EMGC^{2}F: Efficient Multi-view Graph Clustering with Comprehensive Fusion

Danyang Wu[1,2*], **Jitao Lu**[1,2*], **Feiping Nie**[1,2†], **Rong Wang**[2] and **Yuan Yuan**[2]
[1]School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, P. R. China.
[2]School of Artificial Intelligence, Optics and Electronics (iOPEN), and the Key Laboratory of Intelligent Interaction and Applications (Ministry of Industry and Information Technology), Northwestern Polytechnical University, Xi'an 710072, P. R. China.
danyangwu41x@mail.nwpu.edu.cn, dianlujitao@gmail.com, feipingnie@gmail.com, wangrong07@tsinghua.org.cn, y.yuan1.ieee@qq.com

Abstract

This paper proposes an Efficient Multi-view Graph Clustering with Comprehensive Fusion (EMGC^{2}F) model and a corresponding efficient optimization algorithm to address multi-view graph clustering tasks effectively and efficiently. Compared to existing works, our proposals have the following highlights: 1) EMGC^{2}F directly finds a consistent cluster indicator matrix with a Super Nodes Similarity Minimization module from multiple views, which avoids time-consuming spectral decomposition in previous works. 2) EMGC^{2}F comprehensively mines information from multiple views. More formally, it captures the consistency of multiple views via a Cross-view Nearest Neighbors Voting (CN^{2}V) mechanism, meanwhile capturing the importance of multiple views via an adaptive weighted-learning mechanism. 3) EMGC^{2}F is a parameter-free model and the time complexity of the proposed algorithm is far less than existing works, demonstrating the practicability. Empirical results on several benchmark datasets demonstrate that our proposals outperform SOTA competitors both in effectiveness and efficiency.

1 Introduction

Multi-view clustering is a fundamental task in machine learning that aims to seek for a consistent clustering decision via fusing the information of multiple views [Chao *et al.*, 2021; Xu *et al.*, 2013]. As an important branch of multi-view clustering, multi-view graph clustering (MGC) focuses on seeking for clustering decision from linkages of multiple graphs, and has become more popular in recent years. Existing multi-view graph clustering models can be briefly categorized into three types based on the learning scheme. The first type of models (two-stage) aim to learn a consistent spectral embedding from multi-view similarity matrices or spectral embeddings [Belkin and Niyogi, 2003], then the clustering results can be obtained via K-means or other clustering procedure based on the learned spectral embedding. To be specific, the learning schemes mainly include kernel approximation [Kumar *et al.*, 2011; Dong *et al.*, 2021; Wu *et al.*, 2021b], manifold approximation [Wu *et al.*, 2020; Dong *et al.*, 2014] and graph reconstruction [Nie *et al.*, 2016; Nie *et al.*, 2017a]. The second type of models (one-stage) aim to learn a block-diagonal similarity matrix from multi-view similarity matrices, in which the learned block-diagonal similarity matrix directly indicates the clustering result. Existing learning schemes of this type mainly include Frobenius norm minimization [Nie *et al.*, 2017b; Li *et al.*, 2020; Huang *et al.*, 2021] and inner product minimization [Zhan *et al.*, 2018]. The third type of models (one-stage) aim to learn the consistent clustering (soft) indicator matrix from multi-view spectral embeddings, and the learning schemes include factorization approximation [Hu *et al.*, 2020; Liu *et al.*, 2021], adaptive Procrustes [Nie *et al.*, 2018], etc.

However, the three types of existing models cannot avoid spectral decomposition operator that needs $O(n^3)$ (n is the number of samples) time complexity for accurate calculations, which is too time-consuming in practice. In this paper, we consider breaking out the traditional spectral-based form and handling multi-view graph clustering comprehensively to enhance both its effectiveness and efficiency as illustrated in Figure 1. To be specific, we propose an Efficient Multi-view Graph Clustering with Comprehensive Fusion (EMGC^{2}F) model to directly optimize cluster indicator via integrating information from multiple views. EMGC^{2}F first adopts a super-nodes similarity minimization strategy to establish a novel cluster indicator learning paradigm without spectral decomposition. Afterwards, EMGC^{2}F considers two multi-view fusion mechanisms as follows. 1) To capture the consistency among multiple views, EMGC^{2}F considers confirming the credible nearest neighbors relationship for each node via voting from multiple views, and constrain each pair of nodes to the same cluster if they are credible nearest neighbors. 2) To capture the importance of different views, an adaptive-weighted learning strategy is introduced. Besides, a vital transformation is proposed to transform EMGC^{2}F into a two-stage scaled form, *i.e.*, graph coarsening and clustering on coarsened graph, which provides the possibility to perform multi-view graph clustering efficiently. To handle the calculation and optimization problem of the two-stage EMGC^{2}F form, an efficient alternative algorithm with $O(|E| + m^2)(|E| < n^2, m < n)$ time com-

* Equal contribution † Corresponding author

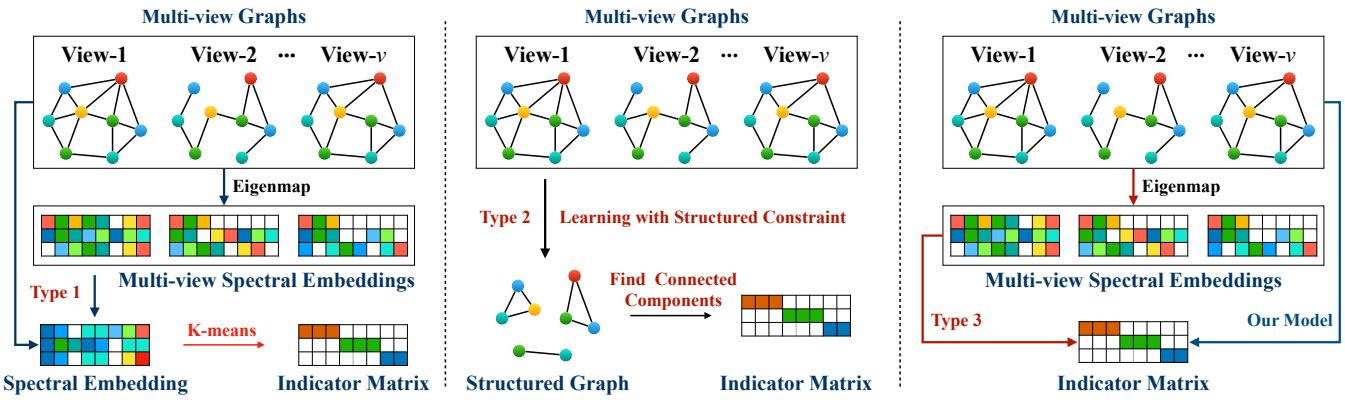


Figure 1: EMGC^{2}F and three different types of existing multi-view graph clustering models. Left: two-stage models that learn a consistent spectral embedding from multiple views and obtain clustering results via K-means. Middle: one-stage models that learn a block-diagonal similarity matrix from multi-view similarity matrices and obtain clustering result from its connectivity. Right: one-stage models that learn consistent clustering (soft) indicator matrix from multi-view spectral embeddings, and our proposed EMGC^{2}F that directly learns clustering labels from input multi-view graphs.

plexity is proposed, where $|E|$ is the number of edges and m is the number of nodes of the coarsened graph. More importantly, EMGC^{2}F is a parameter-free model, and the succinct parameter-free form guarantees the availability in practice. We conduct extensive experiments on several multi-view benchmark datasets, and the experimental results demonstrate that EMGC^{2}F outperforms SOTA competitors on MGC tasks in terms of both effectiveness and efficiency. In a word, it is worthwhile summarizing the main highlights of this paper as the following two sides:

- **Effectiveness:** The proposed EMGC^{2}F model is a novel and systematic model to handle MGC tasks. 1) The designed super-nodes similarity minimization strategy contains rich theoretical meaning with several fields. 2) The designed Cross-view Nearest Neighbors Voting (CN^{2}V) and adaptive-weighted learning mechanism comprehensively mine the multi-view information. In experiments, EMGC^{2}F outperforms SOTA competitors on MGC tasks.
- **Efficiency:** The objective problem of EMGC^{2}F can be equivalently transformed into a two-stage scaled form which can be efficiently solved within $O(|E| + m^2)(|E| < n^2, m < n)$ via the proposed alternative algorithm. Its time complexity is far less than previous works based on $O(n^3)$ spectral decomposition. In experiments, the efficiency of EMGC^{2}F with proposed algorithm is ahead of current competitors on MGC tasks.

Notations. For a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, $x_{i,j}$ denotes the $\langle i,j\rangle$-th element, $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{x}^i \in \mathbb{R}^{1\times d}$ denote the i-th column and i-th row respectively; For a vector $\mathbf{x} \in \mathbb{R}^n$, x_i denotes the i-th element, $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$ and $\|\mathbf{x}\|_0$ denotes the number of non-zero elements in $\mathbf{x}$. For a set Q, $|Q|$ denotes its cardinality; Suppose $Q_k \subset Q$ is a subset of Q, we denote $\bar{Q}_k$ as the complement of Q_k in Q. Besides, in multi-view graph clustering, we denote the undirected weighted graph of s-th view by $\boldsymbol{\mathcal{W}}^{(s)} = (V, E^{(s)}, \mathbf{W}^{(s)})$, where V is the set of nodes, $E^{(s)}$ is the set of edges, $\mathbf{W}^{(s)} \in \mathbb{R}^{n\times n}$ is the similarity matrix; $\mathbf{D}^{(s)} \in \mathbb{R}^{n\times n}$ (diagonal) is the degree matrix of $\mathbf{W}^{(s)}$, where $d_{i,i}^{(s)} = \sum_{j=1}^n w_{i,j}^{(s)}$; $\mathbf{L}^{(s)} = \mathbf{D}^{(s)} - \mathbf{W}^{(s)}$ is the Laplacian matrix of $\mathbf{W}^{(s)}$. Moreover, $\delta(c,n) \in \mathbb{R}^n$ denotes the vector whose c-th element is 1 and the rest elements are 0. For a scalar $x \in \mathbb{Z}$ and $x \in [1,n]$, we utilize the notation $x \in \mathbb{Z}_{[1,n]}$ for convenience. For scalars x and y, $x \propto y$ means that x is positive correlational to y.

2 EMGC^{2}F: Efficient Multi-view Graph Clustering with Comprehensive Fusion

In this section, we propose an Efficient Multi-view Graph Clustering with Comprehensive Fusion (EMGC^{2}F) model to seek for a consistent clustering decision from multi-view graphs $\{\boldsymbol{\mathcal{W}}^{(1)}, \ldots, \boldsymbol{\mathcal{W}}_{(s)}\}$, where [1] $\boldsymbol{\mathcal{W}}^{(s)} = (V, E^{(s)}, \mathbf{W}^{(s)})$, $s \in \mathbb{Z}_{[1,v]}$ is the graph of s-th view. For convenience, we introduce EMGC^{2}F model with the following three folds and summarize the workflow in Figure 2.

2.1 Super-Node Similarity Minimization Module

We first consider a single view form and start describing with set operator. Suppose Q is the set of all samples, Q_k is the set of the samples within k-th cluster, and $\bar{Q}_k$ is the complement of Q_k in Q. Considering the definition of graph clustering, an obvious observation can be obtained that the similarity between sets $(Q_k, \bar{Q}_k)$ should be as small as possible. Inspired of data hierarchical clustering [Reddy and Vinzamuri, 2013], on s-th view, we consider $(Q_k, \bar{Q}_k)$ as two super nodes, and then establish the similarity between $(Q_k, \bar{Q}_k)$ as the average pairwise similarities (average-link) as follows:

$$\mathrm{sim}(Q_k, \bar{Q}_k) = \frac{\sum_{i\in Q_k}\sum_{j\in\bar{Q}_k}(w_{i,j}^{(s)} + w_{j,i}^{(s)})}{2|Q_k||\bar{Q}_k|}. \quad (1)$$

[1] In this paper, we also allow $\mathbf{W}$ to be an asymmetric translation matrix such as $\mathbf{W}_{rw}^{(s)} = \mathbf{D}^{-1}\mathbf{W}^{(s)}$, thus we consider $w_{i,j}^{(s)}$ and $w_{j,i}^{(s)}$ separately in this paper.

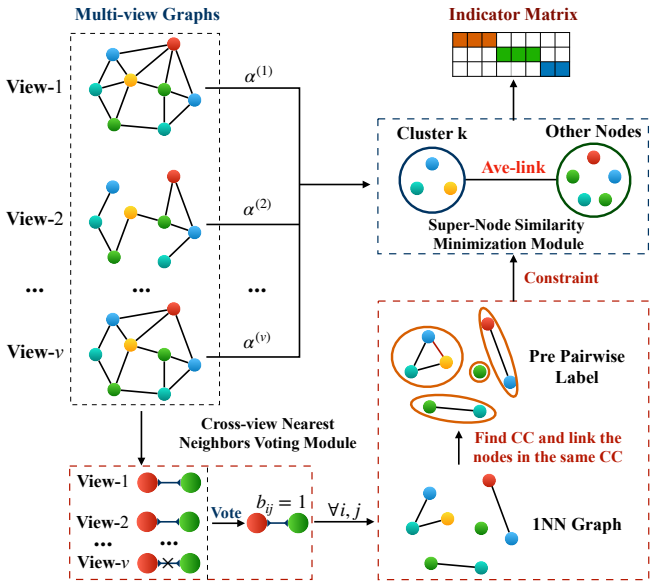


Figure 2: Workflow of the proposed EMGC^{2}F model. A CN^{2}V module is adopted to construct credible nearest neighbors via voting of multiple views, from which a 1NN graph is constructed and nodes in the same connected components are constrained to be in the same cluster. Each cluster is regarded as a super node, and we minimize similarities among different super nodes to obtain good clusters.

Then considering all clusters, the clustering model for s-th view can be written as

$$\begin{aligned} &\min_{\{Q_k\}_c} \sum_{k=1}^{c} \frac{\sum_{i\in Q_k}\sum_{j\in \bar{Q}_k}(w_{i,j}^{(s)}+w_{j,i}^{(s)})}{2|Q_k||\bar{Q}_k|} \\ &s.t.\ \bigcup_{k=1}^{c} Q_k = Q,\ \forall r,\ d\in\mathbb{Z}_{[1,c]},\ Q_r\bigcap Q_d=\oslash. \end{aligned} \tag{2}$$

For better illustration, we introduce an indicator matrix $\mathbf{Y}\in\{0,1\}_{n\times c}, \mathbf{Y1}=\mathbf{1}$ to replace the set operator $\{Q_k\}_{k=1}^c$, and rewrite problem (2) as follows:

$$\min_{\mathbf{Y}\in\{0,1\}_{n\times c},\ \mathbf{Y1}=\mathbf{1}} \sum_{k=1}^{c} \frac{\|\mathbf{L}^{(s)}\mathbf{y}_k\|_1}{2\|\mathbf{y}_k\|_0(n-\|\mathbf{y}_k\|_0)} \tag{3}$$

Afterwards, we consider the multi-view formulation of problem (3). To present the importance of each view, we consider learning an adaptive weight $\alpha^{(s)}$ for s-th view and problem (3) can be improved as

$$\begin{aligned} &\min_{\mathbf{Y},\{\alpha^{(s)}\}_v} \sum_{s=1}^{v}\alpha^{(s)} \sum_{k=1}^{c} \frac{\|\mathbf{L}^{(s)}\mathbf{y}_k\|_1}{2\|\mathbf{y}_k\|_0(n-\|\mathbf{y}_k\|_0)} \\ &s.t.\ \mathbf{Y}\in\{0,1\}_{n\times c},\ \mathbf{Y1}=\mathbf{1},\ \forall s\in\mathbb{Z}_{[1,v]}, \\ &\sum_{s=1}^{v}\frac{1}{\alpha^{(s)}}=1,\ \{\alpha^{(s)}\}_v\succeq 0, \end{aligned} \tag{4}$$

where $\sum_{s=1}^{v}\frac{1}{\alpha^{(s)}}=1$ is a scale constraint to guarantee that the solution $(\alpha^{(s)})^*$ and $\mathbf{y}_k^*$ satisfy the following criterion:

$$(\alpha^{(s)})^* \propto 1/\sum_{k=1}^{c}\frac{\|\mathbf{L}^{(s)}\mathbf{y}_k^*\|_1}{2\|\mathbf{y}_k^*\|_0(n-\|\mathbf{y}_k^*\|_0)}. \tag{5}$$

In Eq. (5), since the weight $(\alpha^{(s)})^*$ is negative correlational to the loss of $\mathbf{y}_k^*$, $(\alpha^{(s)})^*$ can be seen as importance of s-th view. Furthermore, we consider mining the consistent information among multiple views in the following part.

2.2 Cross-view Nearest Neighbors Voting Module

This part aims to find the credible consistency information among multiple views. To this goal, we first define a coefficient matrix $\mathbf{E}^{(s)}\in\mathbb{R}^{n\times n}$ to describe the nearest neighbors relationship in s-th view as follows

$$e_{i,j}^{(s)}=\begin{cases}1, & \text{If } i=\varpi_j^{(s)} \text{ or } j=\varpi_i^{(s)},\\ 0, & \text{Otherwise,}\end{cases} \tag{6}$$

where $e_{i,j}^{(s)}$ is the (i,j)-th element of $\mathbf{E}^{(s)}$ and $i=\varpi_j^{(s)}$ means that node i is the nearest neighbor of node j in $\mathbf{W}^{(s)}$. Since the nearest neighbor information for each pair of nodes may be inconsistent among different views, we define $\mathbf{B}\in\mathbb{R}^{n\times n}$ to describe cross-view nearest neighbors via voting as:

$$b_{i,j}=\begin{cases}1, & \text{If } \sum_{s=1}^{v}e_{i,j}^{(s)}>\lfloor\frac{v}{2}\rfloor,\\ 0, & \text{Otherwise,}\end{cases} \tag{7}$$

where $b_{i,j}=1$ means that i and j are cross-view nearest neighbors if they are nearest neighbors ($e_{i,j}^{(s)}=1$) in more than half of the views. Afterwards, we hope to generate some credible pairwise labels (whether node i and j belong to same cluster) from $\mathbf{B}$. Formally, we construct a graph $\boldsymbol{\mathcal{B}}$ associated with adjacency matrix $\mathbf{B}$ and find the connected components via Tarjan's algorithm [Tarjan, 1972], then we generate a pairwise label matrix $\mathbf{T}$ as follows:

$$t_{i,j}=\begin{cases}1, & \text{If } (i,j)\in \text{same component in } \boldsymbol{\mathcal{B}},\\ 0, & \text{Otherwise.}\end{cases} \tag{8}$$

$\mathbf{T}$ provides consistent pairwise label information via summarizing consistent cross-view nearest neighbors information. Furthermore, we add a [2]pairwise label constraint for $\mathbf{Y}$ via $\mathbf{T}$ and improve problem (4) as follows:

$$\begin{aligned} &\min_{\mathbf{Y},\{\alpha^{(s)}\}_v} \sum_{s=1}^{v}\alpha^{(s)} \sum_{k=1}^{c} \frac{\|\mathbf{L}^{(s)}\mathbf{y}_k\|_1}{2\|\mathbf{y}_k\|_0(n-\|\mathbf{y}_k\|_0)} \\ &s.t.\ \mathbf{Y}\in\{0,1\}_{n\times c},\ \mathbf{Y1}=\mathbf{1},\ \mathbf{YY}^T\circ\mathbf{T}=\mathbf{T}, \\ &\sum_{s=1}^{v}\frac{1}{\alpha^{(s)}}=1,\ \{\alpha^{(s)}\}_v\succeq 0, \end{aligned} \tag{9}$$

where the constraint $\mathbf{YY}^T\circ\mathbf{T}=\mathbf{T}$ guarantees that $t_{i,j}=1\Rightarrow\mathbf{y}^i=\mathbf{y}^j$. In summary, problem (9) mines the importance among different views via weights $\{\alpha^{(s)}\}_{s=1}^v$ and mine the consistent information among multiple views via constraint $\mathbf{YY}^T\circ\mathbf{T}=\mathbf{T}$, which preforms a comprehensive fusion among multi-view graphs.

[2] In fact, the cross-view information in $\mathbf{T}$ can be also utilized as regularization to emphasize on the nearest neighbors relationship, such as maximizing sum of $\mathbf{y}_k^T\mathbf{T}\mathbf{y}_k$. In this paper, we consider the simplest form that utilizing this information as constraint and the empirical results demonstrate its effectiveness.

3 Optimization

Problem (9) is hard to solve due to the highly non-convex constraints. In this section, we propose an effective and efficient algorithm to deal with it.

At first, we handle the constraint $\mathbf{Y}\mathbf{Y}^T \circ \mathbf{T} = \mathbf{T}$ with some equivalent transformations. To this goal, we need to reuse the set operator in Eq. (1). Suppose $\mathbf{T}$ has m diagonal blocks, *i.e.*, graph $\boldsymbol{\mathcal{B}}$ has m connected components, we define P_r as the set of samples within r-th component. Recall that Q_k is the set of samples within k-th cluster of $\boldsymbol{\mathcal{W}}^{(s)}$, its obvious that Q_k is composed of several different P_r according to problem (9), *i.e.*, $Q_k = \bigcup_{r\in\mathcal{I}_k} P_r$ where $\mathcal{I}_k$ as the corresponding indices. Based on the above definitions, we have

$$\begin{aligned}\frac{\|\mathbf{L}^{(s)}\mathbf{y}_k\|_1}{\|\mathbf{y}_k\|_0(n-\|\mathbf{y}_k\|_0)} &= \frac{\sum_{i\in Q_k}\sum_{j\in\bar{Q}_k}(w_{i,j}^{(s)}+w_{j,i}^{(s)})}{|Q_k||\bar{Q}_k|} \\ &= \frac{\sum_{r\in\mathcal{I}_k}\sum_{d\in\bar{\mathcal{I}}_k}\sum_{i\in P_r}\sum_{j\in P_d}(w_{i,j}^{(s)}+w_{j,i}^{(s)})}{(\sum_{r\in\mathcal{I}_k}|P_r|)(n-\sum_{r\in\mathcal{I}_k}|P_r|)}. \quad (10)\end{aligned}$$

Then for s-th view, we construct a graph $\boldsymbol{\mathcal{Z}}^{(s)}$ of m nodes associated with a similarity matrix $\mathbf{Z}^{(s)}$ where $z_{r,d}^{(s)} = \sum_{i\in P_r}\sum_{j\in P_d}(w_{i,j}^{(s)}+w_{j,i}^{(s)})$ and define a vector $\boldsymbol{\sigma}\in\mathbb{R}^m$ where $\sigma_r = |P_r|$. Substituting them into Eq. (10) results in

$$\frac{\|\mathbf{L}^{(s)}\mathbf{y}_k\|_1}{\|\mathbf{y}_k\|_0(n-\|\mathbf{y}_k\|_0)} = \frac{\sum_{r\in\mathcal{I}_k}\sum_{d\in\bar{\mathcal{I}}_k} z_{r,d}^{(s)}}{(\sum_{r\in\mathcal{I}_k}\sigma_r)(n-\sum_{r\in\mathcal{I}_k}\sigma_r)}. \quad (11)$$

Similar to transformation from problem (2) to (3), we introduce a new indicator matrix $\mathbf{H}\in\{0,1\}_{m\times c}$, $\mathbf{H1}=\mathbf{1}$ and rewrite problem (9) as follows from the perspective of coarsened multi-view graphs $\{\boldsymbol{\mathcal{Z}}^{(1)},\ldots,\boldsymbol{\mathcal{Z}}^{(v)}\}$:

$$\begin{aligned}&\min_{\{\alpha^{(s)}\}_v,\mathbf{H}}\sum_{s=1}^{v}\alpha^{(s)}\sum_{k=1}^{c}\frac{\|\widetilde{\mathbf{L}}^{(s)}\mathbf{h}_k\|_1}{2\mathbf{h}_k^T\boldsymbol{\sigma}(n-\mathbf{h}_k^T\boldsymbol{\sigma})} \quad (12)\\ &s.t.\ \mathbf{H}\in\{0,1\}_{m\times c},\ \mathbf{H1}=\mathbf{1},\sum_{s=1}^{v}\frac{1}{\alpha^{(s)}}=1,\{\alpha^{(s)}\}_v>0,\end{aligned}$$

where $\widetilde{\mathbf{L}}^{(s)}$ is Laplacian matrix of $\mathbf{Z}^{(s)}$. Then we solve problem (12) *w.r.t.* variables $\{\alpha^{(s)}\}_v$ and $\mathbf{H}$ alternatively.

1) $\{\alpha^{(s)}\}_v$-subproblem. For better illustration, we denote $\Delta^{(s)} = \sum_{k=1}^{c}\frac{\|\widetilde{\mathbf{L}}^{(s)}\mathbf{h}_k\|_1}{2\mathbf{h}_k^T\boldsymbol{\sigma}(n-\mathbf{h}_k^T\boldsymbol{\sigma})}$. Then when $\mathbf{H}$ is fixed, problem (12) *w.r.t.* $\{\alpha^{(s)}\}_v$ can be expressed as

$$\min_{\{\alpha^{(s)}\}_v}\sum_{s=1}^{v}\alpha^{(s)}\Delta^{(s)}\ s.t.\ \sum_{s=1}^{v}\frac{1}{\alpha^{(s)}}=1,\ \alpha^{(s)}\geqslant 0. \quad (13)$$

According to Cauchy-Schwarz inequality [Garling, 2005] and the constraint $\sum_{s=1}^{v}\frac{1}{\alpha^{(s)}}=1$, we have

$$\sum_{s=1}^{v}\alpha^{(s)}\Delta^{(s)} = \left(\sum_{s=1}^{v}\alpha^{(s)}\Delta^{(s)}\right)\left(\sum_{s=1}^{v}\frac{1}{\alpha^{(s)}}\right) \overset{\text{(i)}}{\geqslant} \left(\sum_{s=1}^{v}\sqrt{\Delta^{(s)}}\right)^2$$

where the equality in (i) holds when $\sqrt{\alpha^{(s)}\Delta^{(s)}} = r\sqrt{\frac{1}{\alpha^{(s)}}}$ with constant r. Combing with $\sum_{s=1}^{v}\frac{1}{\alpha^{(s)}}=1$, we have

$$\begin{aligned}&\sqrt{\alpha^{(s)}\Delta^{(s)}} = r\sqrt{\frac{1}{\alpha^{(s)}}} \Rightarrow \frac{1}{\alpha^{(s)}} = \frac{\sqrt{\Delta^{(s)}}}{r} \\ &\Rightarrow \sum_{s=1}^{v}\frac{1}{\alpha^{(s)}} = \sum_{s=1}^{v}\frac{\sqrt{\Delta^{(s)}}}{r} \Rightarrow r = \sum_{s=1}^{v}\sqrt{\Delta^{(s)}}, \quad (14)\end{aligned}$$

Then $\alpha^{(s)}$ can be calculated as

$$\alpha^{(s)} = \frac{\sum_{s'=1}^{v}\sqrt{\Delta^{(s')}}}{\sqrt{\Delta^{(s)}}}. \quad (15)$$

2) H-subproblem. When $\{\alpha^{(s)}\}_{s=1}^{v}$ is fixed, problem (12) *w.r.t.* $\mathbf{H}$ can be written as

$$\min_{\mathbf{H}\in\{0,1\}_{m\times c},\ \mathbf{H1}=\mathbf{1}}\sum_{s=1}^{v}\alpha^{(s)}\sum_{k=1}^{c}\frac{\|\widetilde{\mathbf{L}}^{(s)}\mathbf{h}_k\|_1}{2\mathbf{h}_k^T\boldsymbol{\sigma}(n-\mathbf{h}_k^T\boldsymbol{\sigma})} \quad (16)$$

For convenience, we denote $\widehat{\mathbf{Z}} = \sum_{s=1}^{v}\alpha^{(s)}(\mathbf{Z}^{(s)}+(\mathbf{Z}^{(s)})^T)$, then we have $\sum_{s=1}^{v}\alpha^{(s)}\|\widetilde{\mathbf{L}}^{(s)}\mathbf{h}_k\|_1 = \sum_{s=1}^{v}\alpha^{(s)}\mathbf{h}_k^T(\mathbf{Z}^{(s)}+(\mathbf{Z}^{(s)})^T)\mathbf{h}_k = \mathbf{h}_k^T\widehat{\mathbf{Z}}\mathbf{h}_k$, then problem (16) can be written as

$$\min_{\mathbf{H}\in\{0,1\}_{m\times c},\ \mathbf{H1}=\mathbf{1}}\sum_{k=1}^{c}\frac{\mathbf{h}_k^T\widehat{\mathbf{Z}}\mathbf{h}_k}{2\mathbf{h}_k^T\boldsymbol{\sigma}(n-\mathbf{h}_k^T\boldsymbol{\sigma})}. \quad (17)$$

Considering that the constraint $\mathbf{H1}=\mathbf{1}$ is row-independent, coordinate descend algorithm [Wright, 2015] can be utilized directly to handle problem (17). To be specific, suppose current solution is $\mathbf{H}$. When we seek for the solution $\mathbf{h}^r\in\mathbb{R}^{1\times c}$ to update r-th row $\mathbf{h}^r$, it can be chosen from c alternatives, including $\{\mathbf{h}^{[1]},\mathbf{h}^{[2]},\ldots,\mathbf{h}^{[c]}\}$, where $\mathbf{h}^{[i]} = \delta(i,c), \forall i\in\mathbb{Z}_{[1,c]}$. For better illustration, we let $\mathbf{h}^{[0]}\in\mathbb{R}^{1\times c}$ be a zero vector, and $\mathbf{H}^{[i]}$, $\forall i\in\mathbb{Z}_{[0,c]}$ be a matrix whose r-th row is $\mathbf{h}^{[i]}$ and the others are same as $\mathbf{H}$. Based on these definitions, $\bar{\mathbf{h}}^r$ can be calculated as $\bar{\mathbf{h}}^r = \delta(t,c)$ where

$$t = \arg\min_{i\in\mathbb{Z}_{[1,c]}}\sum_{k=1}^{c}\frac{(\mathbf{h}_k^{[i]})^T\widehat{\mathbf{Z}}\mathbf{h}_k^{[i]}}{2(\mathbf{h}_k^{[i]})^T\boldsymbol{\sigma}(n-(\mathbf{h}_k^{[i]})^T\boldsymbol{\sigma})}, \quad (18)$$

However, calculating Eq. (18) is time-consuming due to several redundant calculations. To accelerate it, we define

$$\mathcal{J}_{i,k} = \frac{(\mathbf{h}_k^{[i]})^T\widehat{\mathbf{Z}}\mathbf{h}_k^{[i]}}{2(\mathbf{h}_k^{[i]})^T\boldsymbol{\sigma}(n-(\mathbf{h}_k^{[i]})^T\boldsymbol{\sigma})}, i\in\mathbb{Z}_{[0,c]}, k\in\mathbb{Z}_{[1,c]}$$

and propose the following equivalent transformations:

$$\begin{aligned}t &= \arg\min_{i\in\mathbb{Z}_{[1,c]}}\sum_{k=1}^{c}\mathcal{J}_{i,k} = \arg\min_{i\in\mathbb{Z}_{[1,c]}}\sum_{k=1}^{c}\mathcal{J}_{i,k} - \sum_{k=1}^{c}\mathcal{J}_{0,k} \\ &= \arg\min_{i\in\mathbb{Z}_{[1,c]}}\sum_{k=1}^{c}(\mathcal{J}_{i,k}-\mathcal{J}_{0,k}) \overset{\text{(ii)}}{=} \arg\min_{i\in\mathbb{Z}_{[1,c]}}\mathcal{J}_{i,i}-\mathcal{J}_{0,i}, \quad (19)\end{aligned}$$

where the equality (ii) holds since $\mathcal{J}_{i,k} = \mathcal{J}_{0,k}, \forall i\neq k$. Then Eq. (18) can be improved as

$$t = \arg\min_{i\in\mathbb{Z}_{[1,c]}}\mathcal{J}_{i,i}-\mathcal{J}_{0,i}. \quad (20)$$

Algorithm 1 The Algorithm for Problem (9)

Input: $\{\boldsymbol{\mathcal{W}}^{(1)}, \ldots, \boldsymbol{\mathcal{W}}^{(s)}\}$
Output: $\{\alpha^{(s)}\}_{s=1}^{v}, \mathbf{Y}$

```
Calculate Z ∈ R^{m×m} and σ ∈ R^m in Eq. (10).
Initialize H.
while not converge do
    ∀s ∈ Z_[1,v], update α^(s) via Eq. (15).
    Calculate Ẑ = Σ_{s=1}^{v} α^(s)(Z^(s) + (Z^(s))^T).
    ∀i ∈ Z_[1,c], calculate h_i^T Ẑ h_i and h_i^T σ.
    while not converge do
        for r = 1 to n do
            Find the index of 1 in h^r and save as j.
            ∀i ∈ Z_[0,c], calculate H^[i].
            Calculate t via Eqs. (20) to (22) and h̄^r = δ(t, c)
            Update h^r as h̄^r. ∀i ∈ Z_[1,c], update h_i^T Ẑ h_i and
            h_i^T σ via Eqs. (23) and (24).
        end for
    end while
end while
Calculate Y via Eq. (25).
```

Obviously, calculating $\mathcal{J}_{i,i} - \mathcal{J}_{0,i}$ is dominated by repeatedly calculating $(\mathbf{h}_i^{[i]})^T \widehat{\mathbf{Z}} \mathbf{h}_i^{[i]}$ and $(\mathbf{h}_i^{[i]})^T \boldsymbol{\sigma}$ for different i, which is expensive yet redundant. For further acceleration, we propose to calculate them based on $\mathbf{h}_i^T \widehat{\mathbf{Z}} \mathbf{h}_i$, $\mathbf{h}_i^T \boldsymbol{\sigma}$ and $\mathbf{H}$ obtained from the last iteration. Concretely, we denote the index of 1 element in $\mathbf{h}^r$ as j and discuss the strategies for the following two cases:

a) When $i = j$, it's obvious that $\mathbf{h}_i^{[i]} = \mathbf{h}_i$ and $\mathbf{h}_i^{[0]} = \mathbf{h}_i - \boldsymbol{\delta}(r, m)$, then $\mathcal{J}_{i,i} - \mathcal{J}_{0,i}$ can be calculated as:

$$\frac{\mathbf{h}_i^T \widehat{\mathbf{Z}} \mathbf{h}_i}{2\mathbf{h}_i^T \boldsymbol{\sigma}(n - \mathbf{h}_i^T \boldsymbol{\sigma})} - \frac{\mathbf{h}_i^T \widehat{\mathbf{Z}} \mathbf{h}_i - 2\mathbf{h}_i^T \widehat{\mathbf{z}}_i + \widehat{z}_{i,i}}{2(\mathbf{h}_i^T \boldsymbol{\sigma} - \sigma_r)(n - \mathbf{h}_i^T \boldsymbol{\sigma} + \sigma_r)}. \quad (21)$$

b) When $i \neq j$, it's obvious that $\mathbf{h}_i^{[i]} = \mathbf{h}_i + \boldsymbol{\delta}(r, m)$ and $\mathbf{h}_i^{[0]} = \mathbf{h}_i$, then $\mathcal{J}_{i,i} - \mathcal{J}_{0,i}$ can be calculated as:

$$\frac{\mathbf{h}_i^T \widehat{\mathbf{Z}} \mathbf{h}_i + 2\mathbf{h}_i^T \widehat{\mathbf{z}}_i + \widehat{z}_{i,i}}{2(\mathbf{h}_i^T \boldsymbol{\sigma} + \sigma_r)(n - \mathbf{h}_i^T \boldsymbol{\sigma} - \sigma_r)} - \frac{\mathbf{h}_i^T \widehat{\mathbf{Z}} \mathbf{h}_i}{2\mathbf{h}_i^T \boldsymbol{\sigma}(n - \mathbf{h}_i^T \boldsymbol{\sigma})}. \quad (22)$$

t can be efficiently calculated by substituting Eqs. (21) and (22) into Eq. (20) and we obtain the optimal $\bar{\mathbf{h}}$.

Afterwards, we need to calculate $\bar{\mathbf{h}}_i^T \widehat{\mathbf{Z}} \bar{\mathbf{h}}_i$ and $\bar{\mathbf{h}}_i^T \boldsymbol{\sigma}$ for use when updating $\mathbf{h}^{r+1}$. Concretely, we denote the index of 1 element in $\bar{\mathbf{h}}$ as p, and then discuss the calculations of $\bar{\mathbf{h}}_i^T \widehat{\mathbf{Z}} \bar{\mathbf{h}}_i$ and $\bar{\mathbf{h}}_i^T \boldsymbol{\sigma}$ as the following two cases. When $i = p$, it is easy to observe that $\bar{\mathbf{h}}_i = \mathbf{h}_i^{[i]}$, then

$$\bar{\mathbf{h}}_i^T \widehat{\mathbf{Z}} \bar{\mathbf{h}}_i = (\mathbf{h}_i^{[i]})^T \widehat{\mathbf{Z}} \mathbf{h}_i^{[i]}, \ \bar{\mathbf{h}}_i^T \boldsymbol{\sigma} = \mathbf{h}_i^{[i]} \boldsymbol{\sigma}. \quad (23)$$

When $i \neq p$, it is obvious to obtain that $\bar{\mathbf{h}}_i = \mathbf{h}_i^{[0]}$, then

$$\bar{\mathbf{h}}_i^T \widehat{\mathbf{Z}} \bar{\mathbf{h}}_i = (\mathbf{h}_i^{[0]})^T \widehat{\mathbf{Z}} \mathbf{h}_i^{[0]}, \ \bar{\mathbf{h}}_i^T \boldsymbol{\sigma} = \mathbf{h}_i^{[0]} \boldsymbol{\sigma}. \quad (24)$$

Notice and all terms on the right hand side have been calculated, so no additional calculation is needed in this step.

Datasets	COIL20	DIGIT10	MSRC	ORL
Samples	1440	10000	210	400
Clusters	20	10	7	40
View-1	INTE(1024)	ISO(30)	CM(24)	GIST(512)
View-2	LBP(3304)	LDA(9)	HOG(576)	LBP(59)
View-3	GABOR(6750)	NPE(30)	GIST(512)	HOG(864)
View-4	—	—	LBP(256)	CENT(254)
View-5	—	—	CENT(254)	—

Table 1: Dataset descriptions. The numbers in parentheses indicate the dimension of corresponding features.

Updating $\{\alpha^{(s)}\}_{s=1}^{v}$ and $\mathbf{H}$ alternatively until convergence, we can obtain the optimal $\mathbf{H}$. So far, the optimization of problem (12) is completed, and the whole optimization procedure is summarized into Algorithm 1. Afterwards, the indicator matrix $\mathbf{Y}$ of problem (9) can be inferred as

$$i \in P_r \Rightarrow \mathbf{y}^i = \mathbf{h}^r, \ \forall i \in \mathbb{Z}_{[1,n]}. \quad (25)$$

Convergence analysis. Algorithm 1 optimizes $\{\alpha^{(s)}\}_v$ and $\mathbf{H}$ alternatively, and the corresponding sub-problems have been solved. Thus, we can conclude that the objective value of problem (12) will decrease until Algorithm 1 converges.

Time complexity. The time complexity of the proposal can be divided into two parts. 1) The time complexity of Eqs. (6) to (8): $O(|E|)$. 2) The time complexity of Algorithm 1: Lines 1–2 cost $O(|E|)$; In the outer-loop, Lines 4–6 cost $O(m^2 v)$ ($v \ll m$ in most cases). In the inner-loop, Line 11 costs $O(m)$ to calculate $\mathbf{h}_i^T \widehat{\mathbf{z}}_i$ (considering the sparsity of $\mathbf{h}_i$), so the total time complexity of Algorithm 1 is $O(m^2 vt)$ where t is the number of total iterations. In summary, the time complexity of the whole proposal is $O(|E| + m^2 vt)$.

4 Experiment

To evaluate the performance of our proposals, we employ 4 popular benchmark datasets including COIL20 [Nene *et al.*,], DIGIT10 [Wu *et al.*, 2021a], MSRC [Lee and Grauman, 2009], ORL [Samaria and Harter, 1994] and they are summarized into Table 1. 10 SOTA multi-view clustering competitors are employed, including AMGL [Nie *et al.*, 2016], SwMC [Nie *et al.*, 2017b], MLAN [Nie *et al.*, 2017a], MVGL [Zhan *et al.*, 2018], AWP [Nie *et al.*, 2018], MEA [Wu *et al.*, 2020], SFMC [Li *et al.*, 2020], NESE [Hu *et al.*, 2020], CDMGC [Huang *et al.*, 2021], OP-LFMVC [Liu *et al.*, 2021]. All the experiments are implemented in MATLAB R2020b on a desktop with Intel i7-7700k @ 4.2GHz CPU and 32GB RAM. Following [Nie *et al.*, 2017b], we construct KNN graph and set the number of nearest neighbors as 10. For AMGL and MEA that need K-means as post-processing, we run K-means 100 times and record results with minimum objective value. For OP-LFMVC, we utilize the multi-view spectral embeddings as input pre-processed matrix.

Effectiveness evaluations. We evaluate the effectiveness of EMGC^{2}F with 4 popular clustering metrics, including ACC, NMI, F1-Score and ARI. The clustering results are recorded in Table 2, from which we observe that the proposed EMGC^{2}F outperforms all SOTA competitors on all four

Method	Metric	AMGL	SwMC	MLAN	MVGL	AWP	MEA	SFMC	NESE	CDMGC	OP-LFMVC	EMGC^{2}F
COIL20	ACC	0.8340	0.8542	0.8424	0.7854	0.7299	0.8549	0.8069	0.7278	0.8757	0.7069	**0.9250**
	NMI	0.9125	0.9429	0.9250	0.9130	0.8975	0.9240	0.9197	0.8617	0.9450	0.8647	**0.9696**
	F1	0.7756	0.8410	0.8135	0.7867	0.7597	0.8324	0.8197	0.7070	0.8547	0.7135	**0.9244**
	ARI	0.7621	0.8318	0.8025	0.7739	0.7449	0.8233	0.8098	0.6902	0.8462	0.6947	**0.9203**
DIGIT10	ACC	0.8683	0.6148	0.5913	0.8458	0.6502	0.7067	0.7803	0.8711	0.8440	0.6851	**0.8793**
	NMI	0.7712	0.6094	0.6181	0.7639	0.6497	0.6833	0.7190	0.7743	0.7620	0.6628	**0.7836**
	F1	0.7664	0.5444	0.5395	0.7443	0.5725	0.6064	0.6821	0.7713	0.7472	0.6001	**0.7845**
	ARI	0.7403	0.4739	0.4736	0.7149	0.5198	0.5558	0.6416	0.7457	0.7183	0.5511	**0.7605**
MSRC	ACC	0.8571	0.7619	0.6952	0.8714	0.8952	0.8714	0.8238	0.7667	0.8286	0.8952	**0.9143**
	NMI	0.7623	0.7355	0.6565	0.7731	0.7921	0.7951	0.7539	0.7170	0.7642	0.7921	**0.8329**
	F1	0.7494	0.7138	0.6089	0.7560	0.8050	0.7730	0.7398	0.6944	0.7407	0.8050	**0.8364**
	ARI	0.7081	0.6619	0.5332	0.7152	0.7734	0.7354	0.6937	0.6433	0.6952	0.7734	**0.8099**
ORL	ACC	0.8325	0.8200	0.7600	0.8250	0.8000	0.7625	0.7875	0.8075	0.8250	0.7850	**0.8725**
	NMI	0.9273	0.9258	0.8716	0.9351	0.9142	0.9079	0.9039	0.9114	0.9320	0.9145	**0.9522**
	F1	0.7846	0.7416	0.6550	0.7993	0.7691	0.7130	0.6883	0.7512	0.7815	0.7576	**0.8547**
	ARI	0.7791	0.7347	0.6460	0.7942	0.7632	0.7053	0.6796	0.7449	0.7758	0.7513	**0.8511**

Table 2: Clustering results on 4 real-world datasets. Best results are in bold.

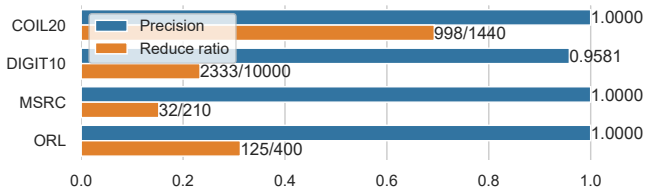


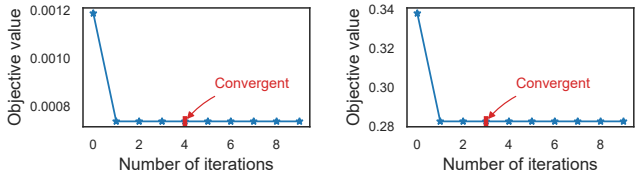
Figure 3: The precision of cross-view nearest neighbors being in the same ground truth cluster and the reduced ratio $(n-m)/n$ of CN^{2}V.

datasets. Specifically, EMGC^{2}F exceeds the best competitor by $\mathbf{0.0400}$(ACC), $\mathbf{0.0171}$(NMI), $\mathbf{0.0554}$(F1), $\mathbf{0.0569}$(ARI) on ORL dataset, and $\mathbf{0.0493}$(ACC), $\mathbf{0.0246}$(NMI), $\mathbf{0.0697}$(F1), $\mathbf{0.0741}$(ARI) on COIL20 dataset. Furthermore, the CN^{2}V module seeks for a part of (the number is adaptive) credible pairwise pre-labels, so the precision (the proportion of correct pairwise labels in predicted pairwise labels) is vital. We summarize the pairwise precision on 4 datasets into Figure 3, from which we observe that the precision of pairwise labels is $\mathbf{1}$ on MSRC, ORL, COIL20 datasets and $\mathbf{0.9581}$ on DIGIT10 dataset. Meanwhile, $\mathbf{998}$ nodes on COIL20 dataset and over 20% nodes on DIGIT10, ORL datasets are successfully reduced. These promising results indicate that CN^{2}V is extremely effective at extracting consistent information from multiple views and thus beneficial for subsequent procedures.

Efficiency evaluations. In this part, we aim to evaluate the efficiency of the optimization (Algorithm 1) of EMGC^{2}F model. For each model, we run it for 5 times and report the averaged execution time in Table 3. From the results, we can observe that the optimization efficiency of EMGC^{2}F is far ahead of other models, which matches the theoretical analysis that the $O(|E|+m^2vt)$ complexity of EMGC^{2}F optimization is much more efficient than $O(n^3)$ of current works that depend on spectral decomposition in pro-processing or (and) learning procedure. The convergence curves on 2 benchmark datasets are plotted into Figure 4, from which we observe that Algorithm 1 can converge rapidly.

Algorithm	COIL20	DIGIT10	MSRC	ORL
AMGL	1.2397	515.5103	0.2326	0.4684
SwMC	323.7284	—	5.9962	11.8308
MLAN	8.4658	—	0.1022	0.2913
MVGL	91.3807	—	0.7722	2.9360
AWP	0.5025	169.6220	0.0169	0.0713
MEA	9.3117	—	0.3328	1.0009
SFMC	7.1074	—	0.7504	2.0840
NESE	3.9386	183.4303	0.1025	0.4033
CDMGC	10.8803	—	0.5603	1.2253
OP-LFMVC	0.9626	228.9462	0.1272	0.4398
EMGC^{2}F	**0.0136**	**0.4829**	**0.0121**	**0.0090**

Table 3: Mean execution time (s) of 5 runs. >1000 s are omitted.



(a) DIGIT10 (b) MSRC

Figure 4: The convergent curves on 2 real datasets.

5 Conclusion

This paper proposes an EMGC^{2}F model for MGC, which learns consistent clustering labels from multiple views directly and avoids time-consuming spectral decomposition in current works. Through our model, the importance and credible consistent information is captured comprehensively. An efficient discrete algorithm of $O(|E|+m^2vt)$ time complexity is provided. Empirical results demonstrate the effectiveness and efficiency of our proposal. Besides, EMGC^{2}F exploit credible consistent information as a constraint directly, which is the simplest way. In the future, how to use the information more flexibly and deeply may require more concern.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101902, in part by the Natural Science Basic Research Program of Shaanxi (Program No. 2021JM-071), in part by the National Natural Science Foundation of China under Grant 62176212, Grant 61936014 and Grant 61772427, and in part by the Fundamental Research Funds for the Central Universities under Grant G2019KY0501.

References

[Belkin and Niyogi, 2003] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

[Chao *et al.*, 2021] Guoqing Chao, Shiliang Sun, and Jinbo Bi. A survey on multi-view clustering. *IEEE Trans. on Artif. Intell.*, 2021.

[Dong *et al.*, 2014] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Trans. Signal Process.*, 62(4):905–918, 2014.

[Dong *et al.*, 2021] Xia Dong, Danyang Wu, Feiping Nie, Rong Wang, and Xuelong Li. Dependence-guided multi-view clustering. In *ICASSP*, pages 3650–3654. IEEE, 2021.

[Garling, 2005] D. J. H. Garling. The cauchy-schwarz master class: An introduction to the art of mathematical inequalities by j. michael steele. *Am. Math. Mon.*, 112(6):575–579, 2005.

[Hu *et al.*, 2020] Zhanxuan Hu, Feiping Nie, Rong Wang, and Xuelong Li. Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding. *Inf. Fusion*, 55:251–259, 2020.

[Huang *et al.*, 2021] Shudong Huang, Ivor Tsang, Zenglin Xu, and Jian Cheng Lv. Measuring diversity in graph learning: A unified framework for structured multi-view clustering. *IEEE Trans. Knowl. Data Eng.*, 2021.

[Kumar *et al.*, 2011] Abhishek Kumar, Piyush Rai, and Hal Daumé III. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[Lee and Grauman, 2009] Yong Jae Lee and Kristen Grauman. Foreground focus: Unsupervised learning from partially matching images. *Int. J. Comput. Vis.*, 85(2):143–166, 2009.

[Li *et al.*, 2020] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. Multi-view clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[Liu *et al.*, 2021] Xinwang Liu, Li Liu, Qing Liao, Siwei Wang, Yi Zhang, Wenxuan Tu, Chang Tang, Jiyuan Liu, and En Zhu. One pass late fusion multi-view clustering. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 6850–6859. PMLR, 2021.

[Nene *et al.*,] Sameer A Nene, Shree K Nayar, and Hiroshi Murase. Columbia object image library (coil-20).

[Nie *et al.*, 2016] Feiping Nie, Jing Li, and Xuelong Li. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *IJCAI*, pages 1881–1887. IJCAI/AAAI Press, 2016.

[Nie *et al.*, 2017a] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*, pages 2408–2414. AAAI Press, 2017.

[Nie *et al.*, 2017b] Feiping Nie, Jing Li, and Xuelong Li. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, pages 2564–2570. ijcai.org, 2017.

[Nie *et al.*, 2018] Feiping Nie, Lai Tian, and Xuelong Li. Multiview clustering via adaptively weighted procrustes. In *KDD*, pages 2022–2030. ACM, 2018.

[Reddy and Vinzamuri, 2013] Chandan K. Reddy and Bhanukiran Vinzamuri. A survey of partitional and hierarchical clustering algorithms. In *Data Clustering: Algorithms and Applications*, pages 87–110. CRC Press, 2013.

[Samaria and Harter, 1994] Ferdinand Samaria and Andy Harter. Parameterisation of a stochastic model for human face identification. In *WACV*, pages 138–142. IEEE, 1994.

[Tarjan, 1972] Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[Wright, 2015] Stephen J. Wright. Coordinate descent algorithms. *Math. Program.*, 151(1):3–34, 2015.

[Wu *et al.*, 2020] Danyang Wu, Feiping Nie, Rong Wang, and Xuelong Li. Multi-view clustering via mixed embedding approximation. In *ICASSP*, pages 3977–3981. IEEE, 2020.

[Wu *et al.*, 2021a] Danyang Wu, Zhanxuan Hu, Feiping Nie, Rong Wang, Hui Yang, and Xuelong Li. Multi-view clustering with interactive mechanism. *Neurocomputing*, 449:378–388, 2021.

[Wu *et al.*, 2021b] Danyang Wu, Feiping Nie, Xia Dong, Rong Wang, and Xuelong Li. Parameter-free consensus embedding learning for multiview graph-based clustering. *IEEE Trans. Neural Netw. Learn. Syst.*, 2021.

[Xu *et al.*, 2013] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *CoRR*, abs/1304.5634, 2013.

[Zhan *et al.*, 2018] Kun Zhan, Changqing Zhang, Junpeng Guan, and Junsheng Wang. Graph learning for multi-view clustering. *IEEE Trans. Cybern.*, 48(10):2887–2895, 2018.