d_j} & \text{if } \mathbf{Y}_{ij} = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where d_i denotes the degree of vertex i in the graph and $\mathbf{D}_{ij}$ measures the mutual diffusion in terms of local structure between connected vertices pairs. When two vertices, say v_i and v_j are connected, $\mathbf{D}_{ij}$ is computed by summarizing the bi-directional probabilities in which the shared local structure is propagated from an end point to another. The computed $\mathbf{D}$ is then used to re-weight the corresponding edges in $\mathbf{Y}$, i.e., $\mathbf{Y}_{ij} = \mathbf{D}_{ij}$, thereby the existence of linkage and its corresponding structural propagationality are incorporated into the newly constructed $\mathbf{Y}$.

Given O_1 in Eq. (2), CCPMVFGC attempts to learn a shared latent space $\mathbf{V}$ from various views. Additionally, we assume that the interrelationship between any pair of vertices is generated by $\mathbf{V}$ and $\mathbf{V}^T$, i.e., $\mathbf{Y}_{ij} = [\mathbf{V}\mathbf{V}^T]_{ij}$+$\gamma_{ij}$, where γ_{ij} denotes the error term. Thus, the interrelationship modeling used by CCPMVFGC is formulated as minimizing the following cost function:

$$O_2 = \left\|\mathbf{Y} - \mathbf{V}\mathbf{V}^T\right\|_F^2. \tag{4}$$

By minimizing O_2 in Eq. (4), the learning of the shared latent space $\mathbf{V}$ is also influenced by edge structure $\mathbf{Y}$ of the AG.

D. Objective Function

Integrating O_1 and O_2 with balancing parameters and constraints, the unified objective function of CCPMVFGC can be defined as follows:

$$\begin{aligned}
&\text{minimize} \\
&O = \sum_i [\left\|\mathbf{F}^i - \mathbf{U}^i\mathbf{V}^T\right\|_F^2 + \left\|\mathbf{X}^i - \mathbf{U}^i\mathbf{U}^{iT}\right\|_F^2] \\
&+ \alpha\left\|\mathbf{Y} - \mathbf{V}\mathbf{V}^T\right\|_F^2 + \lambda\Omega(\mathbf{U}^i), \\
&\text{subject to } \mathbf{V} \geq 0, \mathbf{U}^i \geq 0,
\end{aligned} \tag{5}$$

where α, and λ are balancing parameters used to control the relative significance of interrelationship and feature correlation modeling and the sparseness of $\mathbf{U}^i$ in the learning process. By minimizing the unified objective function in Eq. (5), the optimal cluster preferences for all the n vertices can be learned from multi-view vertex features and graph topology, and the learning process is also regularized by contextual correlations of multi-view vertex features.

The proposed objective function demonstrates that CCPMVFGC is fundamentally different from previous approaches to AG clustering and standard multi-view learning methods. CCPMVFGC is distinguished from previous methods for graph clustering as it uses a generic and flexible objective function to model the clustering problem in AGs. CCPMVFGC attempts to model the interrelationship between pairwise vertices using the latent space $\mathbf{V}$ learned from multi-view features, and such latent space is regularized to consider more impacts brought by the view-wise correlated features. Such a learning scheme is novel to AG clustering. In addition, the model structure of CCPMVFGC is flexible as it can be configured according to the number of vertex feature views. Therefore, CCPMVFGC can be regarded as a generic framework for clustering in AGs. CCPMVFGC is also different from standard approaches to multi-view learning as it emphasizes the role of interrelationship modeling. Last but not the least, unlike the representative multi-view unsupervised learning approaches [49]–[52] proposed for modeling image/vision data, which may consider the artificially constructed edge structure as one of the regular feature views, CCPMVFGC is an effective approach which is specifically designed to discover clusters in real-graphs in which the edge structure is deduced empirically rather than constructed artificially. By generating the edge structure information using latent space learned from multi-view vertex features, the proposed model can better represent the structure of the real-world graph through leveraging the graph topology via multi-view vertex feature learning. In other words, CCPMVFGC is capable of generating an edge between two vertices that share correlated features in all views.

E. Model Optimization

The objective function in Eq. (5) is non-convex. Fortunately, it is convex with respect to $\mathbf{V}$ or $\mathbf{U}^i$ when the other one is fixed. Therefore, we utilize an iterative strategy to optimize $\mathbf{V}$ and $\mathbf{U}^i$ in an alternating manner until convergence. The details of the iterative strategy are described as follows.

1) Updating $\mathbf{V}$*:* Let η_{jk} be the Lagrange multiplier for $\mathbf{V}_{jk} \geq 0$, the Lagrange function for latent variables in $\mathbf{V}$ is given as follows:

$$L(\mathbf{V}, \eta) = O - tr(\eta^T \mathbf{V}). \tag{6}$$

Taking the derivative w.r.t. to $\mathbf{V}$ and according to KKT conditions, we can obtain the following element-wise equation system:

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{V}_{jk}} = 4\alpha[\mathbf{V}\mathbf{V}^T\mathbf{V}]_{jk} - 4\alpha[\mathbf{Y}\mathbf{V}]_{jk} \\ &+2\sum_i [\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i - \mathbf{F}^{iT}\mathbf{U}^i]_{jk} - \eta_{jk} = 0, \\ &\eta_{jk} \cdot \mathbf{V}_{jk} = 0, \eta_{jk} \geq 0. \end{aligned} \tag{7}$$

Solving η_{jk} in the first equation in (7) and substituting it in the second equation, we have the following:

$$\begin{aligned} &[4\alpha\mathbf{V}\mathbf{V}^T\mathbf{V} - 4\alpha\mathbf{Y}\mathbf{V} \\ &+2\sum_i \mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i - \mathbf{F}^{iT}\mathbf{U}^i]_{jk} \cdot \mathbf{V}_{jk} = 0. \end{aligned} \tag{8}$$

Through some mathematical transformations, Eq. (8) can be equivalently rewritten as follows:

$$\begin{aligned} &[4\alpha\mathbf{V}\mathbf{V}^T\mathbf{V} + \sum_i \mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i]^2_{jk} \cdot \mathbf{V}^4_{jk} = [[\sum_i \mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i]^2_{jk} \\ &+8\alpha[\mathbf{V}\mathbf{V}^T\mathbf{V}]_{jk} \cdot [2\alpha\mathbf{Y}\mathbf{V} + \sum_i \mathbf{F}^{iT}\mathbf{U}^i]_{jk}] \cdot \mathbf{V}^4_{jk}. \end{aligned} \tag{9}$$

By solving Eq. (9), the rule for updating $\mathbf{V}$ can be obtained as follows:

$$\begin{aligned} &\mathbf{V}_{jk} \leftarrow \mathbf{V}_{jk} \cdot \frac{\sqrt{\sqrt{\Delta_{jk}} - [\sum_i \mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i]_{jk}}}{\sqrt{4\alpha[\mathbf{V}\mathbf{V}^T\mathbf{V}]_{jk}}}, \\ &\Delta_{jk} = [\sum_i \mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i]^2_{jk} \\ &\quad +[8\alpha\mathbf{V}\mathbf{V}^T\mathbf{V}]_{jk}[2\alpha\mathbf{Y}\mathbf{V} + \sum_i \mathbf{F}^{iT}\mathbf{U}^i]_{jk}. \end{aligned} \tag{10}$$

2) Updating $\mathbf{U}^i$*:* Similarly, we can obtain the rules for updating $\mathbf{U}^i$. Given the corresponding terms containing $\mathbf{U}^i$ in Eq. (5), optimizing the latent variables in $\mathbf{U}^i$ is equivalent to optimizing the following objective function:

$$\begin{aligned} &O(\mathbf{U}^i) = \left\| \widetilde{\mathbf{F}}^i - \mathbf{U}^i\widetilde{\mathbf{V}} \right\|^2_F + \left\| \mathbf{X}^i - \mathbf{U}^i\mathbf{U}^{iT} \right\|^2_F, \\ &\widetilde{\mathbf{F}}^i = [\mathbf{F}^i \ \mathbf{0}_{m\times 1}] \ \widetilde{\mathbf{V}} = [\begin{matrix} \mathbf{V} \\ \sqrt{\lambda}\mathbf{1}_{1\times k} \end{matrix}], \end{aligned} \tag{11}$$

where $\mathbf{0}$ is an m-dimensional column vector with all entries being zero and $\mathbf{1}$ is a k-dimensional row vector with all entries being one. Let σ^i_{jk} be the Lagrange multiplier for $\mathbf{U}^i_{jk} \geq 0$, the Lagrange function for latent variables in $\mathbf{U}^i$ is given as follows:

$$L(\mathbf{U}^i, \sigma^i) = O(\mathbf{U}^i) - tr(\sigma^{iT}\mathbf{U}^i). \tag{12}$$

Taking the derivative w.r.t. to $\mathbf{U}^i$ and according to KKT conditions, we can obtain the following element-wise equation system:

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{U}^i_{jk}} = 4[\mathbf{U}^i\mathbf{U}^{iT}\mathbf{U}^i]_{jk} - 4[\mathbf{X}^i\mathbf{U}^i]_{jk} \\ &+2[\mathbf{U}^i\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}} - \widetilde{\mathbf{F}}^i\widetilde{\mathbf{V}}]_{jk} - \sigma^i_{jk} = 0, \\ &\sigma^i_{jk} \cdot \mathbf{U}^i_{jk} = 0, \sigma^i_{jk} \geq 0. \end{aligned} \tag{13}$$

Solving σ^i_{jk} in the first equation in (13) and substituting it in the second equation, we have:

$$\begin{aligned} &[4\mathbf{U}^i\mathbf{U}^{iT}\mathbf{U}^i + 2\mathbf{U}^i\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}}]_{jk} \cdot \mathbf{U}^i_{jk} = \\ &[4\mathbf{X}^i\mathbf{U}^i + 2\widetilde{\mathbf{F}}^i\widetilde{\mathbf{V}}]_{jk} \cdot \mathbf{U}^i_{jk}. \end{aligned} \tag{14}$$

Through some mathematical transformations, Eq. (14) can be equivalently rewritten as follows:

$$\begin{aligned} &[8\mathbf{U}^i\mathbf{U}^{iT}\mathbf{U}^i + 2\mathbf{U}^i\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}}]^2_{jk} \cdot \mathbf{U}^{i4}_{jk} = [[2\mathbf{U}^i\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}}]^2_{jk} \\ &+[16\mathbf{U}^i\mathbf{U}^{iT}\mathbf{U}^i]_{jk} \cdot [4\mathbf{X}^i\mathbf{U}^i + 2\widetilde{\mathbf{F}}^i\widetilde{\mathbf{V}}]_{jk}] \cdot \mathbf{U}^{i4}_{jk}. \end{aligned} \tag{15}$$

Similarly, we can obtain the updating rule for $\mathbf{U}^i$:

$$\begin{aligned} &\mathbf{U}^i_{jk} \leftarrow \mathbf{U}^i_{jk} \cdot \frac{\sqrt{\sqrt{\Phi_{jk}} - [2\mathbf{U}^i\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}}]_{jk}}}{\sqrt{8[\mathbf{U}^i\mathbf{U}^{iT}\mathbf{U}^i]_{jk}}}, \\ &\Phi_{jk} = [2\mathbf{U}^i\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}}]^2_{jk} \\ &\quad +[16\mathbf{U}^i\mathbf{U}^{iT}\mathbf{U}^i]_{jk} \cdot [4\mathbf{X}^i\mathbf{U}^i + 2\widetilde{\mathbf{F}}^i\widetilde{\mathbf{V}}]_{jk}. \end{aligned} \tag{16}$$

By iteratively updating the latent variables in $\mathbf{V}$ and $\mathbf{U}^i$ according to Eqs. (10) and (16), respectively, the objective function in Eq. (5) can finally converge, which is proved in the following section. The optimization procedure of CCPMVFGC is summarized in Algorithm 1. Note that we optimize

Algorithm 1: Contextual Correlation Preserving Multi-View Featured Graph Clustering (CCPMVFGC)

```
Input: Attributed Graph Data: $\mathbf{Y}$, $\{\mathbf{F}^i\}_{i=1}^d$
Output: Cluster preference for each vertex: $\mathbf{V}$;
        View-wise feature-cluster contribution:
        $\{\mathbf{U}^i\}_{i=1}^d$
Compute $\{\mathbf{X}^i\}_{i=1}^d$ by Eq. (1);
for $i \leftarrow 1 : d$ do
    Initialize $\mathbf{U}^i$;
end
Initialize $\mathbf{V}$;
$t \leftarrow 0$;
while $t < T_{max}$ do
    $t \leftarrow t + 1$;
    for $i \leftarrow 1 : d$ do
        Update $\mathbf{U}^i$ by Eq. (16);
    end
    Update $\mathbf{V}$ by Eq. (10);
    Compute objective value $O^{(t)}$ by Eq. (5);
    if $O^{(t-1)} - O^{(t)} \leq \epsilon$ then
        break;
    end
end
Identify cluster label for each vertex using $\mathbf{V}$;
```

$\mathbf{U}^i$ before $\mathbf{V}$ to improve the efficiency of the optimization process. The view-wise terms used to optimize $\mathbf{V}$, including $\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i$ and $\mathbf{F}^{iT}\mathbf{U}^i$ can be computed immediately after the optimization of each $\mathbf{U}^i$, so that an extra loop to compute these terms can be saved. As the optimization procedure adopted by CCPMVFGC is iterative in nature, changing the order of optimizing $\mathbf{U}^i$ and $\mathbf{V}$ in each iteration will not affect the convergence of the model.

III. Algorithmic Analysis of CCPMVFGC

In this section, we analyze the algorithmic properties of the proposed method, including the algorithm convergence and the computational complexity.

A. Convergence Analysis

To prove the convergence of the algorithm, we make use of one property of an auxiliary function that is also used in the proof of the Expectation-Maximization algorithm [13]. The property of the auxiliary function is described as follows. If there exists an auxiliary function satisfying the conditions that $Q(x,x') \geq F(x)$ and $Q(x,x) = F(x)$, then F is non-increasing under the following updating rule:

$$x^{(t+1)} = \arg\min_x Q(x, x^{(t)}). \tag{17}$$

The equality $F(x^{(t+1)}) = F(x^{(t)})$ holds if and only if x is a local minimum of $Q(x,x')$. By iteratively updating x according to Eq. (17), F will converge to the local minimum $x^{min} = \arg\min_x F(x)$. By defining an appropriate auxiliary function for O, we can demonstrate the convergence of the proposed method.

1) Convergence analysis on $\mathbf{V}$*:* First, we show the convergence of the value of O in Eq. (5) when updating $\mathbf{V}$ according to Eq. (10). Here, we only need to consider the terms in Eq. (5) that are related to the latent variables in $\mathbf{V}$. Thus, we have

$$O_{\mathbf{V}} = \alpha \left\|\mathbf{Y} - \mathbf{V}\mathbf{V}^T\right\|_F^2 + \sum_i \left\|\mathbf{F}^i - \mathbf{U}^i\mathbf{V}^T\right\|_F^2, \tag{18}$$

where $O_{\mathbf{V}}$ denotes the sum of terms in Eq. (5) that are related to the variables in $\mathbf{V}$. Further eliminating the terms that are irrelevant to $\mathbf{V}$, Eq. (18) can be further rewritten as:

$$\begin{aligned} O_{\mathbf{V}} &= \alpha \cdot tr(\mathbf{V}\mathbf{V}^T\mathbf{V}\mathbf{V}^T) - 2\alpha \cdot tr(\mathbf{Y}\mathbf{V}\mathbf{V}^T) \\ &+ \sum_i tr(\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i\mathbf{V}^T) - 2\sum_i tr(\mathbf{F}^{iT}\mathbf{U}^i\mathbf{V}^T). \end{aligned} \tag{19}$$

According to the Lemma 2 in [48], we have

$$\begin{aligned} &- 2\alpha \cdot tr(\mathbf{Y}\mathbf{V}\mathbf{V}^T) = -2\alpha \cdot tr(\mathbf{V}^T\mathbf{Y}\mathbf{V}) \\ &\leq -2\alpha[tr(\mathbf{V}'^T\mathbf{Y}\mathbf{Z}) + tr(\mathbf{Z}^T\mathbf{Y}\mathbf{V}'^T) + tr(\mathbf{V}'^T\mathbf{Y}\mathbf{V}')], \\ &- 2\sum_i tr(\mathbf{F}^{iT}\mathbf{U}^i\mathbf{V}^T) = -2\sum_i tr(\mathbf{V}^T\mathbf{F}^{iT}\mathbf{U}^i) \\ &\leq -2\sum_i tr(\mathbf{Z}^T\mathbf{F}^{iT}\mathbf{U}^i) + tr(\mathbf{V}'^T\mathbf{F}^{iT}\mathbf{U}^i), \\ &\mathbf{Z}_{jk} = \mathbf{V}'_{jk} log\frac{\mathbf{V}_{jk}}{\mathbf{V}'_{jk}}. \end{aligned} \tag{20}$$

It is apparent that two inequalities will become equalities when $\mathbf{V}_{jk} = \mathbf{V}'_{jk}$. According to the Lemmas 6 and 7 in [48], we have:

$$\begin{aligned} &tr(\mathbf{V}\mathbf{U}^{iT}\mathbf{U}^i\mathbf{V}^T) \leq tr(\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i\mathbf{P}^T), \\ &\alpha \cdot tr(\mathbf{V}\mathbf{V}^T\mathbf{V}\mathbf{V}^T) \leq \alpha \cdot tr(\mathbf{V}'\mathbf{V}'^T\mathbf{V}'\mathbf{R}^T), \\ &\mathbf{P}_{jk} = \frac{\mathbf{V}_{jk}^2}{\mathbf{V}'_{jk}}, \mathbf{R}_{jk} = \frac{\mathbf{V}_{jk}^4}{\mathbf{V}'^3_{jk}}. \end{aligned} \tag{21}$$

Similar to Eq. (20), the equality holds when $\mathbf{V}_{jk} = \mathbf{V}'_{jk}$. Based on Eqs. (20) and (21), the auxiliary function used to prove the convergence of Eq. (18) can be defined as:

$$\begin{aligned} Q(\mathbf{V},\mathbf{V}') &= \alpha \cdot tr(\mathbf{V}'\mathbf{V}'^T\mathbf{V}'\mathbf{R}^T) + \sum_i tr(\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i\mathbf{P}^T) \\ &- 2\alpha[tr(\mathbf{V}'^T\mathbf{Y}\mathbf{Z}) + tr(\mathbf{Z}^T\mathbf{Y}\mathbf{V}'^T) + tr(\mathbf{V}'^T\mathbf{Y}\mathbf{V}')] \\ &- 2\sum_i tr(\mathbf{Z}^T\mathbf{F}^{iT}\mathbf{U}^i) + tr(\mathbf{V}'^T\mathbf{F}^{iT}\mathbf{U}^i). \end{aligned} \tag{22}$$

According to Eq. (22), for any element in $\mathbf{V}$, say $\mathbf{V}_{jk}$, the auxiliary function can be written as:

$$\begin{aligned} Q(\mathbf{V}_{jk},\mathbf{V}'_{jk}) &= \alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}\mathbf{R}_{jk} - 4\alpha[\mathbf{Y}\mathbf{V}']_{jk}\mathbf{Z}_{jk} \\ &+ \sum_i [\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}\mathbf{P}_{jk} - 2\sum_i [\mathbf{F}^{iT}\mathbf{U}^i]_{jk}\mathbf{Z}_{jk}. \end{aligned} \tag{23}$$

As Eq. (23) is an auxiliary function w.r.t. $\mathbf{V}_{jk}$, $Q(\mathbf{V}_{jk},\mathbf{V}'_{jk}) \geq O_{\mathbf{V}_{jk}}$, where $O_{\mathbf{V}_{jk}}$ denotes the terms in the objective function related to $\mathbf{V}_{jk}$ and an updating rule can be derived via

Eq. (17). Taking the derivative of Eq. (23) w.r.t. $\mathbf{V}_{jk}$ and letting it equal zero, we have:

$$\begin{aligned}&\mathbf{V}_{jk}^{(t+1)} = \arg\min_{\mathbf{V}_{jk}} Q(\mathbf{V}_{jk}, \mathbf{V}'_{jk})\\ \Rightarrow\ &\frac{\partial Q(\mathbf{V}_{jk}, \mathbf{V}'_{jk})}{\partial \mathbf{V}_{jk}} =\\ &4\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}\frac{\mathbf{V}_{jk}^3}{\mathbf{V}_{jk}'^3} + 2\sum_i[\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}\frac{\mathbf{V}_{jk}}{\mathbf{V}'_{jk}}\\ &-4\alpha[\mathbf{Y}\mathbf{V}']_{jk}\frac{\mathbf{V}'_{jk}}{\mathbf{V}_{jk}} - 2\sum_i[\mathbf{F}^{iT}\mathbf{U}^i]_{jk}\frac{\mathbf{V}'_{jk}}{\mathbf{V}_{jk}} = 0.\end{aligned} \tag{24}$$

Eq. (24) can be equivalently rewritten as:

$$\begin{aligned}&2\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}\mathbf{V}_{jk}^4 - 2\alpha[\mathbf{Y}\mathbf{V}']_{jk}\mathbf{V}_{jk}'^4\\ &+\sum_i[\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}\mathbf{V}_{jk}'^2\mathbf{V}_{jk}^2 - \sum_i[\mathbf{F}^{iT}\mathbf{U}^i]_{jk}\mathbf{V}_{jk}'^4 = 0.\end{aligned} \tag{25}$$

Using the quadratic formula, we have

$$\begin{aligned}&\mathbf{V}_{jk}^2 = \frac{\mathbf{V}_{jk}'^2\sqrt{\Delta_{jk}} - \sum_i[\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}\mathbf{V}_{jk}'^2}{4\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}},\\ &\Delta_{jk} = [\sum_i \mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}^2\\ &+[8\alpha\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}[2\alpha\mathbf{Y}\mathbf{V}' + \sum_i \mathbf{F}^{iT}\mathbf{U}^i]_{jk}.\end{aligned} \tag{26}$$

Thus, we can obtain the following:

$$\mathbf{V}_{jk}^{(t+1)} \leftarrow \mathbf{V}'_{jk} \cdot \frac{\sqrt{\sqrt{\Delta_{jk}} - \sum_i[\mathbf{V}'\mathbf{U}^{iT}\mathbf{U}^i]_{jk}}}{\sqrt{4\alpha[\mathbf{V}'\mathbf{V}'^T\mathbf{V}']_{jk}}}. \tag{27}$$

As shown in Eq. (27), the updating rule toward minimizing $Q(\mathbf{V}_{jk}, \mathbf{V}'_{jk})$ is equivalent to the updating rule in Eq. (10). As $Q(\mathbf{V}_{jk}, \mathbf{V}'_{jk})$ is an auxiliary function of $O_{\mathbf{V}_{jk}}$, $O_{\mathbf{V}_{jk}}$ is non-increasing when $\mathbf{V}_{jk}$ is updated according to Eq. (10).

2) Convergence analysis on $\mathbf{U}^i$*:* Next, we will prove the convergence of Eq. (5) when it is updated using the rule in Eq. (16). Similar to the construction of the auxiliary function for $\mathbf{V}_{jk}$, we define the following auxiliary function for $\mathbf{U}_{jk}^i$:

$$\begin{aligned}&Q(\mathbf{U}_{jk}^i, \mathbf{U}_{jk}^{i\prime}) = [\mathbf{U}^{i\prime}\mathbf{U}^{i\prime T}\mathbf{U}^{i\prime}]_{jk}\mathbf{R}'_{jk} - 4[\mathbf{X}^i\mathbf{U}']_{jk}\mathbf{Z}'_{jk}\\ &+2[\mathbf{U}^{i\prime}\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}}]_{jk}\mathbf{P}'_{jk} - 2[\widetilde{\mathbf{F}}^i\widetilde{\mathbf{V}}]_{jk}\mathbf{Z}'_{jk},\\ &\mathbf{Z}'_{jk} = \mathbf{U}_{jk}^{i\prime} log\frac{\mathbf{U}_{jk}^i}{\mathbf{U}_{jk}^{i\prime}}, \mathbf{P}'_{jk} = \frac{\mathbf{U}_{jk}^{i2}}{\mathbf{U}_{jk}^{i\prime}}, \mathbf{R}'_{jk} = \frac{\mathbf{U}_{jk}^{i4}}{\mathbf{U}_{jk}^{i\prime 3}}.\end{aligned} \tag{28}$$

As the proof that Eq. (28) is an auxiliary function w.r.t. $\mathbf{U}_{jk}^i$ is similar to what Eqs. (20)-(23) demonstrate, we omit the details of the proof. As Eq. (28) is an auxiliary function of $\mathbf{U}_{jk}^i$, we can verify the updating rule w.r.t. $\mathbf{U}_{jk}^i$ by finding the local minimum of Eq. (28):

$$\begin{aligned}&\mathbf{U}_{jk}^{i(t+1)} = \arg\min_{\mathbf{U}_{jk}^i} Q(\mathbf{U}_{jk}^i, \mathbf{U}_{jk}^{i\prime})\\ \Rightarrow\ &\mathbf{U}_{jk}^{i(t+1)} \leftarrow \mathbf{U}_{jk}^{i\prime} \cdot \frac{\sqrt{\sqrt{\Phi_{jk}} - [2\mathbf{U}^{i\prime}\widetilde{\mathbf{V}}^T\widetilde{\mathbf{V}}]_{jk}}}{\sqrt{8[\mathbf{U}^{i\prime}\mathbf{U}^{i\prime T}\mathbf{U}^{i\prime}]_{jk}}}.\end{aligned} \tag{29}$$

As shown in Eq. (29), the updating rule toward minimizing $Q(\mathbf{U}_{jk}^i, \mathbf{U}_{jk}^{i\prime})$ is the equivalent to the updating rule in Eq. (16). As $Q(\mathbf{U}_{jk}^i, \mathbf{U}_{jk}^{i\prime})$ is an auxiliary function of O w.r.t. $\mathbf{U}_{jk}^i$, O is verified to be non-increasing when $\mathbf{U}_{jk}^i$ is updated according to Eq. (16).

Based on the above proof, we have

$$\begin{aligned}&O(\mathbf{V}^{(0)}, \mathbf{U}^{i(0)}) \geq O(\mathbf{V}^{(1)}, \mathbf{U}^{i(0)}) \geq O(\mathbf{V}^{(1)}, \mathbf{U}^{i(1)})\\ &\geq \cdots \geq O(\mathbf{V}^{(T_{max})}, \mathbf{U}^{i(T_{max})}),\end{aligned} \tag{30}$$

where O shows a non-increasing trend in each iteration. Thus, CCPMVFGC is able to converge in a finite number of steps when the latent variables in $\mathbf{U}^i$ and $\mathbf{V}$ are updated according to Eqs. (16) and (10), respectively.

B. Computational Complexity Analysis

Based on the updating rules shown in Eqs. (16) and (10), we can obtain the computational complexity of CCPMVFGC as follows. Given Eq. (10), updating all the latent variables in $\mathbf{V}$ follows the order of $O(n^2k + nk^2 + nm(k^2 + k))$. Given Eq. (16), updating all the latent variables in each $\mathbf{U}^i$ follows the order of $O(2m^i(k^2 + nk) + (m^{i2} + m^i)k^2)$. Therefore, the overall complexity of CCPMVFGC is $O(n^2k + nmk^2)$.

IV. Experiments and Analysis

In this section, we conduct a series of experiments on real-world graph datasets to validate the effectiveness of CCPMVFGC against other classical and state-of-the-art methods.

A. Experimental Setup

1) Baselines for Comparison: We selected ten classical or state-of-the-art approaches as baselines, including NCut [43], AP [17], CNM [37], CoDA [59], k-means [35], CESNA [58], CP-SI [32], CP-PI [32], MISAGA [25], and CoNMF [2].

NCut, AP, CNM, and CoDA are four representative approaches based on graph topology. NCut is a classical spectral-based method for graph clustering, which performs the task via assigning vertices sharing higher structural similarity into the same cluster. AP is able to discover clusters in the graph by maximizing the structural diffusion between each cluster center and other cluster members. CNM is a typical method for community detection which is based on modularity optimization. CoDA is a state-of-the-art algorithm for graph clustering, which performs the task via symmetric probabilistic matrix factorization.

k-means is a classical clustering method, which is able to discover graph clusters using the concatenation of vertex features from multiple views.

CESNA, CP-SI, CP-PI, and MISAGA are four state-of-the-art approaches to AG clustering. CESNA is a probabilistic generative model, which learns a shared latent space as cluster preference for each vertex from edge structure and vertex features of an AG. CP-SI and CP-PI are two effective approaches to AG clustering. They first compute the weight of each edge based on different content propagation models, and then perform Markov clustering in the transition matrix constructed using the edge weights. Therefore, clusters discovered by CP-SI and CP-PI are those whose vertices propagate similar vertex features. MISAGA is an effective method able to learn a shared

TABLE II
STATISTICS OF DATASETS USED IN THE EXPERIMENTS. SOC OR BIO REPRESENTS WHETHER THE DATASET IS A SOCIAL OR BIOLOGICAL GRAPH.

Dataset	Type	n	$\|E\|$	m	d	k
Cal	Soc	769	16656	53	1	10
Ego	Soc	4039	88234	1283	1	191
Twitter	Soc	3687	49881	20905	2	242
Gp-sub	Soc	8725	972899	5913	5	130
Gplus	Soc	107614	3755989	13966	5	463
CLS	Bio	1620	9064	2042	3	200
KRG	Bio	2674	7075	3064	3	200
DIP	Bio	4579	20845	4237	3	200

latent space from the edge structure and pairwise similarity of vertex features.

In addition to the aforementioned graph clustering approaches, we also selected CoNMF as a baseline, to demonstrate the differences between the proposed model and generic approaches to multi-view clustering. Although it does not have functions such as interrelationship modeling or feature correlation modeling, CoNMF is still an effective multi-view clustering approach and is able to learn a shared latent space from multi-view vertex features represented as multiple data matrices. Given its model structure, it is also closely related to clustering methods based on tensor decomposition.

In our experiments, we used the source codes of all the baselines provided by the authors for implementation. Algorithms including AP, CNM, and CESNA do not need any predefined parameter before they run. Therefore, we ran them directly without assigning parameters. NCut, CoDA, k-means, CP-SI, CP-PI, and MISAGA need to pre-determine model parameters before they are executed, so we used the settings recommended in the corresponding papers. For the number of clusters, i.e., k, which have to be predetermined in NCut, CoDA, k-means, CoNMF, CP-SI, CP-PI, and MISAGA, we set it to be equal to the number of ground-truth clusters of the testing dataset. For the proposed CCPMVFGC, we set $\alpha = 10$, $\beta = 0.5$, and $\lambda = 1$. The setting of k in CCPMVFGC is the same as that in the baselines. All of the experiments were performed on a workstation with 4-core 3.4GHz CPU and 16GB RAM and all approaches were executed 10 times to obtain a statistically steady performance.

2) Dataset Description: We used eight real-world graphs with verified ground-truth clusters as testing datasets, five of which are social graphs extracted from different social networking sites while the other three are biological graphs collected from protein-protein interaction networks related to $Saccharomyces\ cerevisiae$. These real-world graphs have different sizes and different numbers of vertex feature views. The detailed descriptions of these eight datasets are as follows.

The datasets $Caltech$ [46], $Ego-facebook$ [36], $Twitter$ [58], $Googleplus-sub$, and $Googleplus$ [36] are five social graphs whose vertices and edges represent the social networking users and the friendship between them, respectively.

$Caltech$ (Cal) is a college social graph extracted from the social networking users in the California Institute of Technology. There are 769 vertices, 16656 edges and 53 vertex features representing the users, the social ties between them, and their profiles, respectively. In the Cal dataset, there are 10 large groups verified according to the college dorm system [46], which can be used as ground-truth clusters to evaluate the clustering performance of different approaches.

$Ego-facebook$ (Ego) is a social graph extracted from facebook.com. This dataset contains 4039 vertices, 88234 edges, and 1283 features which represent the Facebook users, the friendship, and the user profiles, respectively. There are 191 social circles that have been verified as ground-truth clusters.

$Twitter$ is constructed based on a snapshot of the online social networking site twitter.com. In this dataset, there are 3687 vertices and 49881 edges, representing Twitter users and their social ties, respectively. In addition, 20905 features are collected from two sources, including social tags and locations, to characterize 3687 Twitter users.

$Googleplus$ (Gplus) and $Googleplus-sub$ (Gp-sub) are two social graphs constructed based on the users from googleplus.com. Specifically, Gplus-sub is a sub-set of Gplus, which contains 8725 vertices, 972899 edges, and 5913 features, representing the users of googleplus, their social relationship, and their content characterizations, respectively. The vertex information on Gplus is extract from 107614 users on the social networking site. There are 3755989 edges and 13966 features in this dataset. The vertex features in both Gplus and Gplus-sub are collected from five sources: jobs, locations, institutions, universities, and identity information. Thus, there are five views of features in these two datasets. Gplus-sub and Gplus contain 130 and 463 social circles, respectively, which have been verified in previous studies and can be used as ground-truth clusters.

Unlike the five aforementioned graphs, which are constructed based on social data, $Collins$ (CLS) [11], $Krogan$ (KRG) [28], and DIP [55] are three biological graphs used to describe the interactions between proteins related to $Saccharomyces\ cerevisiae$. In these three datasets, vertices, edges and vertex features represent proteins, the protein-protein interactions, and associated Gene-Ontology (GO) terms [3], respectively. Here, GO terms are collected from three different views, i.e., $biological\ processes$, $cellular\ components$, and $molecular\ functions$. Specifically, there are 1620 vertices, 9064 edges, and 2042 vertex features in CLS, 2674 vertices, 7075 edges, and 3064 vertex features in KRG, and 4579 vertices, 20845 edges, and 4237 vertex features in DIP. To evaluate the clustering performance on these three datasets, we used the golden standard of real protein complexes, which are stored in the CYC2008 database [39]. There are 200 laboratory-verified protein complexes that are used as ground-truth clusters. Note that, compared with the five social graphs, the edge structures of CLS, KRG, and DIP are much sparser. Therefore, using different types of graphs can validate the robustness of different approaches. The statistics of these testing datasets are summarized in Table II.

3) Evaluation Metrics: For performance evaluation, we selected two widely used metrics, i.e., the Normalized Mutual

TABLE III
CLUSTERING PERFORMANCE EVALUATED BY NMI (%). THE BEST PERFORMANCE ON EACH DATASET IS HIGHLIGHTED IN BOLD. * INDICATES THAT CCPMVFGC SIGNIFICANTLY OUTPERFORMS THE CORRESPONDING BASELINE ACCORDING TO Z-TEST. ♯ INDICATES AN APPROACH THAT IS EITHER HEURISTIC-SEARCH BASED OR ABLE TO REACH A STEADY STATE, THEREBY NO STANDARD DEVIATION IS COLLECTED.

Datasets Approaches	Cal	Ego	Twitter	Gp-sub	Gplus	CLS	KRG	DIP
NCut	41.113* (±1.027)	53.646* (±0.162)	48.421* (±0.367)	12.915* (±0.588)	– –	74.251* (±0.432)	79.418* (±0.102)	82.687* (±0.114)
AP	38.133* (±0.066)	57.093* (±0.059)	54.264* (±0.105)	40.769 (±0.480)	– –	75.552* (±0.742)	84.418* (±0.340)	84.350* (±0.707)
CNM♯	42.298*	48.266*	35.485*	11.847*	7.554*	83.759*	78.571*	74.796*
CoDA	33.517* (±1.543)	55.505* (±1.066)	60.878* (±0.559)	18.900* (±1.044)	15.126 (±1.022)	88.318 (±0.602)	85.429* (±1.934)	76.314* (±0.707)
k-means	21.064* (±0.102)	40.461* (±0.818)	14.929* (±1.429)	39.735 (±1.393)	– –	86.987 (±0.128)	82.190* (±0.977)	80.757* (±0.106)
CoNMF	30.903* (±0.310)	48.366* (±0.482)	54.025* (±1.462)	25.435* (±0.135)	18.811 (±0.680)	83.499* (±0.496)	86.022* (±0.053)	87.440* (±0.895)
CESNA	39.259* (±1.578)	57.513* (±1.119)	46.588* (±0.155)	21.817* (±0.601)	10.164* (±0.694)	51.348* (±0.045)	74.123* (±0.027)	77.286* (±0.148)
CP-SI♯	21.505*	48.510*	49.752*	21.929*	–	84.698	84.909*	86.598*
CP-PI♯	21.616*	48.981*	48.538*	24.663*	–	81.388*	82.115*	87.499*
MISAGA	29.774* (±0.036)	56.452* (±0.538)	65.329 (±0.226)	21.553* (±1.158)	10.245* (±1.158)	86.344 (±0.304)	86.749* (±0.059)	87.952* (±0.223)
CCPMVFGC	**61.791** (±1.234)	**67.650** (±0.241)	**71.983** (±0.294)	**45.378** (±0.317)	**23.877** (±0.545)	**90.373** (±0.200)	**91.505** (±0.166)	**91.527** (±0.338)
Improvement(%)	46.085	17.626	10.185	11.305	26.931	2.323	5.482	4.065

Information (NMI) and the Accuracy (Acc) [40].

The NMI measures the overall accuracy of the matches between detected clusters and the ground-truth. It is defined as:

$$\begin{aligned} NMI &= \frac{\sum_{C_i, C_j^*} Pr(C_i, C_j^*) log \frac{Pr(C_i, C_j^*)}{Pr(C_i) Pr(C_j^*)}}{\max(H(C), H(C^*))}, \\ H(C) &= -\sum_i Pr(C_i) log Pr(C_i), \\ H(C^*) &= -\sum_j Pr(C_j^*) log Pr(C_j^*), \end{aligned} \tag{31}$$

where Pr(C_i, C_j^*) denotes the probability that the vertices are shared in both the detected cluster i and the ground-truth cluster j, and Pr(C_i) denotes the probability that a vertex belongs to cluster i. According to the definition in Eq. (31), a larger value of NMI indicates a better matching between the detected clusters and the ground-truth.

Unlike NMI, the Acc measures the accuracy of individually detected clusters. It is defined as follows:

$$Acc = \sum_i \frac{|C_i|}{|C|} f(C_i, C^*), \tag{32}$$

where $|C_i|$ denotes the size of detected cluster i, and $f(\cdot, \cdot)$ is defined as the maximum overlap between the detected cluster i and a cluster in the ground-truth database. Thus, Acc evaluates the best matching of each individual cluster. A larger value of Acc indicates a better matching between each detected cluster and the ground-truth.. The larger the Acc values of all clusters detected by an algorithm, the better the performance of an algorithm. The properties of these two evaluation metrics enable them to evaluate the effectiveness of an approach in a complementary manner, so that all the clustering approaches can be evaluated comprehensively.

In addition to the direct comparison between the performance of CCPMVFGC and that of other algorithms in our experiments, we also verified whether CCPMVFGC statistically outperforms other compared baselines by carrying out statistical tests. Specifically, we performed a single-sided z-test to determine whether CCPMVFGC significantly outperforms other baselines at the 95% confidence level.

B. Clustering Performance Comparison

Social community detection in social graphs and functional module identification in biological graphs are two significant applications of graph clustering. In our experiment, we used the aforementioned five social graphs and three biological graphs to test the effectiveness of all of the eleven approaches. As the ground-truth clusters of all the eight testing datasets have been verified in previous studies, we are able to validate the clusters discovered by different approaches against the ground-truth. The experimental results (in terms of NMI and Acc) of all algorithms are summarized in Tables III and IV.

When NMI is considered, CCPMVFGC outperforms all other baselines in both social and biological graph datasets. In six datasets out of eight, the proposed approach significantly outperforms the second best by at least 5%. Specifically, in the Cal dataset, CCPMVFGC outperforms CNM by 46.085%. In the Ego dataset, the proposed approach outperforms CESNA by 17.626%. In the Gp-sub, the improvement against NMI is 11.305%, when CCPMVFGC is compared with the second best approach, i.e., AP. In Gplus, which is the largest dataset in our experiments, we were unable to obtain results of approaches such as NCut, AP, CP-SI, CP-PI, and k-means,

TABLE IV

Clustering Performance Evaluated by Acc (%). The best performance on each dataset is highlighted in bold. * indicates that CCPMVFGC significantly outperforms the corresponding baseline according to z-test. ♯ indicates an approach that is either heuristic-search based or able to reach a steady state, thereby no standard deviation is collected.

Datasets Approaches	Cal	Ego	Twitter	Gp-sub	Gplus	CLS	KRG	DIP
NCut	37.451* (±1.756)	44.689* (±0.520)	42.121* (±0.122)	26.705* (±0.183)	– –	22.037* (±0.524)	7.949* (±0.037)	4.695* (±0.098)
AP	45.774* (±0.065)	41.619* (±0.025)	56.453* (±0.488)	57.996* (±0.946)	– –	27.284* (±0.370)	18.699* (±0.748)	12.929* (±0.001)
CNM$^\sharp$	30.949*	37.979*	40.136*	28.410*	9.603*	33.025*	12.042*	3.451*
CoDA	37.824* (±0.839)	52.091 (±1.597)	66.537 (±2.545)	34.735* (±2.221)	17.705* (±4.109)	35.193* (±0.596)	**26.000** (±2.947)	8.362* (±0.193)
k-means	14.954* (±0.130)	29.116* (±0.248)	28.343* (±1.180)	16.252* (±0.521)	– –	33.398* (±0.154)	12.789* (±0.374)	8.167* (±0.109)
CoNMF	26.268* (±0.325)	38.945* (±0.161)	50.344* (±1.497)	23.035* (±0.431)	48.207 (±0.596)	31.084* (±0.621)	17.166* (±0.306)	10.465* (±0.442)
CESNA	38.429* (±0.794)	46.124* (±1.118)	51.340* (±0.350)	24.038* (±0.025)	26.783* (±0.441)	4.539* (±0.007)	5.508* (±0.184)	2.600* (±0.035)
CP-SI$^\sharp$	15.735*	37.905*	51.587*	25.616*	–	31.467*	15.299*	9.085*
CP-PI$^\sharp$	14.434*	36.915*	51.179*	23.381*	–	26.319*	4.833*	9.528*
MISAGA	25.618* (±0.650)	45.159* (±0.718)	68.619 (±0.203)	53.009* (±0.550)	37.271 (±1.299)	35.419* (±0.318)	18.102* (±0.001)	10.340* (±0.175)
CCPMVFGC	**57.659** (±1.470)	**53.830** (±0.451)	**71.274** (±0.364)	**69.714** (±0.320)	**63.742** (±0.550)	**41.086** (±0.308)	24.488 (±0.001)	**15.642** (±0.313)
Improvement(%)	25.965	3.338	3.869	20.205	32.226	15.999	–	20.984

as these were not designed to run and perform in large-scale datasets. In this largest dataset, CCPMVFGC outperforms CoNMF by 26.931%.

When using the Acc metric, CCPMVFGC still performs robustly. It can outperform all of the baselines in seven datasets, except in KRG, where CoDA performs the best. In five datasets, the proposed model outperforms other baselines by more than 5%. Specifically, when detecting social communities in Cal and Gp-sub, CCPMVFGC outperforms AP by 25.965% and 20.205%, respectively. In Gplus, CCPMVFGC outperforms CoNMF by 32.226%. When discovering functional module in datasets CLS and DIP, CCPMVFGC outperforms MISAGA by 15.999% and AP by 20.984%, respectively.

Additionally, we carried out a z-test to determine whether CCPMVFGC significantly outperforms other baselines at the 95% confidence level. In Tables III and IV, an asterisk is given alongside the result if the proposed CCPMVFGC significantly outperforms the corresponding baseline on a specific dataset. As shown in the tables, CCPMVFGC has a statistically significant improvement when compared with other baselines in most of the testing datasets. For example, when NMI is considered, CCPMVFGC significantly outperforms all the baselines in the Cal, Ego, KRG and DIP datasets. It also significantly outperforms most of the baselines in the remaining four datasets. When evaluating using Acc, CCPMVFGC significantly outperforms all the baselines in the Cal, Gp-sub,CLS, and DIP datasets and also achieves a significant improvement when compared with most of the baselines in the remaining four datasets: Ego, Twitter, Gplus, and KRG.

From the experimental results in terms of NMI, Acc and the z-test, we can observe that CCPMVFGC is effective in AG clustering. The multi-view learning scheme used by the model allows it to capture view-wise impacts on the cluster preference for each vertex. In addition, CCPMVFGC benefits from the modeling of view-wise correlations of vertex features, as it can regularize the model to infer similar cluster preferences for vertices sharing highly correlated features from all of the views.

C. Parameter Sensitivity Analysis

We investigate the parameter sensitivity in this section to understand how the variation of α and λ, leading to different relative weights between interrelationship and feature modeling, and the sparseness of each $\mathbf{U}^i$, respectively, will impact the clustering performance. Specifically, we set $\alpha = [0.1, 0.5, 1, 5, 10, 20, 50, 100]$ and $\lambda = [0.001, 0.01, 0.1, 1, 5, 10, 50, 100]$, and run CCPMVFGC on all datasets. We then evaluate the clusters discovered by CCPMVFGC using different settings of α and λ in terms of NMI and Acc. We present the results obtained from dataset Cal in Fig. 2, as an example, to show how the clustering performance of CCPMVFGC is impacted by different settings of model parameters. As depicted in Fig. 2, both NMI and Acc perform robustly when the value of α is relatively large, e.g., $\alpha \geq 5$. While, varying the value of λ does not impact the clustering performance much. According to the results of sensitivity analysis shown in Fig. 2, the performance of the proposed CCPMVFGC is relatively robust under a wide range of parameter combinations. For simplicity, we set $\alpha = 10$ and $\lambda = 1$ in all our experiments.

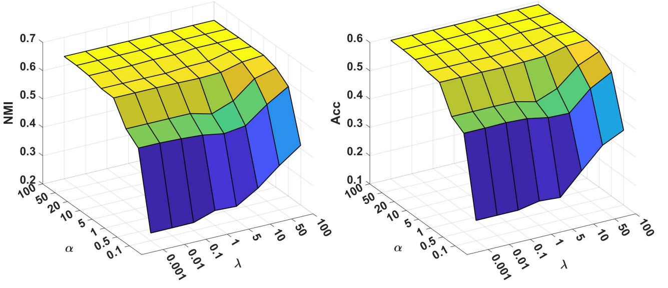

Fig. 2. Sensitivity analysis of CCPMVFGC with respect to the parameter α and λ in Cal dataset

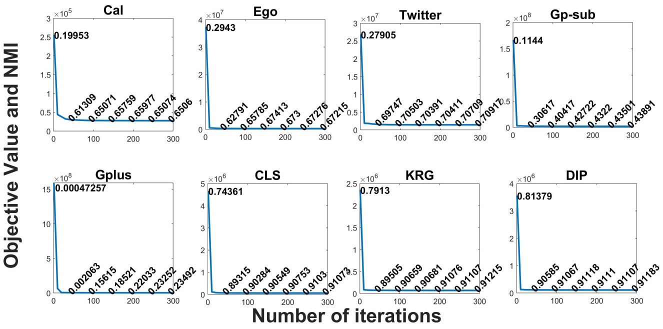

Fig. 3. Model convergence on testing datasets.

D. Model Convergence in Testing Datasets

In addition to the theoretical analysis on model convergence, we also investigated the convergence speed of CCPMVFGC on real graph datasets. Specifically, we recorded the objective function value of CCPMVFGC for the first 300 iterations on all eight datasets. In addition, we recorded the performance of CCPMVFGC (in terms of NMI) every 50 iterations to investigate whether the clustering performance also converges along with the convergence of model optimization. As depicted in Fig. 3, the objective function of CCPMVFGC converges to a stable value in less than 200 iterations on all datasets, showcasing CCPMVFGC's capacity to attain the desired clustering results efficiently. Along with the model convergence, the clustering performance of CCPMVFGC also becomes stable, which demonstrates an approximate synchronization between model convergence and learning performance.

E. Running Time on Model Optimization

In addition to the previous experiment on model convergence, we further tested the computational efficiency of the proposed approach by recording the overall running time on model optimization on all the datasets, and compared it with that obtained by other representative graph clustering approaches based on iterative optimization, including CoDA, CESNA, and MISAGA. The results are shown in Fig. 4. We can observe that CCPMVFGC is comparable to CoDA, which is an efficient approach for performing clustering on massive graphs. In large datasets, such as the Gp-sub and Gplus, the overall optimization time of CCPMVFGC is less than that of all the other baselines. Through investigating the complexity and convergence speed of CCPMVFGC and different baselines, we may gain insights of the proposed model's scalability as compared to other approaches. In terms of computational complexity, CoDA, CESNA, and MISAGA follow the order of O(3n^2k), O(3n^2k+3nmk), and O(4n^2k+4nk^2+4n^2k^2), respectively, whereas the complexity of CCPMVFGC is O(n^2k+nmk^2), as mentioned in Section III.B. It is obvious that the complexity of the proposed model is jointly dominated by n and m. While the computational time of all the compared baselines are dominantly determined by n. Thus, it is reasonable that CCPMVFGC costs slightly more time on model optimization in datasets such as Twitter, CLS, and DIP, whose m is close to, or larger than n. On the contrary, CCPMVFGC is very efficient on large datasets, such as Gp-sub and Gplus, due to the following reasons. Firstly, the value of m is much smaller than n in those massive datasets, thereby the complexity of all the approaches is dominantly determined by n. For example, as shown in Table II, there are 107,614 vertices (n) in Gplus, which is about 9 times larger than that of the vertex features (m). Secondly, CCPMVFGC is able to converge in fewer iterations, while its optimization time in each iteration is approximately equal to, or just slightly longer than the baselines' running time. For instance, the total optimization time used by CCPMVFGC with the Gplus dataset is 4005.9 seconds, which is 2319.1 seconds faster than that of CoDA. In summary, the proposed model's capacity for fast convergence and the acceptable running time in each iteration together make CCPMVFGC an efficient approach for discovering clusters in massive graph data.

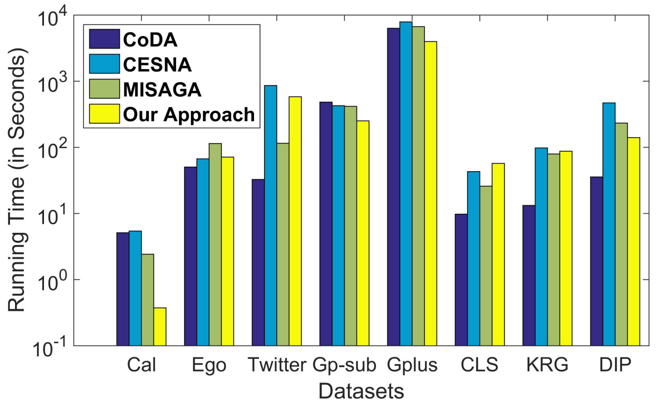

Fig. 4. Scalability comparisons among iteration based methods

V. Conclusions

In this paper, we propose a novel model referred to as Contextual Correlation Preserving Multi-View Featured Graph Clustering (CCPMVFGC) for discovering clusters in attributed graphs. Unlike previous approaches, CCPMVFGC uses an effective multi-view learning scheme that allows for modeling the structural relationship between pairwise vertices using the latent space learned from multi-view vertex features, so that the latent cluster preference is jointly learned by both graph topology and view-wise vertex features. In addition, CCPMVFGC utilizes the latent feature cluster preference to model the contextual correlations between pairwise vertex features. Therefore, the learned cluster features and cluster members are more contextually correlated, which enables the proposed model to discover more meaningful clusters in the attributed graph. CCPMVFGC was tested on a number of real-world graph datasets and compared with both classical and state-of-the-art approaches. The experimental results indicate that CCPMVFGC is capable of revealing clusters in graphs constructed by different types of real-world data with high accuracy, and also in a computationally efficient manner. In the future, we will further improve the efficiency of CCPMVFGC by developing a model based on parallel computing, and improve the effectiveness of CCPMVFGC by allowing the model to consider cross-view feature correlations.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Journal of Machine Learning Research*, vol. 9, no. Sep, pp. 1981–2014, 2008.

[2] Z. Akata, C. Thurau, and C. Bauckhage, "Non-negative matrix factorization in multimodality data for segmentation and label prediction," in *16th Computer vision winter workshop*, 2011.

[3] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, p. 25, 2000.

[4] R. Balasubramanyan and W. W. Cohen, "Block-lda: Jointly modeling entity-annotated text and entity-entity links," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 450–461.

[5] Q. Bao, W. K. Cheung, Y. Zhang, and J. Liu, "A component-based diffusion model with structural diversity for social networks." *IEEE Trans. Cybernetics*, vol. 47, no. 4, pp. 1078–1089, 2017.

[6] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[7] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[8] Z. Bu, H.-J. Li, J. Cao, Z. Wang, and G. Gao, "Dynamic cluster formation game for attributed graph clustering," *IEEE Transactions on Cybernetics*, no. 99, pp. 1–14, 2017.

[9] J. Chang and D. Blei, "Relational topic models for document networks," in *Artificial Intelligence and Statistics*, 2009, pp. 81–88.

[10] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[11] S. R. Collins, P. Kemmeren, X.-C. Zhao, J. F. Greenblatt, F. Spencer, F. C. Holstege, J. S. Weissman, and N. J. Krogan, "Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae," *Molecular & Cellular Proteomics*, vol. 6, no. 3, pp. 439–450, 2007.

[12] S. Coussi-Korbel and D. M. Fragaszy, "On the relation between social dynamics and social learning," *Animal behaviour*, vol. 50, no. 6, pp. 1441–1453, 1995.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[14] A. G. Duran and M. Niepert, "Learning graph representations with embedding propagation," in *Advances in neural information processing systems*, 2017, pp. 5119–5130.

[15] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[16] M. Frank, A. P. Streich, D. Basin, and J. M. Buhmann, "Multi-assignment clustering for boolean data," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 459–489, 2012.

[17] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[18] Z. Gao, D. Wang, Y. Xue, G. Xu, H. Zhang, and Y. Wang, "3d object recognition based on pairwise multi-view convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 56, pp. 305–315, 2018.

[19] Z. Gao, H. Zhang, G. Xu, Y. Xue, and A. G. Hauptmann, "Multi-view discriminative and structured dictionary learning with group sparsity for human action recognition," *Signal Processing*, vol. 112, pp. 83–97, 2015.

[20] J. O. Garcia, A. Ashourvan, S. Muldoon, J. M. Vettel, and D. S. Bassett, "Applications of community detection techniques to brain graphs: Algorithmic considerations and implications for neural function," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 846–867, 2018.

[21] T. Guo, J. Wu, X. Zhu, and C. Zhang, "Combining structured node content and topology information for networked graph clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 11, no. 3, p. 29, 2017.

[22] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1024–1034.

[23] K. He, Y. Li, S. Soundarajan, and J. E. Hopcroft, "Hidden community detection in social networks," *Information Sciences*, vol. 425, pp. 92–106, 2018.

[24] T. He and K. C. Chan, "Discovering fuzzy structural patterns for graph analytics," *IEEE Transactions on Fuzzy Systems*, 2018.

[25] T. He and K. C. Chan, "Misaga: An algorithm for mining interesting subgraphs in attributed graphs," *IEEE transactions on cybernetics*, vol. 48, no. 5, pp. 1369–1382, 2018.

[26] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[27] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[28] N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis *et al.*, "Global landscape of protein complexes in the yeast saccharomyces cerevisiae," *Nature*, vol. 440, no. 7084, p. 637, 2006.

[29] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 631–640.

[30] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in neural information processing systems*, 2014, pp. 2177–2185.

[31] Z. Li, J. Liu, and K. Wu, "A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks," *IEEE transactions on cybernetics*, vol. 48, no. 7, pp. 1963–1976, 2018.

[32] L. Liu, L. Xu, Z. Wangy, and E. Chen, "Community detection based on structure and content: a content propagation perspective," in *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 271–280.

[33] Y. Liu, Z. Gu, T. H. Ko, and J. Liu, "Identifying key opinion leaders in social media via modality-consistent harmonized discriminant embedding," *IEEE Transactions on Cybernetics*, pp. 1–12, 2018.

[34] Y. Liu, T. H. Ko, and Z. Gu, "Who is the mr. right for your brand? – discovering brand key assets via multi-modal asset-aware projection," in *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 1113–1116.

[35] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[36] J. Mcauley and J. Leskovec, "Discovering social circles in ego networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p. 4, 2014.

[37] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[38] C. Pizzuti and A. Socievole, "Multiobjective optimization and local merge for clustering attributed graphs," *IEEE transactions on cybernetics*, 2019.

[39] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak, "Up-to-date catalogues of yeast protein complexes," *Nucleic acids research*, vol. 37, no. 3, pp. 825–831, 2008.

[40] G.-J. Qi, C. C. Aggarwal, and T. Huang, "Community detection with edge content in social media networks," in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE, 2012, pp. 534–545.

[41] M. Qiao, J. Yu, W. Bian, Q. Li, and D. Tao, "Adapting stochastic block models to power-law degree distributions," *IEEE Transactions on Cybernetics*, 2018.

[42] X. Shen, S. Pan, W. Liu, Y.-S. Ong, and Q.-S. Sun, "Discrete network embedding," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 3549–3555.

[43] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[44] Y. Sun, J. Han, J. Gao, and Y. Yu, “itopicmodel: Information network-integrated topic modeling,” in *Data Mining, 2009. ICDM’09. Ninth IEEE International Conference on*. IEEE, 2009, pp. 493–502.

[45] Y. Sun, Y. Yu, and J. Han, “Ranking-based clustering of heterogeneous information networks with star network schema,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 797–806.

[46] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter, “Comparing community structure to characteristics in online collegiate social networks,” *SIAM review*, vol. 53, no. 3, pp. 526–543, 2011.

[47] N. Veldt, D. F. Gleich, and A. Wirth, “A correlation clustering framework for community detection,” in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 439–448.

[48] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, “Community discovery using nonnegative matrix factorization,” *Data Mining and Knowledge Discovery*, vol. 22, no. 3, pp. 493–521, 2011.

[49] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, “Robust subspace clustering for multi-view data by exploiting correlation consensus,” *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3939–3949, 2015.

[50] Y. Wang and L. Wu, “Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering,” *Neural Networks*, vol. 103, pp. 1–8, 2018.

[51] Y. Wang, L. Wu, X. Lin, and J. Gao, “Multiview spectral clustering via structured low-rank matrix factorization,” *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–11, 2018.

[52] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan, “Iterative views agreement: an iterative low-rank based structured optimization method to multi-view spectral clustering,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 2153–2159.

[53] Y. Wang, W. Zhang, L. Wu, X. Lin, and X. Zhao, “Unsupervised metric fusion over multiview data by graph random walk-based cross-view diffusion,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 1, pp. 57–70, 2017.

[54] L. Wu, Y. Wang, and L. Shao, “Cycle-consistent deep generative hashing for cross-modal retrieval,” *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1602–1612, 2019.

[55] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg, “Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions,” *Nucleic acids research*, vol. 30, no. 1, pp. 303–305, 2002.

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

[57] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, “Gbagc: a general bayesian framework for attributed graph clustering,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 9, no. 1, p. 5, 2014.

[58] J. Yang, J. McAuley, and J. Leskovec, “Community detection in networks with node attributes,” in *Data Mining (ICDM), 2013 IEEE 13th international conference on*. IEEE, 2013, pp. 1151–1156.

[59] J. Yang, J. McAuley, and J. Leskovec, “Detecting cohesive and 2-mode communities indirected and undirected networks,” in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 323–332.

[60] L. Yang, X. Cao, D. Jin, X. Wang, and D. Meng, “A unified semi-supervised community detection framework using latent space graph regularization,” *IEEE transactions on cybernetics*, vol. 45, no. 11, pp. 2585–2598, 2015.

[61] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, “Deep multi-view spatial-temporal network for taxi demand prediction,” in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018, pp. 2588–2595.

[62] Y. Zhou, H. Cheng, and J. X. Yu, “Graph clustering based on structural/attribute similarities,” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 718–729, 2009.

[63] Y. Zhou, H. Cheng, and J. X. Yu, “Clustering large attributed graphs: An efficient incremental approach,” in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 689–698.