Effective Clustering for Large Multi-Relational Graphs

XIAOYANG LIN, Hong Kong Baptist University, Hong Kong SAR, China
RUNHAO JIANG, Hong Kong Baptist University, Hong Kong SAR, China
RENCHI YANG*, Hong Kong Baptist University, Hong Kong SAR, China

Multi-relational graphs (MRGs) are an expressive data structure for modeling diverse interactions/relations among real objects (i.e., nodes), which pervade extensive applications and scenarios. Given an MRG $\mathcal{G}$ with N nodes, partitioning the node set therein into K disjoint clusters (referred to as MRGC) is a fundamental task in analyzing MRGs, which has garnered considerable attention. However, the majority of existing solutions towards MRGC either yield severely compromised result quality by ineffective fusion of heterogeneous graph structures and attributes, or struggle to cope with sizable MRGs with millions of nodes and billions of edges due to the adoption of sophisticated and costly deep learning models.

In this paper, we present DEMM and DEMM+, two effective MRGC approaches to address the aforementioned limitations. Specifically, our algorithms are built on novel two-stage optimization objectives, where the former seeks to derive high-caliber node feature vectors by optimizing the *multi-relational Dirichlet energy* specialized for MRGs, while the latter minimizes the *Dirichlet energy* of clustering results over the node affinity graph. In particular, DEMM+ achieves significantly higher scalability and efficiency over our based method DEMM through a suite of well-thought-out optimizations. Key technical contributions include (i) a highly efficient approximation solver for constructing node feature vectors, and (ii) a judicious and theoretically-grounded problem transformation together with carefully-crafted techniques that enable the linear-time clustering without explicitly materializing the $N \times N$ dense affinity matrix. Further, we extend DEMM+ to handle attribute-less MRGs through non-trivial adaptations. Extensive experiments, comparing DEMM+ against 20 baselines over 11 real MRGs, exhibit that DEMM+ is consistently superior in terms of clustering quality measured against ground-truth labels, while often being remarkably faster.

CCS Concepts: • **Computing methodologies → Cluster analysis**; **Spectral methods**; • **Information systems → Clustering**.

Additional Key Words and Phrases: multi-relational graphs, clustering, Dirichlet energy

ACM Reference Format:
Xiaoyang Lin, Runhao Jiang, and Renchi Yang. 2025. Effective Clustering for Large Multi-Relational Graphs. *Proc. ACM Manag. Data* 3, 6 (SIGMOD), Article 319 (December 2025), 28 pages. https://doi.org/10.1145/3769784

1 Introduction

Multi-relational graphs (MRGs) are data structures composed of nodes interconnected via multiple types of relations, which excel in modeling and capturing complex relations and associations among real-world entities. Practical MRGs include social networks, whose users are connected via friendships and varied interactive activities, biological graphs where biological entities (proteins or genes) are associated by interactions, regulatory relationships, or metabolic pathways, as well as financial networks that encompass diverse edges, such as transactions, ownerships, and contractual

*Corresponding Author

Authors' Contact Information: Xiaoyang Lin, Hong Kong Baptist University, Hong Kong SAR, China, csxylin@comp.hkbu.edu.hk; Runhao Jiang, Hong Kong Baptist University, Hong Kong SAR, China, csrhjiang@comp.hkbu.edu.hk; Renchi Yang, Hong Kong Baptist University, Hong Kong SAR, China, renchi@hkbu.edu.hk.

ACM 2836-6573/2025/12-ART319
https://doi.org/10.1145/3769784

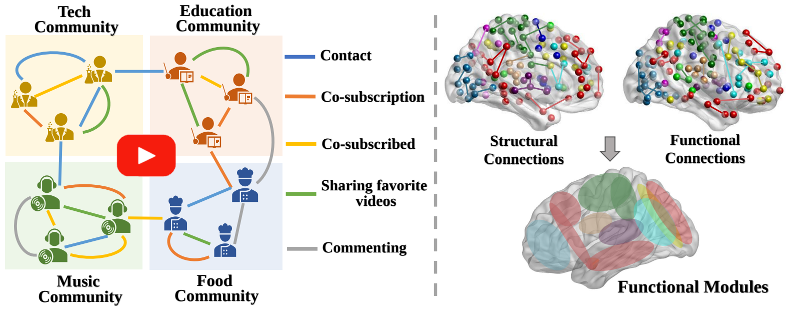


Fig. 1. Real application examples of MRGC.

relationships. Due to the omnipresence of such multi-relational data structures, MRGs find broad applications across various domains, including recommendation systems [20, 37], biomedicine [38, 101], financial risk control [78, 85], academic network mining [17, 98], social network analysis [26, 46], etc.

As a fundamental analytical task, the goal of *multi-relational graph clustering* (MRGC) is to divide the MRG $\mathcal{G}$ into K disjoint groups of nodes that are internally tightly-knit and similar, where the number K of clusters is specified a priori. Two real-world application examples (depicted in Fig 1) of MRGC are as follows:

- **Detecting Social Communities:** On the video sharing website YouTube, as shown in Fig. 1, active users can connect via contact, co-subscription, co-subscribed, sharing favorite videos, and commenting, which form a multiple relational graph (MRG). Through MRGC, we can extract high-quality communities of users sharing similar interests by integrating such heterogeneous interactions/relations [79], thereby facilitating video/YouTuber recommendations and advertising.
- **Neuroscience:** In brain networks, there are structural (e.g., axonal pathways) and functional (e.g., correlations in activity) connections among brain regions (e.g., neurons or cortical areas). The clustering over such multi-relational structures can help identify functional modules and offer valuable insights into brain structures and functions [2, 13].

Despite being superior in practical applications, compared to traditional graph clustering, MRGC poses unique challenges in fusing rich structures underlying heterogeneous relations, as well as exploiting nodal attributes that are widely present in real MRGs.

A straightforward treatment for MRGC is to simply convert the MRG $\mathcal{G}$ into a single-relational graph $\overline{\mathcal{G}}$ through an equal weighting of multiple-typed relations therein, followed by applying *attributed graph clustering* techniques [52, 95] over $\overline{\mathcal{G}}$. This paradigm overlooks the specific nuances and importance of different relation types, engendering biased results and subpar clustering quality. For instance, on social networks, treating relationships, including friendships, family ties, and professional connections equally will obscure the distinction between close family members and distant acquaintances.

Over the past few years, there has been a surge of interest in designing approaches specially catered for MRGC [27, 45, 57, 58] (detailed in Section 6). The majority of them can be categorized into two groups: *Multi-Relational Structure* (MRS)-based and *Multi-View Embedding* (MVE)-based methods. Specifically, as depicted in Figure 2, the MRS-based methodology [45, 47, 60] focuses on automatically adjusting weights for the integration of graph structures $\{\boldsymbol{A}^{(r)}\}$ under heterogeneous relation types in MRGs, before incorporating node attributes $\boldsymbol{X}$ for subsequent clustering. However, this category of methods primarily hinges on graph structures for weight adjustment, which disregards or underexploits the attribute information. Such an oversight results in inaccurate weights and severe misalignment between graph structures and node attributes. In contrast, the

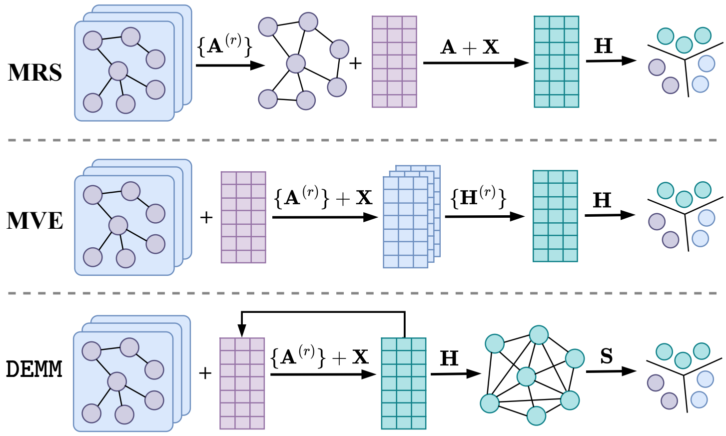


Fig. 2. Workflows of existing MRGC methods and DEMM.

MVE-based models [48, 54, 59, 62, 64, 69, 70, 91] reverse the above two steps (see Figure 2), where the former step turns to encoding attributes $\boldsymbol{X}$ on each single-relational graph $\boldsymbol{A}^{(r)}$ into node feature vectors $\boldsymbol{H}^{(r)}$ severally, whilst the latter step attends to unifying these multi-typed feature vectors $\{\boldsymbol{H}^{(r)}\}$ into the final representations $\boldsymbol{H}$ for node clustering. Although this post-fusion scheme enjoys better result effectiveness, it fails to adequately capture the structural consistencies, disparities, and complementaries of varied types of relations [110].

In summary, extant MRGC studies still have flaws in reconciling multiplex relations and fusing information from heterogeneous structures and attributes, and thus, incur sub-optimal performance. On top of that, most solutions rely on sophisticated matrix solvers or deep learning models that entail substantial memory and compute-intensive operations, which are rather expensive for even medium-sized MRGs.

To overcome the deficiencies of existing methods, we propose DEMM and DEMM+ that achieve superb performance for MRGC over multiple real MRG datasets, through the optimization of our novel two-stage objective functions formulated based on the *Dirichlet energy* (DE) [108] in a principled way. As overviewed in Figure 2, distinct from MRS- and MVE-based approaches, DEMM follows a two-stage pipeline, in which the first stage iteratively refines the node feature vectors $\boldsymbol{H}$ by injecting information from node attributes $\boldsymbol{X}$ and multiplex graph structures $\{\boldsymbol{A}^{(r)}\}$, while the second phase constructs an affinity graph $\boldsymbol{S}$ from $\boldsymbol{H}$ and derives clusters therefrom. More concretely, in the first stage, the feature vectors $\boldsymbol{H}$ and weights for integrating $\{\boldsymbol{A}^{(r)}\}$ are alternatively updated towards optimizing the notion of *multi-relational Dirichlet energy* (MRDE) and ancillary terms, which is a new extension of the DE to MRGs dedicated to enforcing features of adjacent nodes of important relation types to be close. In the same vein, DEMM obtains clusters by minimizing their DE on $\boldsymbol{S}$ such that cluster assignments of nodes with high affinity in $\boldsymbol{S}$ are similar. Unfortunately, DEMM suffers from a quadratic complexity for the computation of $\boldsymbol{H}$ and materialization of $\boldsymbol{S}$, rendering it incompetent for large MRGs.

To this end, we upgrade DEMM to a linear-time method DEMM+, which obtains high efficiency without degrading result utility, via a series of novel algorithmic designs, optimization tricks, and theoretical analyses. Under the hood, DEMM+ includes a carefully-designed approximate solver FAAO for alternative updating of feature vectors $\boldsymbol{H}$ and fusion weights, by uncovering computation bottlenecks and capitalizing on their mathematical properties for fast estimation. In addition, through theoretically-grounded problem transformations along with our SSKC algorithm empowered by mathematical apparatus *random Fourier features* [66] and *Sinkhorn-Knopp normalization* [73], DEMM+

Table 1. Frequently used symbols.

Symbol	Description
$\mathcal{V}, \mathcal{E}^{(r)}$	The node set and edge set of r^{th} relation type.
$N, M^{(r)}, M$	The numbers of nodes, edges in $\mathcal{E}^{(r)}$, and all the edges.
R, K	The numbers of relation types and desired clusters.
D, d	The dimensions of the input attribute and feature vectors.
$\boldsymbol{X}, \boldsymbol{H}$	Initial and target feature vectors of nodes.
$\boldsymbol{D}^{(r)}$	The diagonal degree matrix of $\mathcal{E}^{(r)}$.
$\boldsymbol{A}^{(r)}, \hat{\boldsymbol{A}}^{(r)}$	The adjacency matrix of $\mathcal{E}^{(r)}$ and its normalized version.
ω_r	The importance weight for r^{th} relation type.
$\boldsymbol{Y}, \boldsymbol{S}$	The NCI and affinity matrix defined in Eq. (1) and Eq. (7).
$\mathcal{D}(\boldsymbol{H}, \boldsymbol{A}^{(r)})$	The DE of $\boldsymbol{H}$ on $\boldsymbol{A}^{(r)}$ defined in Eq. (2).
α, β	The coefficients for terms $\mathcal{L}_{\text{MRDE}}$ and $\mathcal{L}_{\text{reg}}$ in Eq. (4).
L, m	The number of hops and sketching dimension in FAAO.

judiciously eliminates the need to materialize a quadratic-sized affinity graph and its rear-mounted arduous eigendecomposition in DEMM. Furthermore, we enable DEMM+ over *attribute-less* MRGs that are under-explored in previous works with an additional orthogonality constraint. Our empirical studies evaluating DEMM+ against 20 competitors on 11 real MRG datasets demonstrate that DEMM+ consistently and conspicuously outperforms the state-of-the-art solutions for MRGC in terms of clustering quality at a fraction of their computational expenses.

The contributions of this paper can be summarized as follows:

- Conceptually, we introduce the new notion of MRDE on MRGs and formulate the MRGC task as a two-stage optimization problem based on the MRDE and DE.
- Methodologically, we develop a brute-force algorithm DEMM to solve the above objectives for effective MRGC, and a computationally tractable solver DEMM+ for practical scalability with non-trivial theories and techniques FAAO and SSKC. DEMM+ is further extended as DEMM-NA to attribute-less MRGs.
- Empirically, we conduct extensive experiments on 9 real datasets of various sizes to validate the effectiveness and efficiency of proposed methods.

2 Problem Formulation

In this section, we set up the necessary preliminaries and provide a formalization of the MRGC problem.

2.1 Symbol and Terminology

Matrix Notation. Throughout this paper, sets are denoted by calligraphic letters, e.g., $\mathcal{V}$. Matrices (resp. vectors) are written in bold uppercase (resp. lowercase) letters, e.g., $\boldsymbol{M}$ (resp. $\mathbf{x}$). We use $\boldsymbol{M}_i$ and $\boldsymbol{M}_{\cdot,i}$ to represent the i^{th} row and column of $\boldsymbol{M}$, respectively. $\|\boldsymbol{M}\|_F$ denotes the Frobenius norm of matrix $\boldsymbol{M}$ and $\mathrm{nnz}(\boldsymbol{M})$ is the number of non-zero entries in $\boldsymbol{M}$. A matrix $\boldsymbol{M}$ is said to be row-normalized (resp. column-normalized) if each i^{th} row (resp. column) is L_2 normalized, i.e., $\|\boldsymbol{M}_i\|_2$=1 (resp. $\|\boldsymbol{M}_{\cdot,i}\|_2 = 1$). For ease of exposition, we say $\boldsymbol{M} \in \mathbb{N}_{\text{row}}$ if $\boldsymbol{M}$ is row-normalized. By "the first K eigenvectors", we refer to the eigenvectors corresponding to the K largest eigenvalues of a matrix.

Graph Nomenclature. A *multi-relational graph* (MRG) is defined as $\mathcal{G} = (\mathcal{V}, \{\mathcal{E}^{(r)}\}_{r=1}^{R})$, where $\mathcal{V}$ denotes the set of N distinct nodes and $\mathcal{E}^{(r)}$ contains a set of $M^{(r)}$ edges (or relations) between nodes in $\mathcal{V}$ in the r^{th} ($1 \le r \le R$) type of relation. The total number of edges in $\mathcal{G}$ is denoted by $M = \sum_{r=1}^{R} M^{(r)}$. For each edge $(v_i, v_j) \in \mathcal{E}^{(r)}$ connecting nodes v_i and v_j, we say v_i and v_j are neighbors to each other under r^{th} relation type. The degree of v_i (i.e., the neighbors of v_i) in $\mathcal{E}^{(r)}$

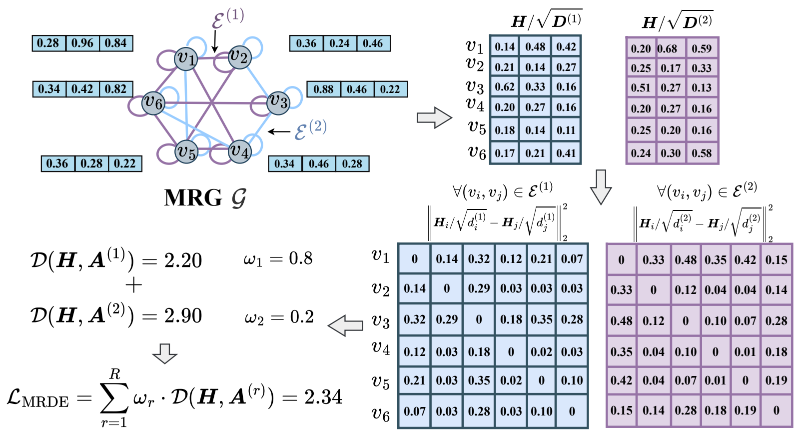


Fig. 3. A running example for MRDE.

is symbolized by $d_i^{(r)}$. In particular, we refer to $\mathcal{G}$ as an *attributed* MRG if each node $v_i \in \mathcal{V}$ is endowed with a D-dimensional attribute vector $\boldsymbol{X}_i$, and otherwise an *attribute-less MRG*. Unless specified otherwise, an MRG $\mathcal{G}$ is assumed to be attributed by default.

We denote by $\boldsymbol{A}^{(r)} \in \{0,1\}^{N\times N}$ the adjacency matrix constructed from the edges in $\mathcal{E}^{(r)}$ and by $\boldsymbol{D}^{(r)}$ the degree matrix whose diagonal entry $\boldsymbol{D}_{i,i}^{(r)} = d_i^{(r)}$. Accordingly, the normalized adjacency matrix $\hat{\boldsymbol{A}}^{(r)}$ is defined as $\hat{\boldsymbol{A}}^{(r)} = \boldsymbol{D}^{(r)^{-\frac{1}{2}}} \boldsymbol{A}^{(r)} \boldsymbol{D}^{(r)^{-\frac{1}{2}}}$ and the normalized Laplacian matrix is $\boldsymbol{I} - \hat{\boldsymbol{A}}^{(r)}$. Additionally, the oriented incidence matrix of $\mathcal{E}^{(r)}$ is symbolized by $\boldsymbol{E}^{(r)} \in \mathbb{R}^{N\times M^{(r)}}$, and $\boldsymbol{E}^{(r)}\boldsymbol{E}^{(r)^\top} = \boldsymbol{D}^{(r)} - \boldsymbol{A}^{(r)}$. In Definition 2.1, we define the (ℓ_1, ℓ_2)*-order maximum eigengap* (OME) of normalized adjacency matrix $\hat{\boldsymbol{A}}$. Table 1 lists the frequently used symbols in this paper.

Definition 2.1 ((ℓ_1, ℓ_2)-Order Maximum Eigengap). Let $\lambda_i(\hat{\boldsymbol{A}})$ be the i^{th} eigenvalue of $\hat{\boldsymbol{A}}$. The (ℓ_1, ℓ_2)-order maximum eigengap is $\mu_{\ell_1,\ell_2} = \max_{1\le i\le N} |\lambda_i(\hat{\boldsymbol{A}})^{\ell_1} - \lambda_i(\hat{\boldsymbol{A}})^{\ell_2}|$.

Multi-Relational Graph Clustering (MRGC). Given an MRG $\mathcal{G}$ and the number K of clusters, the overreaching goal of MRGC is to partition the node set $\mathcal{V}$ into K disjoint groups $\{C_1, \ldots, C_K\}$ (i.e., $\bigcup_{k=1}^{K} C_k = \mathcal{V}$ and $C_i \cap C_j = \varnothing$ for $i \neq j$), such that nodes with high attribute homogeneity and strong connectivity under R relation types are in the same group, while dissimilar and distant ones fall into distinct clusters.

This goal can typically be achieved through two subtasks. Firstly, the task is to construct a feature matrix $\boldsymbol{H}$ that can accurately capture the affinity between nodes in terms of attribute similarity and multiplex structural connectivity in MRGs. Subsequently, clusters $\{C_1, \ldots, C_K\}$ can be derived from $\boldsymbol{H}$ such that similar feature vectors in $\boldsymbol{H}$ are grouped into the same clusters. Particularly, clusters $\{C_1, \ldots, C_K\}$ can be represented in matrix form using an $N \times K$ *node-cluster indicator* (NCI) $\boldsymbol{Y}$ in which

$$\boldsymbol{Y}_{i,k} = \begin{cases} \frac{1}{\sqrt{|C_k|}}, & \text{if } v_i \in C_k, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

2.2 Multi-Relational Dirichlet Energy

The *Dirichlet energy* (DE) [108] of feature matrix $\boldsymbol{H} \in \mathbb{R}^{N\times d}$ over a graph with edges $\mathcal{E}^{(r)}$ is defined by

$$\begin{aligned}\mathcal{D}(\boldsymbol{H}, \boldsymbol{A}^{(r)}) &= \frac{1}{2}\sum_{v_i,v_j\in\mathcal{V}} \boldsymbol{A}^{(r)}_{i,j}\cdot\left\|\boldsymbol{H}_i/\sqrt{\|\boldsymbol{A}^{(r)}_i\|_1}-\boldsymbol{H}_j/\sqrt{\|\boldsymbol{A}^{(r)}_j\|_1}\right\|_2^2 \\ &= \frac{1}{2}\sum_{(v_i,v_j)\in\mathcal{E}^{(r)}}\left\|\boldsymbol{H}_i/\sqrt{d^{(r)}_i}-\boldsymbol{H}_j/\sqrt{d^{(r)}_j}\right\|_2^2,\end{aligned} \tag{2}$$

where $\left\|\boldsymbol{H}_i/\sqrt{d^{(r)}_i}-\boldsymbol{H}_j/\sqrt{d^{(r)}_j}\right\|_2^2$ measures the dissimilarity of the features of two adjacent nodes v_i, v_j in $\mathcal{E}^{(r)}$. Intuitively, $\mathcal{D}(\boldsymbol{H}, \boldsymbol{A}^{(r)})$ assesses the overall *smoothness* of $\boldsymbol{H}$ over $\mathcal{E}^{(r)}$, indicating whether node features in $\boldsymbol{H}$ are similar across adjacent nodes.

To quantify the smoothness of $\boldsymbol{H}$ over the MRG $\mathcal{G}$, we extend the Dirichlet energy to the *multi-relational Dirichlet energy* (MRDE), which is formulated as follows:

$$\mathcal{L}_{\text{MRDE}} = \sum_{r=1}^{R}\omega_r\cdot\mathcal{D}(\boldsymbol{H}, \boldsymbol{A}^{(r)}). \tag{3}$$

$\omega_1, \ldots, \omega_R$ represents the *relation type weights* (hereafter RTWs), which specify the importance of the edges under R relation types, respectively. Particularly, a low MRDE $\mathcal{L}_{\text{MRDE}}$ reflects a high smoothness of $\boldsymbol{H}$ over $\mathcal{G}$, while a high MRDE connotes a large divergence in features of adjacent nodes. In other words, this implies that MRDE can be used to measure the quality of feature matrix $\boldsymbol{H}$ in fusing multiplex structural connectivity in MRG $\mathcal{G}$.

Example 2.2. Figure 3 presents an MRG $\mathcal{G}$ that contains two types of relations ($\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$) and six nodes (i.e., v_1-v_6). The first (resp. second) type of relations is colored in purple (resp. blue). Each node v_i in v_1-v_6 is associated with a 3-dimensional attribute vector $\boldsymbol{H}_i$. By normalizing the attribute vectors by their respective node degrees in two types of relations, i.e., $\boldsymbol{H}_i/\sqrt{d^{(1)}_i}$ and $\boldsymbol{H}_i/\sqrt{d^{(2)}_i}$, we obtain two new node feature matrices $\boldsymbol{H}/\sqrt{\boldsymbol{D}^{(1)}}$ and $\boldsymbol{H}/\sqrt{\boldsymbol{D}^{(2)}}$. For each edge $(v_i, v_j) \in \mathcal{E}^{(1)}$ (resp. $\mathcal{E}^{(2)}$), we calculate $\|\boldsymbol{H}_i/\sqrt{d^{(1)}_i}-\boldsymbol{H}_j/\sqrt{d^{(1)}_j}\|_2^2$ (resp. $\|\boldsymbol{H}_i/\sqrt{d^{(2)}_i}-\boldsymbol{H}_j/\sqrt{d^{(2)}_j}\|_2^2$). Summing up these values, respectively, leads to DE $\mathcal{D}(\boldsymbol{H}, \boldsymbol{A}^{(1)}) = 2.2$ and $\mathcal{D}(\boldsymbol{H}, \boldsymbol{A}^{(2)}) = 2.9$. Suppose that the RTWs are $\omega_1 = 0.8$ and $\omega_2 = 0.2$. The MRDE is then $\mathcal{L}_{\text{MRDE}} = 0.8\times\mathcal{D}(\boldsymbol{H}, \boldsymbol{A}^{(1)}) + 0.2\times\mathcal{D}(\boldsymbol{H}, \boldsymbol{A}^{(2)}) = 2.34$.

Table 2. The MRDE and ACC values by DEMM+ and BMGC [69].

Method	**Metric**	***ACM***	***DBLP***	***ACM2***	***Yelp***	***IMDB***
BMGC	MRDE	1576.6	2837.6	2765.4	2164.5	1456.8
	ACC	93.0	93.4	91.3	91.5	51.0
DEMM+	MRDE	1380.6	2635.6	2505.8	2072.1	1296.4
	ACC	93.6	93.7	91.3	92.7	67.6

Table 2 reports the MRDE values of feature matrices obtained by a state-of-the-art MRGC approach BMGC [69] and our proposed DEMM+, as well as the final clustering accuracies (ACC) on five real datasets, respectively. The empirical results indicate that a smaller MRDE yields a better clustering quality on MRGs.

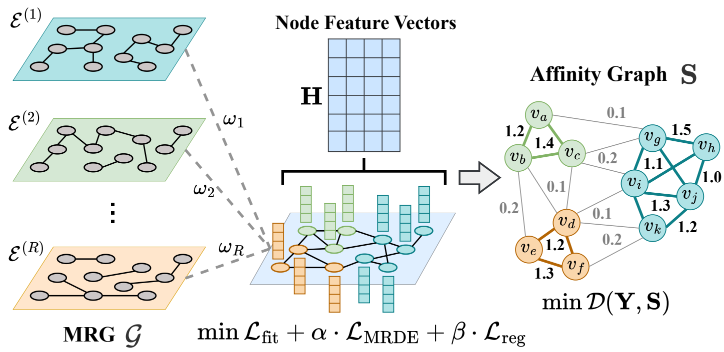


Fig. 4. Two-Stage Optimization Objectives for MRGC.

2.3 Two-Stage Optimization Objectives

Next, we define our two-stage objective functions schematized in Figure 4 for MRGC, based on the notions of DE and MRDE defined in Eq. (2) and Eq. (3).

Stage I Objective. As shown in Figure 4, the first task is to fuse the attribute information in $\boldsymbol{X}$ and the graph structures underlying R types of relations $\{\mathcal{E}^{(1)}, \mathcal{E}^{(1)}, \ldots, \mathcal{E}^{(R)}\}$ into node feature vectors $\boldsymbol{H} \in \mathbb{N}_{\text{row}}$ by optimizing the following objective:

$$\min_{\boldsymbol{H} \in \mathbb{N}_{\text{row}},\ \omega_r \in \mathbb{R}} \mathcal{L}_{\text{fit}} + \alpha \cdot \mathcal{L}_{\text{MRDE}} + \beta \cdot \mathcal{L}_{\text{reg}} \quad \text{s.t.} \quad \sum_{r=1}^{R} \omega_r = 1, \tag{4}$$

where the fitting and regularization terms $\mathcal{L}_{\text{fit}}, \mathcal{L}_{\text{reg}}$ are defined by

$$\mathcal{L}_{\text{fit}} = \|\boldsymbol{H} - \boldsymbol{X}\|_F^2, \quad \mathcal{L}_{\text{reg}} = \sum_{r=1}^{R} \omega_r \cdot \|\hat{\boldsymbol{A}}^{(r)}\|_F^2,$$

and α, β are their respective coefficients. The constraint $\sum_{r=1}^{R} \omega_r = 1$ enforces a normalization on the R RTWs.

More specifically, the fitting term $\mathcal{L}_{\text{fit}}$ seeks to reduce the discrepancy between the target node feature vectors $\boldsymbol{H}$ and initial features[1] $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, whereas the MRDE term $\mathcal{L}_{\text{MRDE}}$ renders feature vectors $\boldsymbol{H}_i$ and $\boldsymbol{H}_j$ of nodes v_i, v_j close to each other when they are connected via an edge of important types, i.e., its RTW ω_r is large. By minimizing MRDE, this stage seeks to obtain node feature vectors $\boldsymbol{H}$ that are consistently smooth over the R types of structural connectivity $\{\mathcal{E}^{(1)}, \mathcal{E}^{(1)}, \ldots, \mathcal{E}^{(R)}\}$ in MRGs. Notably, we additionally incorporate $\mathcal{L}_{\text{reg}}$ to regularize RTWs $\{\omega_r\}_{r=1}^{R}$ with the consideration of the volumes of their associated edges, thereby preventing over-weighting (resp. under-weighting) the large (resp. small) edge set $\mathcal{E}^{(r)}$ (i.e., $\hat{\boldsymbol{A}}^{(r)}$). In a nutshell, the main goal of Stage I is to compute RTWs $\{\omega_r\}_{r=1}^{R}$ automatically by optimizing the objective function to fuse $\{\mathcal{E}^{(r)}\}_{r=1}^{R}$, thereby obtaining node feature vectors $\boldsymbol{H}$ while minimizing MRDE.

Stage II Objective. In the second stage, the goal is to minimize the DE of NCI $\boldsymbol{Y}$ over an affinity graph $\boldsymbol{S}$ constructed from node feature vectors $\boldsymbol{H}$, i.e.,

$$\min_{C_1, \ldots, C_K} \mathcal{D}(\boldsymbol{Y}, \boldsymbol{S}). \tag{5}$$

[1]For notational convenience, we henceforth refer to the node attribute matrix denoised via a principal component analysis as initial features $\boldsymbol{X}$.

Algorithm 1: DEMM Algorithm

Input: An MRG $\mathcal{G}$, parameters α, β, and K.
Output: A set $\{C_1, C_2, \ldots, C_K\}$ of K clusters.

`/* Brute-Force Alternating Optimization */`

$\omega_r \leftarrow \frac{1}{R}\ \forall 1 \leq r \leq R$;
do
    Compute $\hat{\boldsymbol{A}}$ according to Eq. (9);
    Compute $\boldsymbol{H}$ according to Eq.(10);
    Normalize $\boldsymbol{H}$ such that $\boldsymbol{H} \in \mathbb{N}_{\text{row}}$;
    Update ω_r according to Eq. (11) $\forall 1 \leq r \leq R$;
until $\boldsymbol{H}$ *converges*;

`/* Spectral Affinity Graph Clustering */`

Normalize $\boldsymbol{H}$ according to Eq.(13);
Construct affinity matrix $\boldsymbol{S}$ according to Eq. (7);
$\boldsymbol{U} \leftarrow$ the first K eigenvectors of $\boldsymbol{S}$;
Run K-Means over $\boldsymbol{U}$ to generate $\{C_1, \ldots, C_K\}$;

Under certain assumptions on $\boldsymbol{S}$, it can be transformed into

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{v_i \in C_k, v_j \in \mathcal{V} \setminus C_k} \frac{\boldsymbol{S}_{i,j}}{|C_k|}, \tag{6}$$

which is to identify a set $\{C_1, \ldots, C_K\}$ of K clusters that minimize the external connectivity of clusters. As exemplified in Figure 4, clusters v_a-v_c, v_d-v_f, and v_g-v_k are an ideal partitioning of $\mathcal{V}$ over $\boldsymbol{S}$ since the affinity values of inter-partition nodes are merely 0.1 or 0.2, while those of intra-partition nodes are mostly more than 1.0.

In particular, following the conventional choice for the affinity matrix of feature vectors in Euclidean space [68, 72], we employ the *Gaussian kernel* with pairwise distance to measure the affinity of node pair (v_i, v_j):

$$\boldsymbol{S}_{i,j} = \exp\left(-\frac{\|\boldsymbol{H}_i - \boldsymbol{H}_j\|_2^2}{\sigma}\right), \tag{7}$$

where σ is the *kernel width* parameter (typically 1 or 2). To accurately discriminate similar and dissimilar node pairs, node feature vectors $\boldsymbol{H}$ is normalized such that $-1 \leq \boldsymbol{H}_i \cdot \boldsymbol{H}_j \leq 1\ \forall v_i, v_j \in \mathcal{V}$ before constructing $\boldsymbol{S}$. Intuitively, minimizing $\mathcal{D}(Y, \boldsymbol{S})$ is to minimize the Euclidean distances of feature vectors of nodes in the same clusters.

3 The DEMM Method

This section presents our first-cut solution DEMM for MRGC, shown in Algorithm 1. At a high level, DEMM is an approximate method towards optimizing our two-stage objective functions in Eq. (4) and (5) using an alternative optimization and spectral clustering under constraint relaxation, respectively. More concretely, DEMM takes as input an MRG $\mathcal{G}$, coefficients α, β, and the number K of clusters, and runs in two phases. In the following, Section 3.1 details our brute-force alternative optimization method for our first objective in Eq. (4) to construct feature vectors $\boldsymbol{H}$ (Stage I). In Section 3.2, we transform our clustering objective in Eq. (5) to its theoretically equivalent problem and apply a spectral approach to generate clusters $\{C_1, \ldots, C_k\}$ based on $\boldsymbol{H}$ (Stage II). Section 3.3 provides theoretical analyses of DEMM in terms of its correctness and computational complexity.

3.1 Brute-Force Alternating Optimization

Given the hardness of Eq. (4), we resort to an alternative optimization strategy to *approximately* solve this problem. Specifically, we update two variables, i.e., node feature vector $\boldsymbol{H}$ and relation type weights $\{\omega_r\}_{r=1}^{R}$, alternatively, each time fixing one of them and updating the other, using the following rules.

Update $\boldsymbol{H}$ with $\{\omega_r\}_{r=1}^{R}$ fixed. Firstly, for any relation type r, we have the following fact: $\mathcal{D}(\boldsymbol{H}, \mathcal{E}^{(r)}) = \mathtt{trace}(\boldsymbol{H}^\top(\boldsymbol{I} - \hat{\boldsymbol{A}}^{(r)})\boldsymbol{H})$. Given fixed RTWs $\{\omega\}_{r=1}^{R}$, the original optimization objective in Eq. (4) can be simplified as the following partial objective function: $\min_{\boldsymbol{H}\in\mathbb{N}_{\mathrm{row}}} \|\boldsymbol{H} - \boldsymbol{X}\|_F^2 + \alpha \cdot \mathcal{L}_{\mathrm{MRDE}}$, which is equivalent to optimizing

$$\min_{\boldsymbol{H}\in\mathbb{N}_{\mathrm{row}}} \|\boldsymbol{H} - \boldsymbol{X}\|_F^2 + \alpha \cdot \mathtt{trace}(\boldsymbol{H}^\top(\boldsymbol{I} - \hat{\boldsymbol{A}})\boldsymbol{H}), \tag{8}$$

where $\hat{\boldsymbol{A}}$ is the weighted average of $\{\hat{\boldsymbol{A}}^{(r)}\}_{r=1}^{R}$ defined in Eq. (9), henceforth referred to as the *unified normalized adjacency matrix.*

$$\hat{\boldsymbol{A}} = \sum_{r=1}^{R} \omega_r \cdot \hat{\boldsymbol{A}}^{(r)} \tag{9}$$

LEMMA 3.1. *The closed-form solution to Eq.* (8) *is*

$$\boldsymbol{H} = \frac{1}{1+\alpha} \cdot \left(\boldsymbol{I} - \frac{\alpha}{1+\alpha}\hat{\boldsymbol{A}}\right)^{-1} \boldsymbol{X}. \tag{10}$$

Our Lemma 3.1[2] reveals that the optimal $\boldsymbol{H}$ in Eq. (8) (intermediate partial optimum to Eq. (4)) can be obtained through a matrix inverse as in Eq. (10).

Update $\{\omega_r\}_{r=1}^{R}$ with $\boldsymbol{H}$ fixed. When $\boldsymbol{H}$ is at hand, the partial objective function of Eq. (4) can be rewritten as

$$\min_{\{\omega_r\}_{r=1}^{R}} \alpha \sum_{r=1}^{R} \omega_r \cdot \mathtt{trace}\left(\boldsymbol{H}^\top(\boldsymbol{I} - \hat{\boldsymbol{A}}^{(r)})\boldsymbol{H}\right) + \beta \sum_{r=1}^{R} \omega_r \cdot \|\hat{\boldsymbol{A}}^{(r)}\|_F^2$$

such that $\sum_{r=1}^{R} \omega_r = 1$. By leveraging the Cauchy–Schwarz inequality, we can prove that the above partial objective is optimized when we set the RTW

$$\omega_r = \frac{\left(\beta \cdot \|\hat{\boldsymbol{A}}^{(r)}\|_F^2 + \alpha \cdot \mathrm{trace}\left(\boldsymbol{H}^\top(\boldsymbol{I} - \hat{\boldsymbol{A}}^{(r)})\boldsymbol{H}\right)\right)^{-2}}{\sum_{r=1}^{R}\left(\beta \cdot \|\hat{\boldsymbol{A}}^{(r)}\|_F^2 + \alpha \cdot \mathrm{trace}\left(\boldsymbol{H}^\top(\boldsymbol{I} - \hat{\boldsymbol{A}}^{(r)})\boldsymbol{H}\right)\right)^{-2}} \tag{11}$$

for each relation type $1 \le r \le R$. Notice that $\{\|\hat{\boldsymbol{A}}^{(r)}\|_F^2\}_{r=1}^{R}$ can be precomputed and reused in each iteration. We defer the detailed derivative steps to Appendix B [43] for the sake of space.

Based on the above rules for updating $\boldsymbol{H}$ and $\{\omega_r\}_{r=1}^{R}$, DEMM (Algorithm 1) begins by initializing RTWs $\omega_r = \frac{1}{R}\ \forall 1 \le r \le R$ at Line 1. Continuing forth, Algorithm 1 starts an iterative process to update $\boldsymbol{H}$ and $\{\omega_r\}_{r=1}^{R}$ in an alternating fashion (Lines 2-7). To be specific, DEMM first fuses the normalized adjacency matrices of R relation types into the unified normalized adjacency matrix $\hat{\boldsymbol{A}}$ by Eq. (9), followed by an inverse of matrix $\boldsymbol{I} - \frac{\alpha}{1+\alpha}\hat{\boldsymbol{A}}$ to get updated node feature vectors $\boldsymbol{H}$ in Eq. (10) (Lines 3-4). $\boldsymbol{H}$ is further row-normalized such that $\boldsymbol{H} \in \mathbb{N}_{\mathrm{row}}$ at Line 5. After that, Algorithm 1 updates each relation type weight ω_r with the latest $\boldsymbol{H}$ by Eq. (11) at Line 6, and repeats the above procedure until $\boldsymbol{H}$ stabilizes.

[2]All proofs appear in Appendix B in our technical report [43].

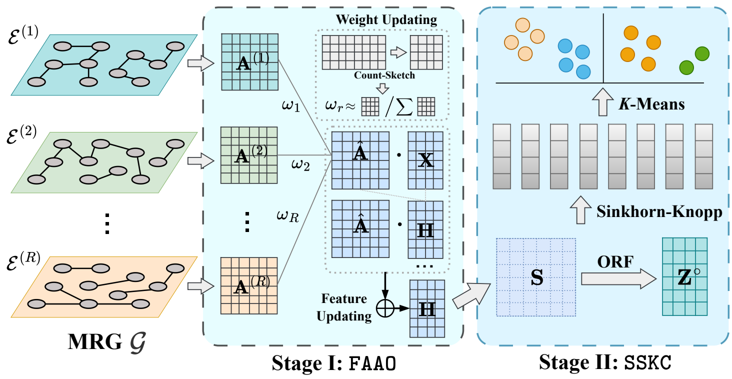


Fig. 5. Overview of DEMM+.

3.2 Spectral Affinity Graph Clustering

Lemma 3.2. *If Y is required to be an $N \times K$ NCI as in Eq.* (1), *then*

$$\min_{Y} \mathcal{D}(Y, S) \Leftrightarrow \max_{Y} \mathtt{trace}(Y^\top S Y). \tag{12}$$

According to Lemma 3.2, our second optimization objective in Eq. (5) can be equivalently transformed to Eq. (12), which is essentially an Ncut problem [72]. Note that the N-cut problem has been proven to be NP-hard [23, 83]. We resort to a standard way of *spectral clustering* [82] to *approximately* solve it by first relaxing the discrete constraint in Eq.(1) on Y, leading to the following objective function:

$$\max_{\tilde{Y} \in \mathbb{R}^{N \times K}} \mathtt{trace}(\tilde{Y}^\top S \tilde{Y}) \quad \text{s.t. } \tilde{Y}^\top \tilde{Y} = \boldsymbol{I},$$

where $\tilde{Y}$ is a continuous version of NCI Y. According to Ky Fan's trace maximization principle [18], the optimal solution is $\boldsymbol{U}$ that contains the first K eigenvectors of the affinity matrix $\boldsymbol{S}$ as columns. The remaining task is then the conversion from $\boldsymbol{U}$ into NCI $\boldsymbol{Y}$ by minimizing their *distance*, which typically can be done using rounding techniques [93, 100] or K-Means.

As illustrated at Lines 8-11 in Algorithm 1, DEMM proceeds to derive clusters from node feature vectors $\boldsymbol{H}$ by first constructing the affinity matrix $\boldsymbol{S}$ according to Eq. (7) (Lines 8-9). Particularly, before computing $\boldsymbol{S}$, for each node $v_i \in \mathcal{V}$, Algorithm 1 applies a standardization $\boldsymbol{H}_i - \overline{h}_i$, followed by an L_2 normalization, i.e.,

$$\boldsymbol{H}_i = \frac{\boldsymbol{H}_i - \overline{h}_i}{\|\boldsymbol{H}_i - \overline{h}_i\|_2}, \tag{13}$$

where $\overline{h}_i$ is the mean of $\boldsymbol{H}_i$, i.e., $\frac{1}{d}\sum_{\ell=1}^{d} \boldsymbol{H}_{i,\ell}$. As stated in Theorem 1 in [76], this operation ensures the affinity $\boldsymbol{H}_i \cdot \boldsymbol{H}_j \in [-1, 1]$ for any two nodes $v_i, v_j \in \mathcal{V}$.

Afterwards, the first K eigenvectors $\boldsymbol{U}$ of $\boldsymbol{S}$ are then calculated through the popular *Arnoldi iterative solver* for partial eigendecomposition [35] at Line 10. Following common practice in spectral clustering, we run K-Means over $\boldsymbol{U}$ to produce NCI Y, i.e., the K clusters $\{C_1, C_2, \ldots, C_K\}$ at Line 11.

3.3 Complexity Analysis

Since Lines 3, 5, and 8 of Algorithm 1 merely involve summation of matrices and matrix normalizations, we focus on analyzing the complexities of computationally intensive operations. Particularly, inverting an $N \times N$ matrix followed by the multiplication with $\boldsymbol{X}$ at Line 4 incurs a time cost of $O(MN + N^2 d)$. Line 6 calculates trace $\left(\boldsymbol{H}^\top(\boldsymbol{I} - \hat{\boldsymbol{A}}^{(r)})\boldsymbol{H}\right)$ when updating each relation type weight

ω_r, leading to a total of $O(Md + Nd^2R)$ time for R relation type weights. In the second stage, Line 9 requires materializing the affinity matrix $\boldsymbol{S}$ in Eq. (7) for all node pairs, consuming $O(N^2d)$ time cost, whereas extracting the first K eigenvectors of $\boldsymbol{S}$ at Line 10 can be done in $O(N^2K)$ time [35]. Therefore, the overall time complexity of DEMM is bounded by $O(MN + N^2d + Nd^2R)$.

Regarding space overhead, since the matrix inversion in Eq. (11) yields an $N \times N$ dense matrix and Line 9 materializes an $N \times N$ affinity matrix $\boldsymbol{S}$, the total space complexity of DEMM is $O(N^2)$.

4 The DEMM+ Algorithm

Despite achieving high clustering quality as exhibited in experiments (Section 5), DEMM incurs quadratic computational cost and space overhead, and thus, is incompetent for large MRGs. As pinpointed in the preceding section, the colossal time and storage space are ascribed to the materialization of $N\times N$ *dense* matrices and expensive matrix operations, including inversion, multiplication, and eigendecomposition, in either the construction of node feature vectors $\boldsymbol{H}$ or the generation of clusters $\{C_1, C_2, \ldots, C_K\}$. To alleviate such issues, this section further proposes DEMM+ for MRGC, which is able to advance MRG clustering performance in efficiency without compromising the effectiveness.

Figure 5 depicts an overview of DEMM+. Akin to DEMM, DEMM+ consists of two secondary algorithms, FAAO and SSKC, for the constructions of $\boldsymbol{H}$ and $\{C_1, C_2, \ldots, C_K\}$, respectively. At a high level, DEMM+ develops a truncated approximation for $\boldsymbol{H}$ and sketching-based estimations for RTWs in the first stage. Subsequently, it transforms the costly spectral clustering in Stage II to a cheap K-Means by adjusting $\boldsymbol{S}$. In Section 4.1, we first elucidate the algorithmic details of FAAO, which approximately updates $\boldsymbol{H}$ and RTWs $\{\omega_r\}_{r=1}^R$ alternatively towards optimizing our objective in Eq. (4) using linear time and space. In lieu of optimizing Eq. (12) to get clusters $\{C_1, C_2, \ldots, C_K\}$ via the explicit construction of the $N \times N$ affinity graph $\boldsymbol{S}$ and costly spectral clustering, Section 4.2 presents our SSKC method that achieves a linear computational time complexity through a theoretically-grounded problem transformation and innovative adoption of mathematical apparatus, i.e., orthogonal random features and Sinkhorn-Knopp normalization. Lastly, we further extend DEMM+ to handle attribute-less MRGs (dubbed as DEMM-NA). The algorithmic details are deferred to Appendix A [43] for the interest of space.

4.1 Fast Approximate Alternating Optimization

Recall that in Section 3.1, the leading cause of the immense computational burden of building $\boldsymbol{H}$ is the inversion of $\boldsymbol{I} - \frac{\alpha}{1+\alpha}\hat{\boldsymbol{A}}$ in Eq. (10), which needs an $O(N^3)$ time. On top of that, although $\{\|\hat{\boldsymbol{A}}^{(r)}\|_F^2\}_{r=1}^R$ can be precomputed and the exact calculation of trace $\left(\boldsymbol{H}^\top(\boldsymbol{I} - \hat{\boldsymbol{A}}^{(r)})\boldsymbol{H}\right)$ for each relation type r in Eq. (11) takes a linear time of $O(Nd^2 + M^{(r)}d)$ per iteration, the overall computational expenditure for updating R relation type weights $\{\omega_r\}_{r=1}^R$ for multiple iterations is also significant. Subsequently, we delineate the rationale behind FAAO for tackling these efficiency challenges.

Theorem 4.1 ([28]). *Let $\boldsymbol{M}$ be a matrix whose dominant eigenvalue $\lambda(\boldsymbol{M})$ satisfies $|\lambda(\boldsymbol{M})| < 1$. Then, the inverse $(\boldsymbol{I} - \boldsymbol{M})^{-1}$ can be expanded as a Neumann series: $(\boldsymbol{I} - \boldsymbol{M})^{-1} = \sum_{\ell=0}^{\infty} \boldsymbol{M}^\ell$.*

Lemma 4.2. *Let $\lambda(\hat{\boldsymbol{A}})$ be the dominant eigenvalue of $\hat{\boldsymbol{A}}$. $|\lambda(\hat{\boldsymbol{A}})| \le 1$.*

Basic Idea. As per our theoretical outcome in Lemma 4.2, the dominant eigenvalue of $\frac{1}{1+\alpha}\hat{\boldsymbol{A}}$ is bounded by $\frac{1}{1+\alpha} < 1$. Combining it with Theorem 4.1 transforms Eq. (10) into an equivalent form:

$$\boldsymbol{H} = \frac{1}{1+\alpha}\sum_{\ell=0}^{\infty}\left(\frac{1}{1+\alpha}\right)^{\ell}\hat{\boldsymbol{A}}^{\ell}\boldsymbol{X}, \tag{14}$$

Algorithm 2: FAAO Algorithm

Input: An MRG $\mathcal{G}$, parameters α, β, and L.
Output: Node feature vectors $\boldsymbol{H}$

$\omega_r = \frac{1}{R}\ \forall 1 \leq r \leq R$;
$\tilde{\boldsymbol{E}}^{(r)} \leftarrow \texttt{CountSketch}(\hat{\boldsymbol{E}}^{(r)}, m)\ \forall 1 \leq r \leq R$;
do
  Compute $\hat{\boldsymbol{A}}$ by Eq. (9);
  $\widehat{\boldsymbol{X}}^{(0)} \leftarrow \frac{1}{1+\alpha} \cdot \boldsymbol{X},\ \boldsymbol{H} \leftarrow \widehat{\boldsymbol{X}}^{(0)}$;
  for $\ell \leftarrow 1$ **to** L **do**
    $\widehat{\boldsymbol{X}}^{(\ell)} \leftarrow \frac{\alpha}{1+\alpha} \cdot \hat{\boldsymbol{A}}\widehat{\boldsymbol{X}}^{(\ell-1)}$;
    $\boldsymbol{H} \leftarrow \boldsymbol{H} + \widehat{\boldsymbol{X}}^{(\ell)}$;
  $\boldsymbol{H} \leftarrow \boldsymbol{H} + \alpha \cdot \widehat{\boldsymbol{X}}^{(L)}$;
  Normalize $\boldsymbol{H}$ such that $\boldsymbol{H} \in \mathbb{N}_{\text{row}}$;
  Update ω_r according to Eq. (16) $\forall 1 \leq r \leq R$;
until $\boldsymbol{H}$ *converges*;

which remains the optimal solution to our conditional objective function in Eq. (8) when RTWs are fixed. Although Eq. (14) offers an iterative way of calculating $\boldsymbol{H}$, its exact computation requires summing up an infinite series, which is infeasible.

Notice that $\hat{\boldsymbol{A}}^L$ can be interpreted as L-hop random walks over $\mathcal{G}$, wherein each entry $\hat{\boldsymbol{A}}^L_{i,j}$ signifies the probability of a random walk originating from node v_i visiting node v_j at the L-th hop. Accordingly, the term $\sum_{\ell=0}^{\infty}\left(\frac{1}{1+\alpha}\right)^{\ell}\hat{\boldsymbol{A}}^{\ell}$ in $\boldsymbol{H}$ can be perceived as the total probabilities of random walks of various lengths, where length-ℓ random walks are weighted with $\left(\frac{1}{1+\alpha}\right)^{\ell}$. As such, one potential solution to estimate $\boldsymbol{H}$ is to discard long random walks, i.e., random walks beyond L (L is a small integer) hops, as their weights are lower.

Due to the *mixing time* [36] of random walks on graphs, the L-hop random walk probability $\hat{\boldsymbol{A}}^L_{i,j}$ converges to an invariant value $a_{i,j}$ after a number of steps. Mathematically, the overall discrepancy between $(L+1)$-hop and L-hop random walk probabilities $\|\hat{\boldsymbol{A}}^{L+1} - \hat{\boldsymbol{A}}^{L}\|_2$ can be proved to be equal to the $(L, L+1)$-OME $\mu_{L,L+1}$:

$$\|\hat{\boldsymbol{A}}^{L+1} - \hat{\boldsymbol{A}}^{L}\|_2 = \mu_{L,L+1}.$$

As reported in Figure 6, $(L, L+1)$-OME of real MRGs *DBLP* [104] and *Yelp* [71] dwindles to nearly zero when L is roughly 8, indicating that the convergence/mixing of $\hat{\boldsymbol{A}}^L$ can be achieved with merely a handful of hops. Inspired by this, our idea is to compute an approximate $\boldsymbol{H}$,

$$\begin{aligned}\boldsymbol{H} &\approx \frac{1}{1+\alpha}\sum_{\ell=0}^{L}\left(\frac{\alpha}{1+\alpha}\right)^{\ell}\hat{\boldsymbol{A}}^{\ell}\boldsymbol{X} + \frac{1}{1+\alpha}\sum_{\ell=L+1}^{\infty}\left(\frac{\alpha}{1+\alpha}\right)^{\ell}\hat{\boldsymbol{A}}^{L}\boldsymbol{X} \\ &= \frac{1}{1+\alpha}\sum_{\ell=0}^{L}\left(\frac{\alpha}{1+\alpha}\right)^{\ell}\hat{\boldsymbol{A}}^{\ell}\boldsymbol{X} + \left(\frac{\alpha}{1+\alpha}\right)^{L+1}\hat{\boldsymbol{A}}^{L}\boldsymbol{X},\end{aligned} \tag{15}$$

wherein the terms $\hat{\boldsymbol{A}}^{\ell}$ beyond L-th orders ($\ell \geq L+1$) are estimated using $\hat{\boldsymbol{A}}^{L}$. In doing so, $\boldsymbol{H}$ can be efficiently calculated as L is merely up to a few dozen in practice.

Lemma 4.3. *Let* $\hat{\boldsymbol{E}}^{(r)} = \boldsymbol{D}^{(r)-\frac{1}{2}}\boldsymbol{E}^{(r)}$. $\texttt{trace}\left(\boldsymbol{H}^{\top}(\boldsymbol{I} - \hat{\boldsymbol{A}}^{(r)})\boldsymbol{H}\right) = \|\boldsymbol{H}^{\top}\hat{\boldsymbol{E}}^{(r)}\|_F^2\ \forall 1 \leq r \leq R.$

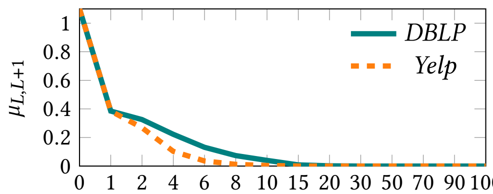


Fig. 6. The OME $\mu_{L,L+1}$ when varying L.

On the other hand, Lemma 4.3 suggests that we can leverage the matrix norm $\|\boldsymbol{H}^\top\hat{\boldsymbol{E}}^{(r)}\|_F^2$ instead of the matrix trace for updating RTW ω_r in Eq. (11) in $O(M^{(r)}d)$ time since the normalized oriented incidence matrix $\hat{\boldsymbol{E}}^{(r)}$ contains $2M^{(r)}$ non-zero entries and can be materialized in the preprocessing. This time cost can be further reduced if a low-dimensional sparse matrix $\tilde{\boldsymbol{E}}^{(r)} \in \mathbb{R}^{N\times m}$ ($m \ll M^{(r)}$ and $\mathtt{nnz}(\tilde{\boldsymbol{E}}^{(r)}) \ll M^{(r)}$) can be created such that $\|\boldsymbol{H}^\top\hat{\boldsymbol{E}}^{(r)}\|_F^2 \approx \|\boldsymbol{H}^\top\tilde{\boldsymbol{E}}^{(r)}\|_F^2$ for estimating ω_r.

Algorithm. Algorithm 2 displays the pseudo-code of FAAO. Similar in spirit to the brute-force approach in Section 3.1, FAAO initializes ω_r as $\frac{1}{R}$ for each relation type at Line 1, and iteratively updates $\boldsymbol{H}$ and $\{\omega_r\}_{r=1}^R$ (Lines 3-12). The differences are as follows. Algorithm 2 takes as input additional parameters m, L and generates an m-dimensional approximation $\tilde{\boldsymbol{E}}^{(r)}$ of $\hat{\boldsymbol{E}}^{(r)}$ via CountSketch [11] at Line 2 before entering the iterations. Moreover, in each iteration, FAAO builds terms $\widehat{\boldsymbol{X}}^{(\ell)} = \left(\frac{\alpha}{1+\alpha}\right)^\ell \hat{\boldsymbol{A}}^\ell \boldsymbol{X}\ \forall 0 \le \ell \le L$ using L rounds of iterative sparse matrix multiplications (Lines 5-8), followed by assembling them with $\alpha \cdot \widehat{\boldsymbol{X}}^{(L)}$ into $\boldsymbol{H}$ as in Eq. (15) at Line 9. On the basis of updated node feature vectors $\boldsymbol{H}$ and precomputed $\{\|\hat{\boldsymbol{A}}^{(r)}\|_F^2\}_{r=1}^R$, FAAO calculates matrix norm $\|\boldsymbol{H}'^\top\tilde{\boldsymbol{E}}^{(r)}\|_F^2$ for each relation type and updates the estimated relation type weight ω_r by

$$\omega_r = \frac{\left(\beta\cdot\|\hat{\boldsymbol{A}}^{(r)}\|_F^2 + \alpha\cdot\|\boldsymbol{H}'^\top\tilde{\boldsymbol{E}}^{(r)}\|_F^2\right)^{-2}}{\sum_{r=1}^R\left(\beta\cdot\|\hat{\boldsymbol{A}}^{(r)}\|_F^2 + \alpha\cdot\|\boldsymbol{H}'^\top\tilde{\boldsymbol{E}}^{(r)}\|_F^2\right)^{-2}}. \tag{16}$$

Correctness Analysis. Denote by $\boldsymbol{H}^*$ the exact node feature vectors defined in Eq. (14). The following theorem establishes the approximation guarantees of $\boldsymbol{H}$ obtained at Line 9 in Algorithm 2.

Theorem 4.4. *$\|\boldsymbol{H}-\boldsymbol{H}^*\|_F \le \sum_{\ell=L+1}^\infty \frac{\alpha^\ell}{(1+\alpha)^{\ell+1}}\left\|\hat{\boldsymbol{A}}^\ell - \hat{\boldsymbol{A}}^L\right\|_2 \cdot \|\boldsymbol{X}\|_F$, which can be upper bounded by $\left(\frac{\alpha}{1+\alpha}\right)^{L+1}\cdot\|\boldsymbol{X}\|_F\cdot\max_{\ell\ge1}\mu_{L,L+\ell}$.*

Recall that in Figure 6, the empirical values of $(L, L+1)$-OME are negligible when L is small, which implies that $\hat{\boldsymbol{A}}^L$ is close to $\hat{\boldsymbol{A}}^{L+1}$, and thus, $\hat{\boldsymbol{A}}^{L+\ell}$ for $\ell > L+1$, rendering approximation error $\|\boldsymbol{H}-\boldsymbol{H}^*\|_F = 0$.

As for the relation type weights $\{\omega_r\}_{r=1}^R$ in Eq. (16), FAAO harnesses $\left\|\boldsymbol{H}^\top\tilde{\boldsymbol{E}}^{(r)}\right\|_F^2$ as an approximation of $\mathtt{trace}\left(\boldsymbol{H}^\top(\boldsymbol{I}-\hat{\boldsymbol{A}}^{(r)})\boldsymbol{H}\right)$. Particularly, we can derive the following corollary using Theorem 11 in Ref. [11]:

Corollary 4.5. *Let $\boldsymbol{Q} \in \mathbb{R}^{M\times m}$ be a count-sketch matrix and $\tilde{\boldsymbol{E}}^{(r)} = \hat{\boldsymbol{E}}^{(r)}\boldsymbol{Q}$, where $m = O(r\epsilon^{-4}\log(r/\epsilon\delta)\cdot(r+\log(1/\epsilon\delta)))$, ϵ is an error threshold and r is the rank of $\hat{\boldsymbol{E}}^{(r)}$. Then,*

$$\left\|\boldsymbol{H}^\top\tilde{\boldsymbol{E}}^{(r)}\right\|_F^2 = (1\pm\epsilon)^2\cdot\mathtt{trace}\left(\boldsymbol{H}^\top(\boldsymbol{I}-\hat{\boldsymbol{A}}^{(r)})\boldsymbol{H}\right)$$

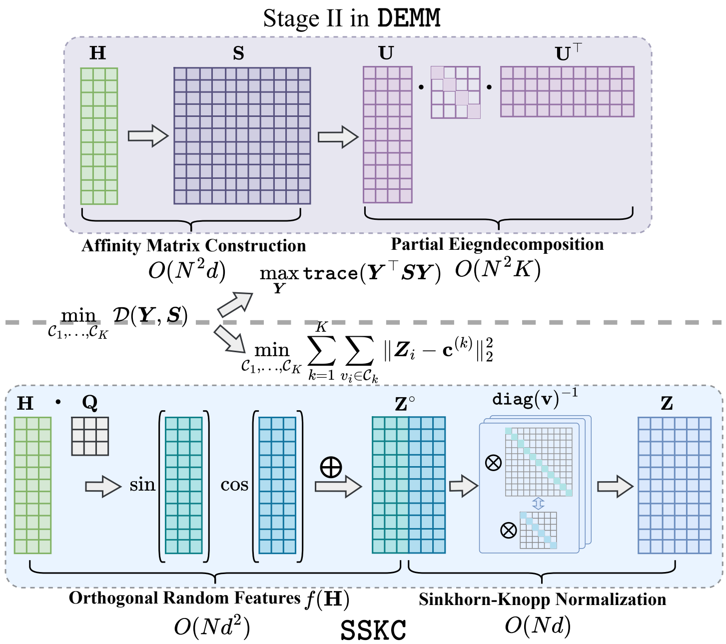


Fig. 7. Illustration of SSKC.

holds with a probability of at least $1 - \delta$.

As empirically validated in Appendix D.4 [43], a small m (e.g., 20) leads to an accurate approximation of $\hat{\boldsymbol{E}}^{(r)}$, ensuring excellent and stable final clustering quality.

Complexity Analysis. Recall that the invocation of CountSketch at Line 2 essentially computes $\hat{\boldsymbol{E}}^{(r)}\mathbf{R}^\top$, where $\hat{\boldsymbol{E}}^{(r)}$ is the normalized oriented incidence matrix of $\mathcal{E}^{(r)}$ with $2M^{(r)}$ non-zero entries (each column has two entries) and sketching matrix $\mathbf{R} \in \mathbb{R}^{m \times M^{(r)}}$ ($m \ll M^{(r)}$) solely has a single non-zero entry in each column. The sparse matrix multiplication $\hat{\boldsymbol{E}}^{(r)}\mathbf{R}^\top$ hence entails $O(M^{(r)})$ time, summing up to $O(M)$ time for all the R relation types. In each iteration (Lines 4-11) of the alternative optimization, the dominant computational overhead lies in Lines 7 and 11. The former costs $O(Md)$ time for each sparse matrix multiplication $\hat{\boldsymbol{A}}\widehat{\boldsymbol{X}}^{(\ell-1)}$, and hence, $O(MLd)$ time for L rounds, while the latter calculates $\|\boldsymbol{H}^\top \tilde{\boldsymbol{E}}^{(r)}\|_F^2$ for updating each relation type weight ω_r, which needs $NdmR$ operations for all the R relation types. In short, the time cost of each iteration for updating $\boldsymbol{H}$ and $\{\omega_r\}_{r=1}^R$ is $O(MLd + NdmR)$. Given that L, m, and the number of iterations are at most a few dozen in practice, and thus, can be considered as constants, the overall time complexity of FAAO is $O(Md + NdR)$.

Algorithm 2 only needs incidence and adjacency matrices with $O(M)$ non-zero entries in total, sketched incidence matrix $\tilde{\boldsymbol{E}}^{(r)} \in \mathbb{R}^{N \times m}$, and $N \times d$ intermediate feature vectors $\widehat{\boldsymbol{X}}^{(\ell)}$ and $\boldsymbol{H}$ in the main memory. Consequently, its space cost is $O(M + Nd + Nm)$, which equals $O(M + Nd)$ when m is regarded as a constant.

Let w_r^* be the new weight of the next iteration. Define $\boldsymbol{\Delta}$ as

$$\boldsymbol{\Delta} = \sum_{r=1}^{R} (w_r^* - w_r) \cdot \hat{\boldsymbol{A}}^{(r)}. \tag{17}$$

Algorithm 3: SSKC Algorithm

Input: Node feature vectors $\boldsymbol{H}$ and the number K of clusters
Output: A set of K clusters $\{C_1, \ldots, C_K\}$.

Normalize $\boldsymbol{H}$ according to Eq.(13);
$\boldsymbol{Z}^\circ \leftarrow \mathtt{ORF}(\boldsymbol{H})$;
$\overleftarrow{\boldsymbol{Z}} \leftarrow \boldsymbol{Z}^\circ$, $\overrightarrow{\boldsymbol{Z}} \leftarrow \boldsymbol{Z}^\circ$;
do
  $\mathbf{v} \leftarrow \overleftarrow{\boldsymbol{Z}} \cdot \left(\overrightarrow{\boldsymbol{Z}}^\top \cdot \mathbf{1}\right)$;
  $\overleftarrow{\boldsymbol{Z}} \leftarrow \mathtt{diag}(\mathbf{v})^{-1} \cdot \overleftarrow{\boldsymbol{Z}}$;
  $\mathbf{v} \leftarrow \left(\mathbf{1}^\top \cdot \overleftarrow{\boldsymbol{Z}}\right) \cdot \overrightarrow{\boldsymbol{Z}}^\top$;
  $\overrightarrow{\boldsymbol{Z}} \leftarrow \mathtt{diag}(\mathbf{v})^{-1} \cdot \overrightarrow{\boldsymbol{Z}}$;
until $\overrightarrow{\boldsymbol{Z}}$ *converges*;
Run K-Means over $\overrightarrow{\boldsymbol{Z}}$ to generate $\{C_1, \ldots, C_K\}$;

The new normalized adjacency matrix of the next iteration is

$$\hat{\boldsymbol{A}}^* = \hat{\boldsymbol{A}} + \Delta. \tag{18}$$

4.2 Symmetric Sinkhorn-Knopp Clustering

Theorem 4.6. *If $\boldsymbol{S}$ is doubly stochastic and $\boldsymbol{S} = \boldsymbol{Z}\boldsymbol{Z}^\top$, optimizing Eq.* (5) *is equivalent to optimizing* $\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{v_i \in C_k} \|\boldsymbol{Z}_i - \mathbf{c}^{(k)}\|_2^2$, *where* $\mathbf{c}^{(k)} = \sum_{v_j \in C_k} \frac{\boldsymbol{Z}_j}{|C_k|}$ *stands for the center of cluster C_k.*

Basic Idea. As remarked in Figure 7, DEMM relies on a partial eigendecomposition of the $N \times N$ dense affinity matrix $\boldsymbol{S}$ to approximately solve the NP-hard problem in Eq. (5), which takes $O(N^2 \cdot (d+K))$ time and is still prohibitively expensive. Our theoretical finding in Theorem 4.6 pinpoints that the clustering objective is equivalent to minimizing the *within-cluster sum of squares* (WCSS) on a matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times z}$ that satisfies $\boldsymbol{Z}\boldsymbol{Z}^\top = \boldsymbol{S}$ where $\boldsymbol{S}$ is *doubly stochastic*. This implies that the above spectral clustering over $\boldsymbol{S}$ can be further transformed and simplified into a tractable task, i.e., running K-Means over $\boldsymbol{Z}$, if we make an adjustment to (a normalization) $\boldsymbol{S}$ and calculate $\boldsymbol{Z}$ such that $\boldsymbol{Z}\boldsymbol{Z}^\top = \boldsymbol{S}$ is doubly stochastic. Doing so sidesteps the costly eigendecomposition, and hence, results in a time cost of $O(NKz)$, which is almost linear when $z \ll N$.

To make $\boldsymbol{Z}\boldsymbol{Z}^\top = \boldsymbol{S}$ doubly stochastic, a straightforward way is to first materialize the affinity matrix $\boldsymbol{S}$ as in DEMM, apply a doubly stochastic normalization of $\boldsymbol{S}$, and then decompose it into the product of $\boldsymbol{Z}$ and its transpose, all of which, however, are rather costly. Inspired by the *kernel tricks* [49], the idea of SSKC is to eliminate the need to explicitly materialize $\boldsymbol{S}$ via a mapping function $f(\cdot)$ on $\boldsymbol{H}$ such that

$$\boldsymbol{S} \approx f(\boldsymbol{H}) \cdot f(\boldsymbol{H})^\top,$$

and $f(\boldsymbol{H})$ can be used as $\boldsymbol{Z}$ for subsequent K-Means clustering. Since $\boldsymbol{S}$ is defined using a Gaussian kernel, such a mapping function $f(\cdot)$ can be derived via *random Fourier features* (RFF) [66]. RFF serves as an alternative to the Gaussian kernel, reducing the computational complexity of kernel methods from nonlinear to linear. That is to say, RFF leverages the Bochner theorem [66] to map the kernel function with $f(\cdot)$, which avoids computing Eq. (7) with $O(N^2)$ computational complexity. Along this line, the next task is to make $\boldsymbol{Z}\boldsymbol{Z}^\top$ doubly stochastic.

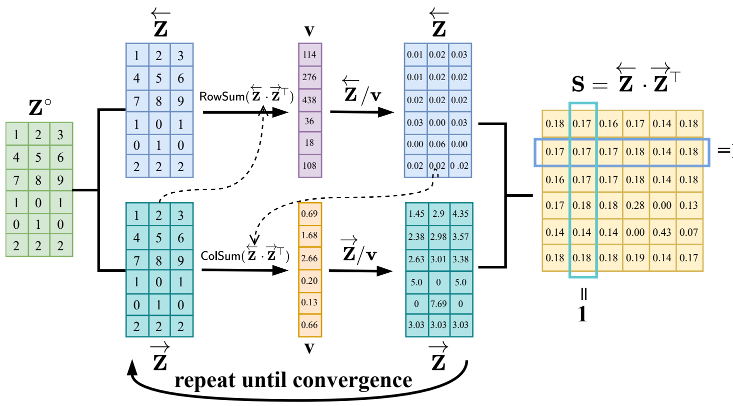


Fig. 8. A running example for the SK normalization.

Algorithm. Figure 7 summarizes the core steps of SSKC. It first constructs the mapping function $f(\cdot)$ and $Z^\circ = f(H)$, i.e., the initial version of Z, using random Fourier features, followed by a normalization of Z° into Z for subsequent clustering, both of which can be done in $O(Nd)$ time.

In Algorithm 3, we present the details of SSKC. Initially, SSKC leverages the *Orthogonal Random Features* (ORF) technique [99] as the mapping function $f(\cdot)$ to transform node feature vectors H to Z°, an initial version of target Z, such that $Z^\circ {Z^\circ}^\top \approx S$ (Line 1). More concretely, ORF first transforms H into $\tilde{H} = H \cdot Q^\top$, using a uniformly distributed random orthogonal matrix $Q \in \mathbb{R}^{d\times d}$, and then constructs Z° by

$$Z^\circ = \frac{1}{\sqrt{d}} \cdot (\mathtt{sin}(\tilde{H}) \,\|\, \mathtt{cos}(\tilde{H})),$$

where $\|$ denotes the horizontal concatenation operator for matrices. It is worth mentioning that the resulting feature dimension z of Z° is $2d$ and $d \ll N$. Subsequently, SSKC begins the doubly stochastic normalization of $Z^\circ {Z^\circ}^\top$. We introduce *Sinkhorn-Knopp algorithm* [73] (SK), which obtains a doubly stochastic matrix by iteratively normalizing the rows and columns of the affinity matrix $ZZ^\top$. Instead of simply employing the SK that requires materializing $Z^\circ {Z^\circ}^\top$ for normalization, Algorithm 3 initializes $\overleftarrow{Z}$ and $\overrightarrow{Z}$ as Z° at Line 2 and iteratively normalizes them alternately (Lines 3-8), thereby enforcing $\overleftrightarrow{Z Z}^\top$ bistochastic. Particularly, in each iteration, SSKC computes the row sum vector $\mathbf{v}$ of $\overleftrightarrow{Z Z}^\top$ using a trick reordering the matrix multiplication as $\overleftarrow{Z} \cdot \left(\overrightarrow{Z}^\top \cdot \mathbf{1}\right)$ for higher efficiency, followed by normalizing each row $\overleftarrow{Z}$ by $\mathtt{diag}(\mathbf{v})^{-1} \cdot \overleftarrow{Z}$ (Lines 4-5). In the same vein, $\overrightarrow{Z}$ is normalized by the column sum vector of $\overleftrightarrow{Z Z}^\top$ (Lines 6-7). As such, at the end of each iteration, a symmetric normalization of rows and columns is imposed on $\overleftrightarrow{Z Z}^\top$. The following theorem indicates that $\overleftrightarrow{Z Z}^\top$ is doubly stochastic with sufficient iterations and $\overleftarrow{Z} = \overrightarrow{Z} = f(H)$.

Theorem 4.7. *$\overleftrightarrow{Z Z}^\top$ is doubly stochastic and $\overleftarrow{Z} = \overrightarrow{Z}$.*

Finally, Algorithm 3 applies the K-Means over $\overrightarrow{Z}$ and generates clusters $\{C_1, C_2, \ldots, C_K\}$.

Example 4.8. Figure 8 exemplifies how SSKC leverages the SK normalization to achieve $S = ZZ^\top$. Given a 6×3 feature matrix Z° output by ORF (see example in Appendix C.3 [43]), we initialize $\overleftarrow{Z} = \overrightarrow{Z} = Z^\circ$. In the first iteration, SK calculates the sum of entries in each row of $\overleftrightarrow{Z Z}^\top$, yielding a vector $\mathbf{v}$ with six rows $[114, 276, 438, 36, 18, 108]^\top$. Afterwards, six rows in $\overleftarrow{Z}$ are normalized by

Table 3. Statistics of Datasets.

Dataset	N	**Relation Types**	M	D	K
ACM	3K	Paper-Subject-Paper	2.2M	1,870	3
		Paper-Author-Paper	29.3K		
DBLP	4K	Author-Paper-Author	11.1K	334	4
		Author-Paper-Venue-Paper-Author	5M		
		Author-Paper-Term-Paper-Author	6.8M		
ACM2	4K	Paper-Subject-Paper	4.3M	1,902	3
		Paper-Author-Paper	58K		
Yelp	2.6K	Business-User-Business	528.3K	82	3
		Business-Rating-Business	1.5M		
		Business-Service-Business	2.5M		
IMDB	3.6K	Movie-Actor-Movie	66.4K	2,000	3
		Movie-Director-Movie	13.8K		
Protein	18.8K	Protein-Protein	2.0M	1280	6
		Protein-Gene-Protein	18.9K		
		Protein-Disease-Protein	60.1K		
Amazon	11.9K	User-Product-User	363.2K	25	2
		User-Star-User	7.1M		
		User-Review-User	2.1M		
MAG	113.9K	Paper-Paper	1.8M	128	4
		Paper-Author-Paper	10.1M		
OAG-ENG	370.6K	Paper-Field-Paper	14.6M	768	20
		Paper-Author-Paper	455.7K		
		Paper-Paper	2.1M		
OAG-CS	546.7K	Paper-Field-Paper	53.9M	768	20
		Paper-Author-Paper	1.6M		
		Paper-Paper	11.7M		
RCDD	11.9M	Item-b-Item	421.1M	256	2
		Item-f-Item	353.7M		

dividing their respective entries in $\mathbf{v}$, e.g., $[1,2,3]/114 = [0.01, 0.02, 0.03]$. Based on the updated $\overleftarrow{Z}$, we start to normalize $\overrightarrow{Z}$. SK then calculates the sum of entries in each column of $\overleftarrow{Z}\overrightarrow{Z}^{\top}$, leading to a new length-6 vector $\mathbf{v} = [0.69, 1.68, 2.66, 0.2, 0.13, , 0.66]^{\top}$. $\overrightarrow{Z}$ is subsequently updated by dividing each row by its respective entry in the new $\mathbf{v}$. By repeating the above alternate procedure sufficiently, we can finally obtain $\overleftarrow{Z} = \overrightarrow{Z}$ such that the entries in each row and column of $\boldsymbol{S} = \overleftarrow{Z}\overrightarrow{Z}^{\top}$ sum up to 1.0, i.e., doubly stochastic. As such, the clusters can be obtained by simply running K-means over row vectors of $\overleftarrow{Z}$ or $\overrightarrow{Z}$.

Complexity Analysis. According to [99], Z° can be obtained in $O(Nd^2)$ time. By reordering the matrix multiplications as in Lines 5 and 7, $\mathbf{v}$ can be calculated using $O(Nd)$ time. Since the normalizations at Lines 6 and 8 involve Nd operations, each iteration (Lines 5-8) then takes $O(Nd)$ time. Recall that K-Means runs in $O(NK)$ time per iteration. In sum, the total time cost of SSKC is bounded by $O(Nd^2 + NK)$ when the numbers of iterations are considered as constants. Its space cost is $O(Nd)$ since $\boldsymbol{H}$ and Z° contain Nd and $2Nd$ entries, respectively.

5 Experiments

This section experimentally evaluates DEMM, DEMM+, and DEMM-NA against 20 competitors regarding clustering quality and efficiency on 9 real MRGs of varied volumes. All experiments are conducted on a Linux machine with an NVIDIA Ampere A100 GPU (80 GB memory), AMD EPYC 7513 CPUs (2.6 GHz), and 1TB RAM. The codes of all algorithms are collected from their respective authors, and all are implemented in Python, except LMVSC and MCGC. For reproducibility, the source code and datasets are available at https://github.com/HKBU-LAGAS/DEMM.

Table 4. Summary of evaluated method.

	Category	**Complexity**	**Objective**	**Backbone**
DMG	MRGC (MVE)	$O(RNd + Md)$	Reconstruction	GNN
DuaLGR	MRGC (MRS)	$O(RN^2)$	Reconstruction	GNN
MGDCR	MRGC (MVE)	$O(R^2Nd^2 + Md)$	Mutual Info. Max.	GNN
DMGI	MRGC (MRS)	$O(RNd^2 + Md)$	Modularity Max.	GNN
MvAGC	MRGC(MRS)	$O(Nd^2)$	Subspace Clustering	-
MGDCR	MRGC(MVE)	$O(MN + NK^2)$	Subspace Clustering	-
BTGF	MRGC(MVE)	$O(N^2d + M^2Nd^2)$	Reconstruction	GNN
BMGC	MRGC(MVE)	$O(MN^2 + MNd)$	Contrastive	GNN
MCGC	MVGC	$O(MN^2(d + K))$	Contrastive	-
LMVSC	MVGC	$O(MN + NK^2)$	Subspace Clustering	-
MMGC	MVGC	$O(MN^2K + MNK)$	Subspace Clustering	-
DMoN	AGC	$O(Nd^2 + Md)$	Contrastive	GNN
Dink-Net	AGC	$O(NdK + dK^2)$	Adversarial	GNN
S3GC	AGC	$O(Nd^2)$	Contrastive	GNN
S2AGC	AGC	$O(NKd)$	Subspace Clustering	-
DEMM	MRGC	$O(MN + Nd(N + dR))$	MRDE	-
DEMM+	MRGC	$O(Nd^2 + Md)$	MRDE	-

5.1 Experimental Setup

Datasets. We experiment with 11 benchmark MRG datasets of varied volumes and types, whose statistics are presented in Table 3. Amid them, *ACM* [19], *ACM2* [22], *DBLP* [104], *MAG* [30], *OAG-CS*, and *OAG-ENG* [102] are academic citation networks; *Yelp* [71] and *Amazon* [61] are e-commerce review networks; *IMDB* [87]is a movie review network; *RCDD* [50] is risk commodity detection network; and *Protein* [25] is a biological network.

Baselines and Parameters. For a comprehensive evaluation, we include 20 competing methods in the experiments, which can be categorized into four types:

- MRGC: DMGI [60], MvAGC [45], MGDCR [53], BTGF [64], DuaLGR [47], BMGC [69], and DMG [54];
- Multi-view graph clustering: MCGC [58], MMGC [76], and LMVSC [33];
- Attributed graph clustering: Dink-Net [50], DMoN [81], S3GC [16], and S^{2}CAG [44];
- Attribute-less graph clustering: LeadEigvec [55], SpecClust [82], LabelProg [65], Louvain [4], node2vec [24], DeepWalk [63].

In attributed and attribute-less graph clustering baselines, we input the single-relational graph converted from the MRG with equal weights. For multi-view graph clustering methods, we use the same parameters as in FAAO to generate the feature matrix for each relation type. The number of iterations in DEMM, DEMM+, and DEMM-NA is fixed to 10 due to the rapid convergence. For a fair comparison, we run grid searches on the parameters and report the best clustering performance attained by each evaluated method. Table 4 summarizes the categories, complexities, objectives, and backbone models of the main competitors and our methods.

Evaluation Protocol. Following previous works [3, 7], we adopt three classic metrics *clustering accuracy* (ACC), *Normalized Mutual Information* (NMI), *Adjusted Rand Index* (ARI) to assess the quality of output clusters. All of them are calculated against the ground-truth cluster labels, and higher values indicate better quality. Particularly, ACC and NMI scores range from 0 to 1.0, whereas ARI falls in the range of $[-0.5, 1.0]$.

For the interest of space, we refer interested readers to Appendix D [43] for more details regarding datasets, baselines, parameters, and evaluation metrics.

Table 5. Clustering quality on small MRGs (best is highlighted in blue and best baseline underlined).

	Method	ACM			DBLP			ACM2			Yelp			IMDB		
		ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
w/o attributes	node2vec [24]	60.8	40.7	32.1	28.5	0.4	0.3	65.1	39.7	31.5	35.7	0.2	0.1	35.4	0.3	0.2
	DeepWalk [63]	61.4	34.9	31.6	75.9	60.4	55.7	56.5	21.1	15.9	51.7	14.4	13.5	36.2	0.2	0.1
	LeadEigvec [55]	35.2	0.7	0.0	79.3	66.1	65.7	49.5	0.2	−0.1	66.0	29.7	35.6	36.3	6.8	0.0
	LabelProg [65]	57.1	40.3	39.4	29.5	0.0	0.0	63.2	40.6	35.0	41.4	0.0	0.0	11.3	10.9	0.6
	Louvain [4]	55.3	40.1	36.4	79.3	67.6	66.1	60.7	39.3	34.8	60.6	36.6	40.9	13.3	4.9	1.1
	SpecClust [82]	35.3	0.4	0.0	91.6	76.7	80.3	70.3	51.1	41.0	65.2	37.5	41.4	37.9	0.3	0.0
	Improv.	+6.4	+4.0	+3.1	+0.6	+0.9	+1.7	+2.7	-9.8	+1.6	+2.5	-2.2	-3.7	+0.9	-10.5	-1.1
	DEMM-NA	68.0	44.7	42.5	92.2	77.6	82.0	73.0	41.3	42.6	68.5	35.3	37.7	38.8	0.4	0.0
w/ attributes	S3GC [16]	66.7	41.9	44.7	54.1	38	20.3	64.2	50.9	46.6	66.5	41.7	44.3	44.7	5.5	5.8
	DMoN [81]	70.7	45.6	49.5	80.6	54.6	60.2	69.7	38.7	37.6	75.3	51.5	52.2	49.4	12	9.7
	Dink-Net [50]	72.3	49.2	46.1	90.6	74.9	77.4	76.9	48.2	47.8	71.8	42.6	46.1	51.2	10.6	12.5
	S^{2}CAG [44]	88.6	65	69.5	83.1	58.1	63.2	80.9	55.2	55.2	87.0	59.9	64	53.9	18.0	18.9
	DMGI [60]	84.8	59.6	61.5	89.0	68.5	74.5	76.0	46.5	40.0	69.2	37.3	39.2	58.5	19.0	18.9
	LMVSC [33]	91.6	72.5	76.7	70.1	46.6	39.9	89.5	64.5	70.1	85.7	58.6	58.4	51.9	11.9	12.3
	MvAGC [45]	89.8	67.4	72.1	92.8	77.3	82.8	49.6	0.1	0.0	74.4	38.7	40.7	56.3	3.7	9.7
	MCGC [58]	91.5	71.3	76.3	92.9	77.5	83.0	70.1	45.8	36.5	56.6	20.9	8.8	61.8	11.5	18.1
	MMGC [76]	86.6	58.1	64.5	65.8	29.4	58.5	82.3	48.4	53.1	54.9	28.0	55.7	45.2	19.5	20.1
	MGDCR [53]	91.9	72.1	77.0	91.9	75.9	80.7	66.4	54.3	50.3	71.6	38.9	42.6	56.3	21.2	19.5
	BTGF [64]	93.2	75.8	80.9	83.1	62.4	59.7	88.3	64.2	67.6	73.2	44.2	45.4	66.8	22.6	25.7
	DualGR [47]	92.7	73.2	79.4	92.4	75.5	81.7	87.3	61.3	64.8	88.1	63.4	65.0	52.4	16.0	14.5
	DMG [54]	93.0	73.6	80.3	93.4	79.1	83.3	87.9	67.3	63.4	56.1	42.6	39.1	48.3	11.3	14.5
	BMGC [69]	93.0	75.7	80.4	93.4	78.3	84.0	91.3	72.0	74.2	91.5	71.7	73.8	51.0	14.3	14.4
	DEMM	93.2	75.6	80.7	92.6	76.5	82.1	90.8	70.1	73.2	91.7	69.7	74.7	68.5	25.0	28.1
	Improv.	0.0	-0.2	-0.2	-0.8	-2.6	-1.9	-0.5	-1.9	-1.0	+0.2	-2.0	+0.9	+1.7	+2.4	+2.4
	DEMM+	93.6	77.2	81.9	93.7	79.6	84.8	91.3	71.2	74.7	92.7	72.6	77.7	67.6	24.4	26.5
	Improv.	+0.4	+1.4	+1.0	+0.3	+0.5	+0.8	+0.0	-0.8	+0.5	+1.2	+1.3	+3.9	+0.8	+1.8	+0.8

Table 6. Clustering quality on large MRGs (best is highlighted in blue and best baseline underlined).

	Method	Protein			Amazon			MAG			OAG-ENG			OAG-CS			RCDD		
		ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
w/o attributes	node2vec [24]	27.1	4.9	2.7	57.2	3.0	−2.8	52.1	31.8	19.1	19.7	18.4	2.1	19.5	11.8	6.5	50.3	0.0	0.0
	DeepWalk [63]	33.5	4.7	2.5	60.2	1.5	2.0	49.9	35.6	30.1	9.1	3.0	1.1	18.3	12.2	6.1	54.7	0.0	0.2
	LeadEigvec [55]	32.4	0.3	−0.1	61.4	0.7	−1.9	27.1	2.1	0.0	7.3	14.8	0.2	9.8	1.7	0.0	–	–	–
	LabelProg [65]	31.5	5.5	0.3	91.4	1.2	4.2	15.7	24.5	12.6	11.4	36.8	5.5	17.0	19.4	5.3	4.3	4.9	0.1
	Louvain [4]	32.6	11.4	4.6	40.1	0.5	0.2	40.8	37.5	28.6	23.2	30.0	10.6	18.2	13.7	5.6	4.1	4.6	0.1
	SpecClust [82]	35.6	5.8	2.8	76.3	1.6	−5.6	27.2	0.1	0.0	7.5	0.6	0.0	9.8	0.1	0.0	–	–	–
	Improv.	-3.2	-9.5	-4.6	+0.2	+1.4	+11.2	+11.5	+24.8	+21.2	+2.8	-14.7	-0.3	+9.0	+18.9	+10.7	-2.6	-4.9	-0.2
	DEMM-NA	32.3	1.9	0.0	91.6	4.4	15.4	63.6	62.3	51.3	26.0	22.1	10.3	28.5	38.3	17.2	52.1	0.0	0.0
w/ attributes	S3GC [16]	37.7	15.5	9.7	87.3	10.3	2.6	64.5	61.5	51.5	5.6	3.7	3.4	35.4	38.5	21.4	–	–	–
	DMoN [81]	38.0	6.9	5.5	44.5	5.8	6.7	55.8	43.5	53.7	13.0	8.4	3.9	11.1	8.5	6.0	–	–	–
	Dink-Net [50]	33.1	8.7	4.5	76.8	2.3	2.1	64.8	61.7	49.6	–	–	–	–	–	–	–	–	–
	S^{2}CAG [44]	22.8	1.4	0.6	63.7	1.4	3.6	66.7	62.5	53.5	6.9	0.1	0.0	6.8	0.1	0.0	69.3	13.2	16.9
	DMGI [60]	23.4	2.1	0.9	56.0	3.8	1.3	29.1	0.7	1.0	8.2	1.8	0.6	9.8	4.7	1.3	67.7	2.6	4.2
	LMVSC [33]	29.6	3.7	0.0	63.7	0.0	0.0	41.7	19.5	13.1	18.6	16.4	9.5	19.3	14.2	5.7	69.9	1.6	1.9
	MvAGC [45]	35.1	11.5	8.8	75.1	8.8	14.6	54.0	32.7	27.7	12.2	5.4	2.0	10.9	4.4	1.6	75.1	4.2	11.3
	MGDCR [53]	29.1	0.3	0.0	81.6	2.6	0.0	61.4	54.5	44.0	25.7	21.0	13.8	25.3	25.9	16.8	–	–	–
	DMG [54]	32.2	0.2	0.1	90.9	1.4	7.6	55.3	43.1	34.9	25.2	24.5	10.9	25.9	28.3	13.9	–	–	–
	BMGC [69]	37.5	17.3	10.3	77.5	0.4	1.8	65.3	57.0	47.8	16.5	14.3	4.9	16.5	16.5	14.3	–	–	–
	DEMM	38.9	14.1	8.2	91.2	14.3	32.4	68.0	64.4	52.6	–	–	–	–	–	–	–	–	–
	Improv.	+0.9	-3.2	-2.1	+0.3	+4.0	+17.8	+1.3	+1.9	-1.1	–	–	–	–	–	–	–	–	–
	DEMM+	39.2	19.4	12.8	92.6	15.7	34.2	67.8	63.3	52.3	42.3	41.8	24.8	40.1	42.7	24.1	83.4	18.6	29.0
	Improv.	+1.2	+2.1	+2.5	+1.5	+5.4	+19.6	+1.1	+0.8	-1.4	+16.6	+17.3	+11.0	+4.7	+4.2	+2.7	+8.3	+5.4	+12.1

Table 7. Ablation studies on small MRGs.

Method	ACM			DBLP			ACM2			Yelp			IMDB		
	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
w/o $\{\omega_r\}_{r=1}^{R}$	92.5	73.1	78.8	93.3	78.2	83.7	90.8	68.6	73.5	92.4	71.6	76.8	66.8	23.5	25.6
w/o $\alpha \cdot \widehat{X}^{(L)}$	93.4	76.6	81.3	92.9	76.9	82.8	91.3	70.2	74.7	92.3	71.5	76.4	67.0	24.1	24.4
w/o $\mathcal{L}_{\text{reg}}$	92.9	75.8	80.1	91.6	73.5	79.7	90.0	69.4	71.0	92.0	71.6	75.5	67.4	24.3	26.2
DEMM+	93.6	77.2	81.9	93.7	79.6	84.8	91.3	71.2	74.7	92.7	72.6	77.7	67.6	24.4	26.5

Table 8. Ablation studies on large MRGs.

Method	OAG-ENG			MAG			OAG-CS		
	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
w/o $\{\omega_r\}_{r=1}^{R}$	36.1	38.6	20.7	65.7	62.5	50.9	36.1	35.9	20.2
w/o $\alpha \cdot \widehat{X}^{(L)}$	31.7	33.7	17.4	67.7	61.7	51.3	32.7	32.7	16.8
w/o $\mathcal{L}_{\text{reg}}$	24.4	22.8	10.1	67.8	63.4	52.4	20.2	15.5	5.5
DEMM+	42.3	41.8	24.8	67.8	63.3	52.3	40.1	42.7	24.1

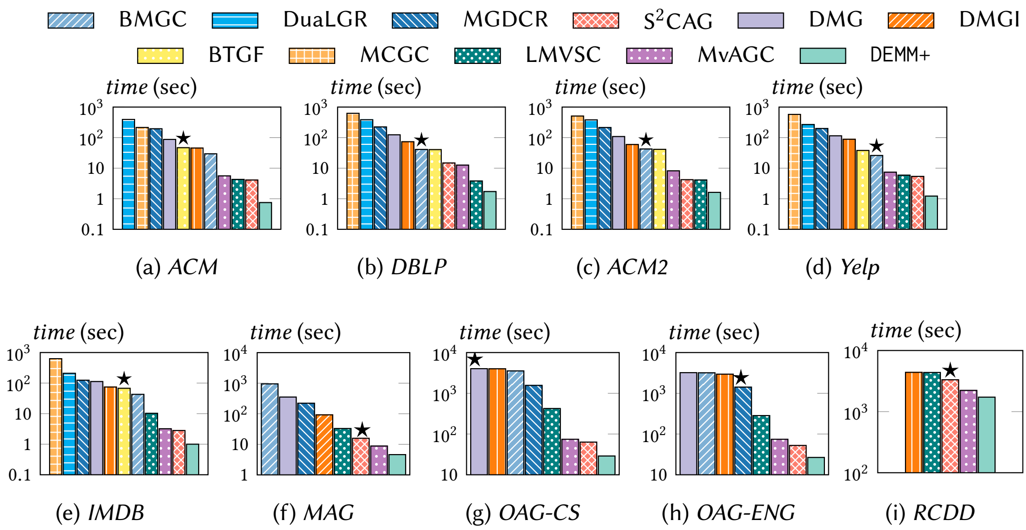


Fig. 9. Computational efficiency comparison. (best baselines in Tables 5 and 6 are marked with ⋆)

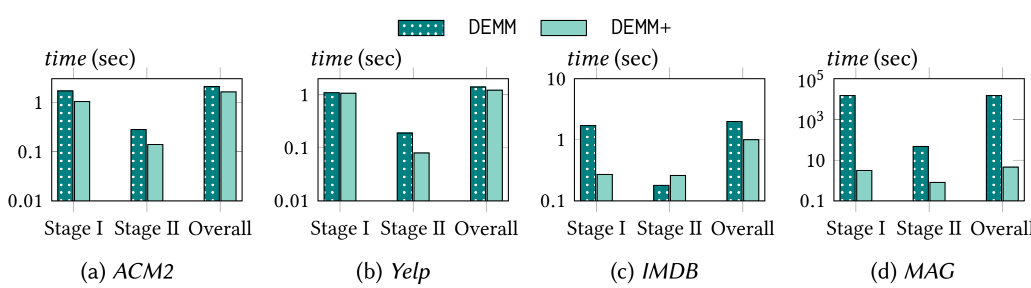
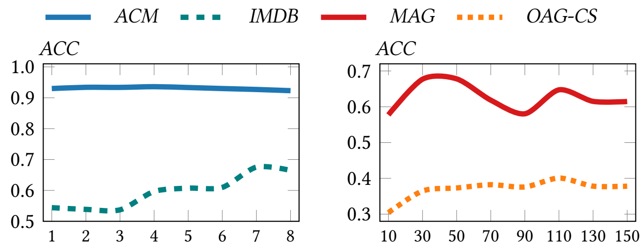


Fig. 10. Efficiency analysis of DEMM and DEMM+.

5.2 Clustering Quality Evaluation

This set of experiments studies the clustering quality attained by DEMM, DEMM+, DEMM-NA, and 20 competitors on all 9 MRG datasets. We exclude a method or omit its results if it fails to return valid outcomes within 2 days or runs beyond physical memory limits. Tables 5 and 6 report the ACC, NMI and ARI scores of all evaluated methods on small and large MRGs, respectively. Each table is divided into two parts, where the top part compares DEMM-NA against attribute-less graph clustering baselines by discarding the attributes of all datasets. The best results are highlighted in blue, and the best baselines are underlined.

From the tables, we can make the following observations. Firstly, DEMM+ consistently and considerably outperforms the best baselines in almost all cases. Particularly, on the large datasets, DEMM+ is able to achieve significant gains of 16.6%, 17.3%, and 11.0% in ACC, NMI, and ARI on *OAG-ENG* and remarkable improvements of 8.3%, 5.4%, and 12.1% on *RCDD*, respectively. On medium-sized datasets *Protein* and *Amazon*, DEMM+ also outperforms all baselines, yielding notable gains of 1.2%, 2.1%, 2.5%, and 1.7%, 5.4%, and 19.6% in ACC, NMI and ARI, respectively. In addition, it can be observed that DEMM is comparable to DEMM+ on most small MRGs but slightly better on *IMDB* and *MAG*. On larger datasets, DEMM fails to report results due to the quadratic complexity analyzed in Section 3.3. The superiority of DEMM and DEMM+ over MRGC, attributed graph clustering, and



(a) *ACM* and *IMDB* (b) *MAG* and *OAG-CS*

Fig. 11. Clustering accuracy when varying α

multi-view graph clustering baselines substantiates the effectiveness of our proposed two-stage objectives based on MRDE and DE in fusing multi-relational graph structures.

On attribute-less MRGs, the variant DEMM-NA of DEMM+ surpasses the best baselines in terms of ACC on all datasets except *RCDD*. Most notably, on *MAG*, DEMM-NA takes a lead of 11.5%, 24.8%, and 21.2% in ACC, NMI, and ARI. Notice that LabelProg and Louvain determine the number of clusters automatically, which accidentally leads to higher NMI and ARI values on *Yelp*, *IMDB*, and *OAG-ENG* compared to DEMM-NA.

5.3 Clustering Efficiency Evaluation

Figure 9 plots the runtime costs consumed by DEMM+ and 10 strong baselines in Tables 5 and 6. Note that the y-axis is in log-scale and the measurement unit for running time is seconds (sec). For fairness, we exclude the time costs needed for loading input data and outputting results in all methods, as well as their pre-training or pre-processing costs. The baselines with the best clustering quality are marked with ★. We exclude MCGC, MMGC, BTGF, and DuaLGR on large MRGs as they are unable to terminate with valid outcomes.

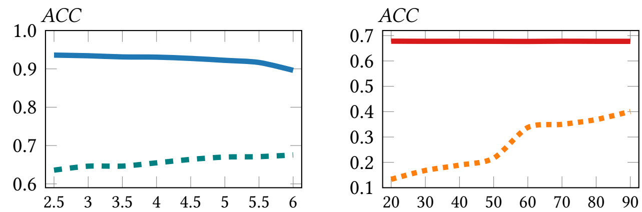
As evidenced in Figure 9, DEMM+ consistently demonstrates higher efficiency across all benchmark datasets. Compared to the best baselines in Tables 5 and 6, DEMM+ is able to achieve remarkable speedups of 62.5×, 23.9×, 25.6×, 21.4×, and 67.6× on small datasets *ACM*, *DBLP*, *ACM2*, *Yelp*, and *IMDB*, respectively. Notably, on large MRGs *OAG-CS* and *OAG-ENG* datasets with tens of millions of edges, the accelerations achieved by DEMM+ are over 139× and 53×, respectively. Even on the largest dataset *RCDD* with 11.9 million nodes and 0.78 billion edges, where most recent competitive MRGC approaches BTGF, DuaLGR, MGDCR, DMG, and BMGC fail, DEMM+ is still nearly 2× faster compared to the best viable baseline $\mathrm{S^2CAG}$, while producing significant improvements of 14.1%, 5.4%, and 12.1% in ACC, NMI, and ARI.

In Figure 10, we further corroborate the effectiveness of our proposed algorithms FAAO (Stage I) and SSKC (Stage II) in enhancing computational efficiency. As reported, DEMM+ accelerates the computation of both stages in DEMM, i.e., the construction of $\boldsymbol{H}$ and the generation of clusters. The acceleration is particularly pronounced on the large MRG dataset *MAG*, where DEMM+ obtains an overall speedup of 3,252× than DEMM. Moreover, DEMM cannot handle larger MRGs within 2 days, whereas DEMM+ finishes the clustering over *RCDD* using less than 30 minutes (see Figure 9).

5.4 Ablation Study

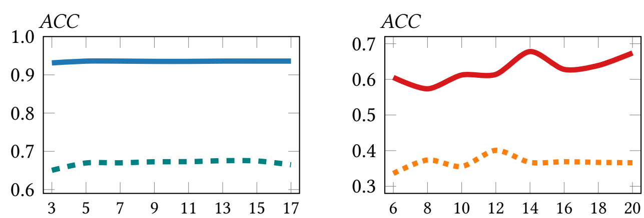
In this set of experiments, we empirically analyze the efficiency of three key ingredients in DEMM+, including the adjustments of RTWs $\{\omega_r\}_{r=1}^{R}$, the estimator $\alpha \cdot \widehat{\boldsymbol{X}}^{(L)}$ of the terms beyond L hops in $\boldsymbol{H}$ in Eq. (14), and the regularization term $\mathcal{L}_{\text{reg}}$ in Eq. (4).

According to Tables 7 and 8, compared to three ablated versions that remove the three ingredients, the complete DEMM+ always obtains conspicuously superior ACC, NMI, and ARI results on all MRGs.



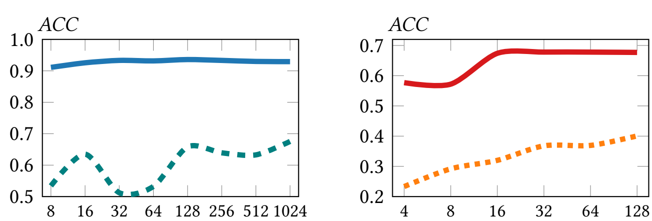
(a) *ACM* and *IMDB* (b) *MAG* and *OAG-CS*

Fig. 12. Clustering accuracy when varying β



(a) *ACM* and *IMDB* (b) *MAG* and *OAG-CS*

Fig. 13. Clustering accuracy when varying L



(a) *ACM* and *IMDB* (b) *MAG* and *OAG-CS*

Fig. 14. Clustering accuracy when varying d

Notably, on the *ACM2* and *DBLP*, the ACC scores increase by 1.3% and 2.1%, respectively, by including $\mathcal{L}_{\text{reg}}$ term, which indicates the significance of the regularization term in balanced fusion of multiplex graph structures. The improvements are more significant on *OAG-NEG* and *OAG-CS*, where substantial ACC improvements of 17.9% and 19.9% can be gained. On *MAG*, the conducive effects of the first and second ingredients are still noticeable, whereas the $\mathcal{L}_{\text{reg}}$ term contributes minimally.

5.5 Parameter Analysis

This section investigates the impact of parameters α, β, L, and d in DEMM+ on two small datasets *ACM* and *IMDB* and two large MRGs *MAG* and *OAG-CS*, respectively, by varying each parameter while fixing others. We report ACC scores only as NMI and ARI results are quantitatively similar, and thus, are deferred to Appendix D [43].

Varying α. Figure 11(a) shows the impact of varying α from 1 to 8 on the clustering performance on *ACM* and *IMDB*, while Figure 11(b) presents its effects on *MAG* and *OAG-CS* when varying it from 10 to 150. The results reveal that α has a negligible influence on *ACM*, but a profound impact on *IMDB*, *MAG*, and *OAG-CS*. Specifically, the ACC scores of *IMDB* improve monotonically with α until reaching its maximum value at $\alpha = 7$, whereas *MAG* and *OAG-CS* exhibit oscillatory behaviors, attaining peak values at $\alpha = 50$ and 110, respectively. Recall that in Eq. (4), α is the weight assigned to the MRDE term towards injecting graph topology information into the node feature vectors $\boldsymbol{H}$. Thus, a higher α indicates a larger portion of structural features encoded into $\boldsymbol{H}$. Generally, on the four datasets, a large α is preferred, implying the importance of graph structures in MRGC.

Varying β. Figure 12 displays the effects of the regularization weight β on ACC scores in Eq.(4). In Figure 12(a), where β varies within a short range from 2.5 to 6, the ACC scores of datasets *ACM* and *IMDB* exhibit divergent trends: the clustering performance of *ACM* deteriorates monotonically with increasing β, whereas that on *IMDB* grows progressively. In Figure 12(b), when varying β from 20 to 90, it can be observed that increasing β has little impact on *MAG*, but brings a considerable performance rise on *OAG-CS*. The differences can be ascribed to their unique structural disparities and volume differences between edges of different relation types.

Varying L. Figures 13(a) and 13(b) depict how the ACC scores change when L is varied from 3 to 17 on *ACM* and *IMDB*, and from 6 to 20 on *MAG* and *OAG-CS*. It can be seen that increasing L has little impact on ACC scores on *ACM* and *IMDB*. In comparison, on larger MRGs *MAG* and *OAG-CS*, the ACC scores first undergo upticks when increasing L to roughly 12 or 14, followed by a decrease or plateau. The results imply that estimating $\boldsymbol{H}$ as in Eq. (15) with up to a small number L hops of terms is sufficiently accurate, consistent with our empirical and theoretical analyses in Section 4.1.

Varying d. The parameter d represents the dimension of initial feature vectors $\boldsymbol{X}$, which are reduced from the input attribute matrix through a principal component analysis (Section 2.3). Figures 14(a) and 14(b) illustrate the changes in ACC scores on all four datasets when varying d in the ranges of $[8, 1024]$ and $[4, 128]$. For all datasets, we can see a clear rise in performance when enlarging d from 4 to 128, meaning more features are retained. However, the performance of DEMM+ starts to remain invariant or even undergoes minor drops when d exceeds 128, on either *ACM* and *IMDB* whose original attribute dimensions D are up to 2,000, or *MAG* and *OAG-CS* with $D = 128$ and 768. The drops are caused by data noise embodied in original attribute vectors, while the invariance can be explained by the well-known Johnson-Lindenstrauss lemma.

6 Related Work

Multi-relational Graph Clustering. MRGC focuses on generating consistent node representation by integrating consistency information across different relation types. Previous methods typically use adaptive weights to fuse each relation together and construct a unified graph [27, 45, 58], SwMC [57] and MvAGC [45] are the representative methods with a self-adjusting weight computation algorithm. To further extract shared patterns from MRG, numerous methods have incorporated consistency information during the fusion of different relation types. DuaLGR [47] proposed a method where soft labels derived from consistency information are used to refine the graphs of each relation type before fusion. DMGI [60] reconstructs MRG by maximizing the mutual information across relation types. However, these methods cannot fully exploit the dependencies between different relation types and the feature matrices, resulting in their underperformance in MRGs.

Recently, many approaches generate node embeddings for each relation type individually and identify cross-relational consistencies from different relational graphs [48, 58, 59, 62, 70, 91]. BTGF [64] designs filters with non-shared parameters for each relation type to obtain node embeddings from diverse perspectives. DMG [54] disentangles consistent and redundant information from the features of different relations. BMGC [69] introduces imbalanced multiview learning to refine embeddings derived from less important relation types. Nevertheless, these methods overlook the complementary information introduced by fusing MRGs, thus hindering the exploitation of MRGs.

Attributed Graph Clustering. Attributed graph clustering (AGC) has been extensively studied nowadays [6, 34, 39, 41, 92, 95, 96, 106]. Most recent research has focused on integrating graph topology with node attributes to produce cohesive embeddings [1, 12, 40, 94, 109], which are then clustered by using classical clustering methods to obtain the final results. With the widespread adoption of deep learning, methods that leverage deep learning models like GNNs [67] to learn consistent node representations have gained popularity [5, 15, 32, 51, 52], DMoN [81], Dink-Net [50], and S3GC [16] are the representative methods among them. H-GCN [29] introduces graph coarsening to capture long-range information, thereby addressing the potential overfitting caused by increasing the depth of GNN models. To fully integrate topological and attribute information of graphs, attention mechanisms [84, 90, 105] and graph contrastive learning [27, 97, 103] have also been widely employed in this process. Some recent approaches [21, 44] integrate subspace clustering with spectral clustering techniques [56]. However, AGC fails to account for the varying significance of distinct relations, rendering it inapplicable to MRGs.

Multi-View Graph Clustering. Multi-view clustering is to group data with heterogeneous feature representations. Due to dimensional differences across vertices, directly linearly combining features from different views is not feasible. Early graph-based approaches rely on constructing similarity matrices followed by spectral clustering [74, 75, 80, 107], LMVSC [33] enhances scalability by introducing anchor graphs to replace fully connective graph. GTLEC [9] and CGL [42] enhance multi-view consistency through optimized affinity matrix construction. These methods often incur significant memory consumption for similarity matrix construction. To this end, UOMvSC [77] eliminates the need for explicit similarity matrix construction. Matrix factorization-based methods extract cross-view shared information through matrix decomposition and integrate it into a unified representation [8, 14, 31, 86, 88, 89].

Recent deep learning-based approaches define and optimize specific metrics such as MCGC [58] and MAGCN [10]. Despite effectively integrating cross-dimensional features, they struggle to generalize to MRG due to incompatible relation modeling.

7 Conclusion

This paper proposes two effective methods, DEMM and DEMM+, for MRGC. DEMM achieves remarkable clustering performance on MRGs, via our innovative two-stage optimization objectives formulated upon the MRDE of MRGs and DE of affinity graphs. Based thereon, we develop DEMM+, which significantly advances the efficiency and scalability of DEMM via two elaborate secondary algorithms FAAO and SSKC containing several non-trivial optimization techniques. Our extensive evaluations experimentally manifest the consistent superiority of DEMM+ over a wide range of baselines in clustering quality and empirical efficiency. However, the proposed techniques are mainly designed for static MRGs, which struggle to cope with dynamic MRGs with frequent updates. In the future, our work can be extended to dynamic MRGs by devising sampling and incremental techniques for structural changes (e.g., node/edge insertions/deletions). Moreover, the notion of MRDE can be further generalized to heterogeneous graphs with multiple node types, enabling broader applications in real-world scenarios.

Acknowledgments

This work is supported by the Hong Kong RGC ECS grant (No. 22202623), NSFC No. 62302414, and the Huawei gift fund.

References

[1] Esra Akbas and Peixiang Zhao. 2017. Attributed Graph Clustering: an Attribute-aware Graph Embedding Approach. *ASONAM* (2017).

[2] Arian Ashourvan, Qawi K Telesford, Timothy Verstynen, Jean M Vettel, and Danielle S Bassett. 2019. Multi-scale detection of hierarchical community architecture in structural and functional brain networks. *Plos one* (2019), e0215520.

[3] Aritra Bhowmick, Mert Kosan, Zexi Huang, Ambuj Singh, and Sourav Medya. 2024. DGCLUSTER: A Neural Framework for Attributed Graph Clustering via Modularity Maximization. In *AAAI*, Vol. 38. 11069–11077.

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[5] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. 2020. Structural Deep Clustering Network. In *Proceedings of The Web Conference 2020*. Association for Computing Machinery, 1400–1410.

[6] Cécile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenkova. 2015. Clustering attributed graphs: models, measures and methods. *Network Science* 3, 3 (2015), 408–444.

[7] Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. 2022. Efficient Deep Embedded Subspace Clustering. *CVPR* (2022), 21–30.

[8] Kamalika Chaudhuri, Sham M. Kakade, Karen Livescu, and Karthik Sridharan. 2009. Multi-view clustering via canonical correlation analysis. In *International Conference on Machine Learning*.

[9] Mansheng Chen, Jia-Qi Lin, Changdong Wang, Wu-Dong Xi, and Dong Huang. 2023. On Regularizing Multiple Clusterings for Ensemble Clustering by Graph Tensor Learning. *MM* (2023).

[10] Jiafeng Cheng, Qianqian Wang, Zhiqiang Tao, Deyan Xie, and Quanxue Gao. 2020. Multi-View Attribute Graph Convolution Networks for Clustering. In *International Joint Conference on Artificial Intelligence*.

[11] Kenneth L Clarkson and David P Woodruff. 2017. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)* 63, 6 (2017), 1–45.

[12] David Combe, Christine Largeron, Mathias Géry, and Elöd Egyed-Zsigmond. 2015. I-Louvain: An Attributed Graph Clustering Method. In *International Symposium on Intelligent Data Analysis*.

[13] J. J. Crofts, M. Forrester, S. Coombes, and R. D. O'Dea. 2022. Structure-Function Clustering in Weighted Brain Networks. *Scientific Reports* 12 (2022), 16793.

[14] Chenhang Cui, Yazhou Ren, Jingyu Pu, Xiaorong Pu, and Lifang He. 2023. Deep multi-view subspace clustering with anchor graph. In *IJCAI*. 3577–3585.

[15] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. 2020. Adaptive Graph Encoder for Attributed Graph Embedding. *KDD* (2020).

[16] Fnu Devvrit, Aditya Sinha, Inderjit Dhillon, and Prateek Jain. 2022. S3GC: scalable self-supervised graph clustering. *NeurIPS* 35 (2022), 3248–3261.

[17] Ouxia Du and Ya Li. 2022. Academic Collaborator Recommendation Based on Attributed Network Embedding. *J. Data Inf. Sci.* 7, 1 (2022), 37–56.

[18] Ky Fan. 1949. On a theorem of Weyl concerning eigenvalues of linear transformations I. *PNAS* 35, 11 (1949), 652–655.

[19] Shaohua Fan, Xiao Wang, Chuan Shi, Emiao Lu, Ken Lin, and Bai Wang. 2020. One2Multi Graph Autoencoder for Multi-view Graph Clustering. *WWW* (2020).

[20] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2022. A Graph Neural Network Framework for Social Recommendations. *TKDE* 34, 5 (2022), 2033–2047.

[21] Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. 2023. Scalable Attributed-Graph Subspace Clustering. In *AAAI*, Vol. 37.

[22] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. *WWW* (2020).

[23] Olivier Goldschmidt and Dorit S Hochbaum. 1988. Polynomial algorithm for the k-cut problem. In *FOCS*. 444–451.

[24] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. 855–864.

[25] Yaowen Gu, Si Zheng, Qijin Yin, Rui Jiang, and Jiao Li. 2022. REDDA: Integrating multiple biological relations to heterogeneous graph neural network for drug-disease association prediction. *Computers in Biology and Medicine* 150 (2022), 106127.

[26] Soumaya Guesmi, Chiraz Trabelsi, and Chiraz Latiri. 2019. Community detection in multi-relational social networks based on relational concept analysis. *PCS* 159 (2019), 291–300.

[27] Kaveh Hassani and Amir Hosein Khas Ahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *ICML*.

[28] Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

[29] Fenyu Hu, Yanqiao Zhu, Shu Wu, Liang Wang, and Tieniu Tan. 2019. Hierarchical graph convolutional networks for semi-supervised node classification. In *IJCAI*.

[30] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *ArXiv* abs/2005.00687 (2020).
[31] Shudong Huang, Yixi Liu, Ivor Wai-Hung Tsang, Zenglin Xu, and Jiancheng Lv. 2023. Multi-View Subspace Clustering by Joint Measuring of Consistency and Diversity. *TKDE* 35 (2023), 8270–8281.
[32] Guangyu Huo, Yong Zhang, Junbin Gao, Boyue Wang, Yongli Hu, and Baocai Yin. 2021. CaEGCN: Cross-Attention Fusion Based Enhanced Graph Convolutional Network for Clustering. *IEEE Transactions on Knowledge and Data Engineering* 35 (2021), 3471–3483.
[33] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. 2020. Large-scale multi-view subspace clustering in linear time. In *AAAI*, Vol. 34. 4412–4419.
[34] Xinying Lai, Dingming Wu, Christian S Jensen, and Kezhong Lu. 2023. A Re-evaluation of Deep Learning Methods for Attributed Graph Clustering. In *CIKM*. 1168–1177.
[35] Richard B Lehoucq and Danny C Sorensen. 1996. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl.* 17, 4 (1996), 789–821.
[36] David A Levin and Yuval Peres. 2017. *Markov chains and mixing times*. Vol. 107. American Mathematical Soc.
[37] Mingqi Li, Wenming Ma, and Zihao Chu. 2024. KGIE: Knowledge graph convolutional network for recommender system with interactive embedding. *Knowledge-Based Systems* 295 (2024), 111813.
[38] Rui Li, Xin Yuan, Mohsen Radfar, Peter Marendy, Wei Ni, Terrence J O'Brien, and Pablo M Casillas-Espinosa. 2021. Graph signal processing, graph neural network and graph learning on biological data: a systematic review. *IEEE Reviews in Biomedical Engineering* 16 (2021), 109–135.
[39] Yiran Li, Gongyao Guo, Jieming Shi, Renchi Yang, Shiqi Shen, Qing Li, and Jun Luo. 2024. A versatile framework for attributed network clustering via K-nearest neighbor augmentation. *VLDBJ* (2024), 1–31.
[40] Ye Li, Chaofeng Sha, Xin Huang, and Yanchun Zhang. 2018. Community Detection in Attributed Graphs: An Embedding Approach. In *AAAI*.
[41] Yiran Li, Renchi Yang, and Jieming Shi. 2023. Efficient and effective attributed hypergraph clustering via k-nearest neighbor augmentation. *SIGMOD* 1, 2 (2023), 1–23.
[42] Zhenglai Li, Chang Tang, Xinwang Liu, Xiao Zheng, Guanghui Yue, Wei Zhang, and En Zhu. 2021. Consensus Graph Learning for Multi-View Clustering. *IEEE Transactions on Multimedia* 24 (2021), 2461–2472.
[43] Xiaoyang Lin, Runhao Jiang, and Renchi Yang. 2025. Effective Clustering for Large Multi-Relational Graphs. arXiv:2508.17388 [cs.LG] https://arxiv.org/abs/2508.17388
[44] Xiaoyang Lin, Renchi Yang, Haoran Zheng, and Xiangyu Ke. 2025. Spectral Subspace Clustering for Attributed Graphs. In *KDD*. 789–799.
[45] Zhiping Lin and Zhao Kang. 2021. Graph Filter-based Multi-view Attributed Graph Clustering. In *IJCAI*.
[46] Zizheng Lin, Haowen Ke, Ngo-Yin Wong, Jiaxin Bai, Yangqiu Song, Huan Zhao, and Junpeng Ye. 2021. Multi-relational graph based heterogeneous multi-task learning in community question answering. In *CIKM*. 1038–1047.
[47] Yawen Ling, Jianpeng Chen, Yazhou Ren, Xiaorong Pu, Jie Xu, Xiaofeng Zhu, and Lifang He. 2023. Dual Label-Guided Graph Refinement for Multi-View Graph Clustering. In *AAAI*.
[48] Liang Liu, Zhao Kang, Ling Tian, Wenbo Xu, and Xixu He. 2021. Multilayer Graph Contrastive Clustering Network. *ArXiv* abs/2112.14021 (2021).
[49] Weifeng Liu, Jose C Principe, and Simon Haykin. 2011. *Kernel adaptive filtering: a comprehensive introduction*. John Wiley & Sons.
[50] Yue Liu, Ke Liang, Jun Xia, Sihang Zhou, Xihong Yang, Xinwang Liu, and Stan Z Li. 2023. Dink-net: Neural clustering on large graphs. In *International Conference on Machine Learning*. PMLR, 21794–21812.
[51] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. 2022. Deep Graph Clustering via Dual Correlation Reduction. In *AAAI*, Vol. 36. 7603–7611.
[52] Yue Liu, Jun Xia, Sihang Zhou, Xihong Yang, Ke Liang, Chenchen Fan, Yan Zhuang, Stan Z Li, Xinwang Liu, and Kunlun He. 2022. A Survey of Deep Graph Clustering: Taxonomy, Challenge, Application, and Open Resource. *arXiv preprint arXiv:2211.12875* (2022).
[53] Yujie Mo, Yuhuan Chen, Yajie Lei, Liang Peng, Xiaoshuang Shi, Changan Yuan, and Xiaofeng Zhu. 2023. Multiplex Graph Representation Learning Via Dual Correlation Reduction. *TKDE* 35 (2023), 12814–12827.
[54] Yujie Mo, Yajie Lei, Jialie Shen, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. 2023. Disentangled Multiplex Graph Representation Learning. In *International Conference on Machine Learning*.
[55] Mark EJ Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 74, 3 (2006), 036104.
[56] A. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an algorithm. In *Neural Information Processing Systems*.
[57] Feiping Nie, Jing Li, and Xuelong Li. 2017. Self-weighted Multiview Clustering with Multiple Graphs. In *IJCAI*.
[58] Erlin Pan and Zhao Kang. 2021. Multi-view Contrastive Graph Clustering. In *NIPS*.

[59] Erlin Pan and Zhao Kang. 2023. Beyond Homophily: Reconstructing Structure for Graph-agnostic Clustering. In *ICML*.
[60] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. 2019. Unsupervised Attributed Multiplex Network Embedding. In *AAAI*.
[61] Hao Peng, Ruitong Zhang, Yingtong Dou, Renyu Yang, Jingyi Zhang, and Philip S. Yu. 2021. Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks. *ACM Trans. Inf. Syst.*, Article 69 (Dec. 2021), 46 pages.
[62] Liang Peng, Xin Wang, and Xiaofeng Zhu. 2023. Unsupervised Multiplex Graph learning with Complementary and Consistent Information. *MM* (2023).
[63] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*. 701–710.
[64] Xiaowei Qian, Bingheng Li, and Zhao Kang. 2023. Upper Bounding Barlow Twins: A Novel Filter for Multi-Relational Clustering. *ArXiv* abs/2312.14066 (2023).
[65] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 76, 3 (2007), 036106.
[66] Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems* 20 (2007).
[67] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20 (2009), 61–80.
[68] John Shawe-Taylor and Nello Cristianini. 2004. *Kernel methods for pattern analysis.* Cambridge university press.
[69] Zhixiang Shen, Haolan He, and Zhao Kang. 2024. Balanced Multi-Relational Graph Clustering. *ArXiv* abs/2407.16863 (2024).
[70] Zhixiang Shen, Shuo Wang, and Zhao Kang. 2024. Beyond Redundancy: Information-aware Unsupervised Multiplex Graph Structure Learning. *ArXiv* abs/2409.17386 (2024).
[71] Chuan Shi, Yuanfu Lu, Linmei Hu, Zhiyuan Liu, and Huadong Ma. 2022. RHINE: Relation Structure-Aware Heterogeneous Information Network Embedding. *IEEE Transactions on Knowledge and Data Engineering* 34 (2022), 433–447.
[72] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *TPAMI* 22, 8 (2000), 888–905.
[73] Richard Sinkhorn and Paul Knopp. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.* 21, 2 (1967), 343–348.
[74] Jeong-Woo Son, Junekey Jeon, Sang-Yun Lee, and Sun-Joong Kim. 2016. Adaptive spectral co-clustering for multiview data. *2016 18th International Conference on Advanced Communication Technology (ICACT)* (2016), 447–450.
[75] Alexander Strehl and Joydeep Ghosh. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3 (2003), 583–617.
[76] Yuze Tan, Yixi Liu, Hongjie Wu, Jiancheng Lv, and Shudong Huang. 2023. Metric multi-view graph clustering. In *AAAI*, Vol. 37. 9962–9970.
[77] Chang Tang, Zhenglai Li, J. Wang, Xinwang Liu, Wei Zhang, and En Zhu. 2023. Unified One-Step Multi-View Spectral Clustering. *IEEE Transactions on Knowledge and Data Engineering* 35 (2023), 6449–6460.
[78] Jiangnan Tang, Huanhuan Gu, Darko B Vuković, Guandong Xu, Youquan Wang, Haicheng Tao, and Jie Cao. 2025. Fraud detection in multi-relation graph: Contrastive Learning on Feature and Structural Levels. *Neurocomputing* (2025), 130063.
[79] Lei Tang, Xufei Wang, and Huan Liu. 2012. Community detection via heterogeneous interaction analysis. *DMKD* (2012), 1–33.
[80] Wei Tang, Zhengdong Lu, and Inderjit S. Dhillon. 2009. Clustering with Multiple Graphs. *ICDM* (2009), 1016–1021.
[81] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. 2023. Graph clustering with graph neural networks. *Journal of Machine Learning Research* 24, 127 (2023), 1–21.
[82] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17 (2007), 395–416.
[83] Dorothea Wagner and Frank Wagner. 1993. Between min cut and graph bisection. In *Mathematical Foundations of Computer Science 1993: 18th International Symposium, MFCS'93 Gdańsk, Poland, August 30–September 3, 1993 Proceedings 18*. Springer, 744–750.
[84] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Attributed graph clustering: a deep attentional embedding approach. In *IJCAI*. 3670–3676.
[85] Chenxu Wang, Mengqin Wang, Xiaoguang Wang, Luyue Zhang, and Yi Long. 2024. Multi-Relational Graph Representation Learning for Financial Statement Fraud Detection. *Big Data Mining and Analytics* 7, 3 (2024), 920–941.
[86] Xiaobo Wang, Xiaojie Guo, Zhen Lei, Changqing Zhang, and S. Li. 2017. Exclusivity-Consistency Regularized Multi-view Subspace Clustering. *CVPR* (2017), 1–9.

[87] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, Philip S. Yu, and Yanfang Ye. 2019. Heterogeneous Graph Attention Network. *The World Wide Web Conference* (2019).
[88] Yunchao Wei, Yao Zhao, Zhenfeng Zhu, Yanhui Xiao, and Shikui Wei. 2014. Learning a mid-level feature space for cross-media regularization. *ICME* (2014), 1–6.
[89] Martha White, Yaoliang Yu, Xinhua Zhang, and Dale Schuurmans. 2012. Convex Multi-view Subspace Learning. In *Neural Information Processing Systems*.
[90] Hui Xia, Shu shu Shao, Chun qiang Hu, Rui Zhang, Tie Qiu, and Fu Xiao. 2023. Robust Clustering Model Based on Attention Mechanism and Graph Convolutional Network. *TKDE* 35 (2023), 5203–5215.
[91] Wei Xia, Sen Wang, Ming Yang, Quanxue Gao, Jungong Han, and Xinbo Gao. 2021. Multi-view graph embedding clustering network: Joint self-supervision and block diagonal representation. *Neural networks : the official journal of the International Neural Network Society* 145 (2021), 1–9.
[92] Kun Xie, Renchi Yang, and Sibo Wang. 2025. Diffusion-based Graph-agnostic Clustering. In *TheWebConf.* 1353–1364.
[93] Renchi Yang and Jieming Shi. 2024. Efficient High-Quality Clustering for Large Bipartite Graphs. *SIGMOD* 2, 1 (2024), 1–27.
[94] Renchi Yang, Jieming Shi, Xiaokui Xiao, Yin Yang, Sourav S Bhowmick, and Juncheng Liu. 2023. PANE: scalable and effective attributed network embedding. *VLDBJ* 32, 6 (2023), 1237–1262.
[95] Renchi Yang, Jieming Shi, Yin Yang, Keke Huang, Shiqi Zhang, and Xiaokui Xiao. 2021. Effective and scalable clustering on massive attributed graphs. In *TheWebConf.* 3675–3687.
[96] Renchi Yang, Yidu Wu, Xiaoyang Lin, Qichen Wang, Tsz Nam Chan, and Jieming Shi. 2024. Effective Clustering on Large Attributed Bipartite Graphs. In *KDD*. 3782–3793.
[97] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. 2023. Cluster-guided Contrastive Graph Clustering Network. *ArXiv* abs/2301.01098 (2023).
[98] Zaihan Yang, Dawei Yin, and Brian D Davison. 2014. Recommendation in academia: A joint multi-relational model. In *ASONAM 2014*. IEEE, 566–571.
[99] Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. 2016. Orthogonal random features. *Advances in neural information processing systems* 29 (2016).
[100] Stella X. Yu and Jianbo Shi. 2003. Multiclass Spectral Clustering. In *ICCV*. 313–319.
[101] Xiang Yue, Zhen Wang, Jingong Huang, Srinivasan Parthasarathy, Soheil Moosavinasab, Yungui Huang, Simon M Lin, Wen Zhang, Ping Zhang, and Huan Sun. 2019. Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics* 36, 4 (10 2019), 1241–1251.
[102] Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, and Kuansan Wang. 2019. OAG: Toward Linking Large-scale Heterogeneous Entity Graphs. *KDD* (2019).
[103] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. 2021. Graph Debiased Contrastive Learning with Joint Representation Clustering. In *International Joint Conference on Artificial Intelligence*.
[104] Jianan Zhao, Xiao Wang, Chuan Shi, Zekuan Liu, and Yanfang Ye. 2020. Network Schema Preserving Heterogeneous Information Network Embedding.. In *IJCAI*. 1366–1372. Scheduled for July 2020, Yokohama, Japan, postponed due to the Corona pandemic..
[105] Qiqi Zhao, Huifang Ma, Lijun Guo, and Zhixin Li. 2022. Hierarchical attention network for attributed community detection of joint representation. *Neural Computing and Applications* 34 (2022), 5587 – 5601.
[106] Haoran Zheng, Renchi Yang, and Jianliang Xu. 2025. Adaptive Local Clustering over Attributed Graphs. In *ICDE*. IEEE Computer Society, 2052–2065.
[107] Dengyong Zhou and Christopher J. C. Burges. 2007. Spectral clustering and transductive learning with multiple views. In *ICML*.
[108] Dengyong Zhou and Bernhard Schölkopf. 2005. Regularization on discrete spaces. In *Joint Pattern Recognition Symposium*. Springer, 361–368.
[109] Hao Zhu and Piotr Koniusz. 2021. Simple Spectral Graph Convolution. In *ICLR*.
[110] Shuman Zhuang, Sujia Huang, Wei Huang, Yuhong Chen, Zhihao Wu, and Ximeng Liu. 2024. Enhancing Multi-view Graph Neural Network with Cross-view Confluent Message Passing. In *MM*.

Received April 2025; revised July 2025; accepted August 2025