Let the Data Choose: Flexible and Diverse Anchor Graph Fusion for Scalable Multi-View Clustering

Pei Zhang[1], **Siwei Wang**[1], **Liang Li**[1], **Changwang Zhang**[2], **Xinwang Liu**[1]*, **En Zhu**[1]*,
Zhe Liu[3], **Lu Zhou**[3], **Lei Luo**[1]*
[1] School of Computer, National University of Defense Technology, Changsha, China, 410073
[2] Huawei Poisson Lab, Shenzhen, China, 518129
[3] Nanjing University of Aeronautics and Astronautics, Nanjing, China, 210000
{zhangpei,wangsiwei13}@nudt.edu.cn, lianglee@alumni.hust.edu.cn, changwangzhang@foxmail.com,
{xinwangliu,enzhu}@nudt.edu.cn, {zhe.liu, lu.zhou}@nuaa.edu.cn, l.luo@nudt.edu.cn

Abstract

In the past few years, numerous multi-view graph clustering algorithms have been proposed to enhance the clustering performance by exploring information from multiple views. Despite the superior performance, the high time and space expenditures limit their scalability. Accordingly, anchor graph learning has been introduced to alleviate the computational complexity. However, existing approaches can be further improved by the following considerations: (i) Existing anchor-based methods share the same number of anchors across views. This strategy violates the diversity and flexibility of multi-view data distribution. (ii) Searching for the optimal anchor number within hyper-parameters takes much extra tuning time, which makes existing methods impractical. (iii) How to flexibly fuse multi-view anchor graphs of diverse sizes has not been well explored in existing literature. To address the above issues, we propose a novel anchor-based method termed Flexible and Diverse Anchor Graph Fusion for Scalable Multi-view Clustering (FDAGF) in this paper. Instead of manually tuning optimal anchor with massive hyper-parameters, we propose to optimize the contribution weights of a group of pre-defined anchor numbers to avoid extra time expenditure among views. Most importantly, we propose a novel hybrid fusion strategy for multi-size anchor graphs with theoretical proof, which allows flexible and diverse anchor graph fusion. Then, an efficient linear optimization algorithm is proposed to solve the resultant problem. Comprehensive experimental results demonstrate the effectiveness and efficiency of our proposed framework. The source code is available at https://github.com/Jeaninezpp/FDAGF.

Introduction

Nowadays, many contrastive learning methods employ augmented views to enhance the learning of original data (Yang et al. 2022; Liu et al. 2022b). However, the data itself can be represented from multiple views thanks to the improvement in data collection and feature extraction techniques. Multi-view clustering (MVC) is an important unsupervised learning method in machine learning and data mining, which is based on the concept of integrating complementary and diverse information across views (Nie, Cai, and Li 2017; Liu et al. 2018). Multi-view graph clustering (MVGC) (Zhang et al. 2022b; Sun et al. 2021), multi-kernel clustering (MKC) (Zhang et al. 2022a,c; Li et al. 2022) and multi-view matrix factorization clustering (MVMFC) (Liu et al. 2013; Zhao, Ding, and Fu 2017; Yang et al. 2020; Zhang et al. 2021a) are three common solutions for MVC task. Among them, MVGC methods aim at constructing one or multiple similarity matrices (Cao et al. 2015; Li et al. 2019; Kang et al. 2020a). These approaches explore local or global relationships between samples to establish connections across views. Despite the promising performance of MVGC method, their capability to scale up is severely constrained by the high complexity. Therefore, it is essential to design efficient algorithms to tackle large-scale MVC issues in the current era of big data. Recently, researchers propose the anchor-based MVGC approaches to alleviate the complexity of traditional multi-view graph clustering (Li et al. 2015; Kang et al. 2020b; Li et al. 2020; Sun et al. 2021; Wang et al. 2021; Zhang et al. 2021b). The basic goal of these methods is to construct the anchor graph between anchors and the entire samples instead of constructing the full graph. In this way, the time expenditures $\mathcal{O}(n^3)$ is reduced to $\mathcal{O}(n)$. For instance, Kang et al. use the pre-defined clustering centers as the anchor to obtain the anchor graph for each view independently. Sun et al. and Wang et al. further incorporate anchor into the optimization framework and finally obtain a unified anchor graph across multiple views. Li et al. provide a parameter-free and adaptive weighted fusion framework to obtain a joint graph.

Although the anchor-based MVGC has improved the efficiency of algorithms, it can still be enhanced in the following ways. Firstly, current anchor-based MVGC approaches require all views to share the same number of anchors, which violates the diversity principle in multi-view data distribution. In this case, the optimal number of anchors is determined by hyper-parameter traversal in a wide range for each dataset, intensively degrading model efficiency and application. Therefore, how to automatically choose the diverse anchor number for different views is worthwhile to investigate. Secondly, the problem of multi-size anchor fusion brought by the flexible number of anchors has not received enough attention. Therefore, designing a flexible and scalable anchor fusion algorithm is an urgent need in MVGC.

*Corresponding authors.

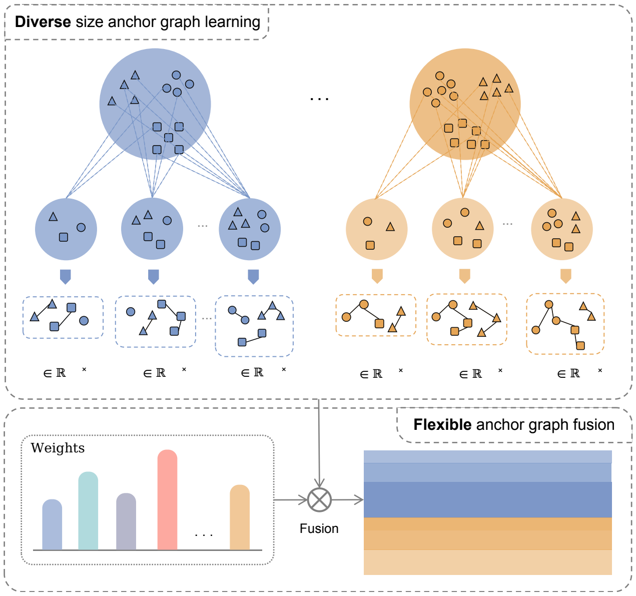

Diverse size anchor graph learning
...
∈ℝ ×
∈ℝ ×
∈ℝ ×
∈ℝ ×
∈ℝ ×
∈ℝ ×
Flexible anchor graph fusion
Weights
...
Fusion

Figure 1: The framework of our proposed FDAGF algorithm. For each view of data X_v, we provide diverse choices of anchor (from A_v^1 to A_v^R), resulting in multiple flexible sizes of anchor graphs $\{Z_v^1, \cdots, Z_v^R\}$. Then, the importance corresponding to each anchor choice w.r.t each view is automatically learned to incorporate the fused anchor graph.

To address the above issues, we propose a novel anchor-based multi-view graph clustering method in this paper, termed **F**lexible and **D**iverse **A**nchor **G**raph **F**usion for Scalable Multi-view Clustering (FDAGF). Figure 1 shows the framework of the proposed method. Specifically, we offer diverse choices for the number of anchors, resulting in diverse anchor graphs with flexible sizes for each view. The weights across diverse choices of multiple views are automatically determined via optimization based on the discrimination of anchor graphs. Most significantly, we propose a novel paradigm for fusing hybrid multi-size anchor graphs based on rigorous theoretical analysis. Moreover, the linear complexity of our proposed method, which inherits the benefits of anchor strategy, also makes FDAGF applicable to large-scale multi-view data. Compared with the state-of-the-art MVC methods, extensive experiments demonstrate the effectiveness and efficiency of our proposed FDAGF. Our main contributions can be summarized as follows:

1. We propose to optimize the diverse weights of multi-view multi-choice anchor graphs as opposed to manually determining the ideal number of anchors through hyper-parameter-traversal. In addition to avoiding searching for optimal anchor numbers, optimizing the importance of each choice on each view also allows for a deeper exploration of multiple view information, which naturally enhances the clustering performance.
2. We present a hybrid multi-size anchor graph fusion paradigm to obtain the unified fusion graph in this paper. Compared to the current anchor graph fusion methods, our proposed strategy is more flexible and compatible with the multi-view principle.
3. We design an alternating optimization algorithm to solve the resulting optimization problem with linear complexity about the number of samples. Extensive experimental results demonstrate the superiority of our clustering performance and running time.

Related Work

In this section, we review the rationale and literature of anchor-based multi-view graph clustering methods. Then the most relevant algorithms are introduced in detail. The notations used in this paper are listed in Table 1.

Notation	Description
n, k, V	Number of samples, clusters and views
R	Number of anchor choice
d_v	Feature dimension of v-th view
$X_v \in \mathbb{R}^{d_v \times n}$	Data matrix of v-th view
$\boldsymbol{\beta} \in \mathbb{R}^{V \times R}$	Importance of choice on each view
$A_v^r \in \mathbb{R}^{d_v \times m_r}$	Anchor matrix of r-th choice on v-th view
$Z_v^r \in \mathbb{R}^{m_r \times n}$	Anchor graph of r-th choice on v-th view
$\hat{Z}_v^r$	Normalized anchor graph
$S_v^r \in \mathbb{R}^{n \times n}$	The full graph
$M_{[:,j]}$	The j-th column of matrix M

Table 1: Description of notations in this paper.

Anchor-based MVGC

Sampling techniques are widely employed in both academia and industry due to the enormous data scale. It has also been used for the existing multi-view spectral clustering (Li et al. 2015) and multi-view subspace clustering (Kang et al. 2020b) in the field of multi-view graph clustering. Among anchor-based MVGC methods, researchers choose a small number of instances as anchors and then learn anchor graphs between the anchors and original samples. By doing this, the inefficient $n \times n$ affinity graph is replaced by the $m \times n$ anchor graph, where m is the number of anchor points. The success of sampling shows that the sampling strategy contributes to faster computation and storage while maintaining equivalent clustering performance.

Recently, Kang et al. proposed a Large-scale Multi-view Subspace Clustering in Linear Time (LMVSC) algorithm to replace the self-expression of multi-view subspace clustering with anchor-samples expression, making it possible to handle large-scale clustering scenarios. Sun et al. consider that the anchors obtained by optimization are more representative than fixed anchors. They unified the individual anchor learning with anchor graph optimization in one framework. Wang et al. simplify the above method to a parameter-free version, further improving the clustering performance and efficiency. Based on it, Liu et al. propose a one-pass approach to directly obtain the clustering labels by imposing graph connectivity constraints on the anchor graph. Moreover, Li et al. propose an anchor sampling strategy for constructing independent anchor graphs and then fusing them adaptively into the consensus graph.

Large-scale Multi-view Subspace Clustering in Linear Time (LMVSC)

Kang et al. make the first effort at anchor-based MVGC. Firstly, they perform k-means to obtain the view-specific anchors and fix them. Anchor graphs between anchors and all samples on each view are constructed independently. Finally, they execute Singular Value Decomposition (SVD) on the concatenated graph to obtain the common representation and then execute k-means to get the final clustering result. The formulation of LMVSC can be written as:

$$\begin{aligned}&\min_{Z_v} \sum_{v=1}^{V} \|X_v - A_v Z_v\|_F^2 + \alpha \|Z_v\|_F^2 \\ &\text{s.t. } 0 \leq Z_v, (Z_v)^\top \mathbf{1} = \mathbf{1},\end{aligned} \tag{1}$$

where $X_v \in \mathbb{R}^{d_v \times n}$ and $Z_v \in \mathbb{R}^{m \times n}$. α is a trade-off parameter. m refers to the number of anchors, which is a hyper-parameter and remains the same across views. Z_v can be obtained by solving the Quadratic Programming (QP) problem. The common representation can be obtained by computing SVD of the concatenated matrix $\overline{Z} \in \mathbb{R}^{mv \times n}$.

In general, LMVSC provides a learning framework for anchor-based MVGC, but it can still be improved by the following considerations. The anchor is pre-generated without optimization, which leads to the separation of anchor learning and clustering. LMVSC treats each view identically and lacks consistent and diverse information across multiple views. Based on this framework, we propose a more flexible and diverse anchor fusion paradigm in the next section.

Methodology

We begin this section by defining the anchor-based MVGC and then explain how to build our model step by step.

Definition 1 (Anchor-based MVGC) *Given a multi-view dataset $X = \{X_v\}_{v=1}^{V}$ with n samples, V views and k clusters. For the v-th view, $X_v \in \mathbb{R}^{d_v \times n}$ has d_v dimensional feature. The anchor matrix of v-th view is $A_v \in \mathbb{R}^{d_v \times m}$, where m is the number of anchors. Correspondingly, the anchor graph of v-th view is $Z_v \in \mathbb{R}^{m \times n}$, depicting the relationships between m anchors and original n instances.*

Multi-view Diversity

According to the generally recognized multi-view principle (Yang and Wang 2018), the distribution of multi-view data has the property of complementary and consistency. The traditional anchor-based MVGC framework, as illustrated in the Definition 1, requires multiple views to share the same number of anchors, which not only violates the complementary principle but is also time-consuming when searching for the optimal anchor number. Considering this, the framework in Eq. 1 can be upgraded to search for the ideal number of anchors independently on each view. The difference is that the dimension of Z_v is $m_{v_{\text{optimal}}} \times n$ instead of $m \times n$. However, searching for the ideal number of anchors for each view takes more time than searching for a common optimal number. Therefore, it is worthwhile to explore *how to choose the self-adaptable number of anchors for different views while maintaining the diversity of multiple views.*

Generate Multi-size Multi-view Anchor Graph

Compared to existing methods that require a range of hyper-parameters to find an optimal dataset-related anchor number, we consider providing a group of anchors with the diverse number for each view to approximate the optimum. In this manner, the hyper-parameter traversal for optimal number is avoided, and the diversity of each view is preserved.

Specifically, we offer R number of anchor choices for each view. Hence the anchor matrices can be expressed by $\{\{A_v^r\}_{v=1}^{V}\}_{r=1}^{R}$, where $A_v^r \in \mathbb{R}^{d_v \times m_r}$, m_r is the anchor number of the r-th choice. Correspondingly, the resultant flexible graphs with diverse sizes can be written as $\{\{Z_v^r\}_{v=1}^{V}\}_{r=1}^{R}$, where $Z_v^r \in \mathbb{R}^{m_r \times n}$. Taking the i-th view as an example, the corresponding anchor matrices and anchor graphs can be formulated as $\{A_i^1, A_i^2, \cdots, A_i^R\}$ and $\{Z_i^1, Z_i^2, \cdots, Z_i^R\}$. Consequently, the proposed flexible and diverse anchor graph fusion model for multi-view clustering can be formulated as follows:

$$\begin{aligned}&\min_{A_v^r, Z_v^r, \boldsymbol{\beta}} \sum_{v=1}^{V} \sum_{r=1}^{R} \beta_v^r \left(\|X_v^r - A_v^r Z_v^r\|_F^2 + \alpha \|Z_v^r\|_F^2 \right) + \lambda \|\boldsymbol{\beta}\|_F^2 \\ &\text{s.t. } (A_v^r)^\top A_v^r = I, Z_v^r \geq 0, (Z_v^r)^\top \mathbf{1} = \mathbf{1}, \boldsymbol{\beta}\mathbf{1} = \mathbf{1},\end{aligned} \tag{2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{V \times R}$ and β_v^r indicates the importance of the r-th anchor choice on v-th view. α and λ are the trade-off parameters. The soft assignment of the automatic weight learning framework benefits from the regularization term of $\boldsymbol{\beta}$. When $\lambda = 0$, the framework is degraded to a trivial solution of $\boldsymbol{\beta}$ with only one choice chosen.

Flexible Multi-view Anchor Graph Fusion

Once we obtain multi-size anchor graphs across multiple views, the next crucial problem still remains to be explored: *how to fuse multiple anchor graphs with diverse sizes?*

Traditionally, the flexible and diverse anchor graphs $Z_v^r \in \mathbb{R}^{m_r \times n}$ could restore the full $n \times n$ affinity matrix by applying Eq. (3) according to the landmark spectral clustering and anchor graphs (Liu, He, and Chang 2010). The resultant $S_v^r \in \mathbb{R}^{n \times n}$ is a doubly-stochastic matrix where the sum of each column and each row is one.

$$S_v^r = {Z_v^r}^\top \Sigma^{-1} Z_v^r, \text{ where } \Sigma = \text{diag}(Z_v^r \mathbf{1}). \tag{3}$$

By setting $\hat{Z}_v^r = \Sigma^{-\frac{1}{2}} Z_v^r$, we can obtain the full graph $S_v^r = {\hat{Z}_v^r}^\top \hat{Z}_v^r$, and the adaptive fusion graph $\overline{S} = \sum_{v=1}^{V} \sum_{r=1}^{R} \beta_v^r S_v^r$. The common solution for MVGC is to conduct spectral clustering on $\overline{S}$ and then feed the spectral embedding into k-means to produce the clustering result. However, performing SVD on a $n \times n$ matrix necessitates $\mathcal{O}(n^3)$ time complexity, which is exactly why most MVGC algorithms are unsuitable for large-scale data. In this paper, instead of constructing S_v^r and $\overline{S}$, we propose an alternative strategy to compute the k right singular vectors of the concatenated anchor graph $\overline{Z}$ in Eq. (4) as the common embeddings, according to Proposition 1 (Affeldt, Labiod, and Nadif 2020; Kang et al. 2020b).

$$\overline{Z} = \left[\sqrt{\beta_1^1} \hat{Z}_1^1; \sqrt{\beta_1^2} \hat{Z}_1^2; \cdots; \sqrt{\beta_V^R} \hat{Z}_V^R \right]. \tag{4}$$

Proposition 1 *Given similarity matrices $\{\{S_v^r\}_{v=1}^V\}_{r=1}^R$ and the adaptive fused graph $\overline{S} = \sum_{v=1}^V \sum_{r=1}^R \beta_v^r S_v^r$. Each of S_v^r can be expressed by $S_v^r = {\hat{Z}_v^r}^\top \hat{Z}_v^r$. Let $\overline{Z} = \left[\sqrt{\beta_1^1}\hat{Z}_1^1; \sqrt{\beta_1^2}\hat{Z}_1^2; \cdots; \sqrt{\beta_V^R}\hat{Z}_V^R\right]$, $\overline{Z} \in \mathbb{R}^{\frac{kVR(R+1)}{2}\times n}$. Assuming the SVD of $\overline{Z}$ is $\overline{Z} = U\Lambda H^\top$, where $U^\top U = I, H^\top H = I$. We can conclude that the eigenvectors of $\overline{S}$ equal to the right singular vectors of $\overline{Z}$.*

Proof 1 *Given the current circumstances, we can obtain*

$$\begin{aligned}\overline{S} &= \sum_{v=1}^V \sum_{r=1}^R \beta_v^r S_v^r = \sum_{v=1}^V \sum_{r=1}^R \beta_v^r {\hat{Z}_v^r}^\top \hat{Z}_v^r \\ &= \overline{Z}^\top \overline{Z} = (U\Lambda H^\top)^\top (U\Lambda H^\top) \\ &= H\Lambda^2 H^\top.\end{aligned}$$

Thereby the eigenvectors of $\overline{S}$ equal to the right singular vectors of $\overline{Z}$.

Since $\frac{kVR(R+1)}{2} \ll n$ in most large-scale scenarios, we conduct SVD decomposition on $\overline{Z}$ instead of eigenvalue decomposition on $\overline{S}$, which significantly reduces the time expenditure. Additionally, avoiding the construction of $n \times n$ S_v^r and $\overline{S}$ greatly decreases the cost of storage.

Optimization and Analysis

Optimization

To solve the optimization problem in Eq. (2), we design a three-step alternating algorithm.

Update A_v^r Fixing other variables, the optimization problem of A_v^r can be formulated as

$$\min_{A_v^r} \|X_v - A_v^r Z_v^r\|_F^2, \text{ s.t. } (A_v^r)^\top A_v^r = I. \tag{5}$$

By expanding the Frobenius norm into trace form, the problem of optimizing A_v^r is

$$\max_{A_v^r} \operatorname{Tr}\left((A_v^r)^\top B_v^r\right), \text{ s.t. } (A_v^r)^\top A_v^r = I, \tag{6}$$

where $B_v^r = X_v (Z_v^r)^\top$. The optimum of A_v^r can be obtained by computing $A_v^r = U_b V_b^\top$, where U_b and V_b are the left and right singular matrices of B_v^r, respectively.

Update Z_v^r When optimizing variable Z_v^r, for each view and each anchor choice, anchor graph Z_v^r can be rewritten as the following trace form,

$$\begin{aligned}&\min_{Z_v^r} \operatorname{Tr}\left((1+\alpha)(Z_v^r)^\top Z_v^r - 2(Z_v^r)^\top (A_v^r)^\top X_v\right), \\ &\text{s.t. } (Z_v^r)^\top \mathbf{1} = \mathbf{1}, Z_v^r \geq 0.\end{aligned} \tag{7}$$

Since $(A_v^r)^\top X_v$ is a constant when optimizing Z_v^r, minimizing Eq. (7) is equivalent to

$$\min_{Z_v^r} \left\| Z_v^r - \frac{1}{1+\alpha}(A_v^r)^\top X_v \right\|_F^2, \text{s.t. } (Z_v^r)^\top \mathbf{1} = \mathbf{1}, Z_v^r \geq 0. \tag{8}$$

Denoting $E = Z_v^r$ and $H = \frac{1}{1+\alpha}(A_v^r)^\top X_v$, Eq. (8) can be solved by the closed-form solution (Nie, Wang, and Huang 2014):

$$E_{[:,j]} = \max\left(H_{[:,j]} + \eta_j \mathbf{1}, 0\right), \text{where } \eta_j = \frac{1 + H_{[:,j]}^\top \mathbf{1}}{n}. \tag{9}$$

Update $\boldsymbol{\beta}$ Fixing other variables, the optimization problem of $\boldsymbol{\beta}$ can be formulated as

$$\min_{\boldsymbol{\beta}} \sum_{r=1}^R \beta_v^r \xi_v^r + \lambda(\beta_v^r)^2, \text{s.t. } \boldsymbol{\beta}\mathbf{1} = \mathbf{1}, \tag{10}$$

where $\xi_v^r = \|X_v^r - A_v^r Z_v^r\|_F^2 + \alpha\|Z_v^r\|_F^2$ and β_v^r refers to the element in v-th row and r-th column of $\boldsymbol{\beta}$. $\boldsymbol{\beta}_{[v,:]}$ denotes v-th row of $\boldsymbol{\beta}$, representing the weights of v-th view. For each $\boldsymbol{\beta}_{[v,:]}$, we obtain the following problem,

$$\min_{\boldsymbol{\beta}_{[v,:]}} \left\| \boldsymbol{\beta}_{[v,:]} - \vartheta_v \right\|_2^2, \text{ s.t. } \boldsymbol{\beta}_{[v,:]}\mathbf{1} = 1, \tag{11}$$

where $\vartheta_v = -\frac{1}{2\lambda}[\xi_v^1, \xi_v^2, \cdots, \xi_v^R] \in \mathbb{R}^{1\times R}$. Eq. (11) is the same problem as in Eq. (8) and can also be solved with a closed-form solution.

The entire procedures of our proposed FDAGF algorithm are summarized in Algorithm 1.

Algorithm 1: FDAGF algorithm

Input: Multi-view data $\{X_v\}_{v=1}^V$, cluster k, choice R.
Parameter: Trade-off parameter α and λ.
Output: Perform k-means on H.

```
Initialize $A_v^r$, $Z_v^r$ and $\boldsymbol{\beta}$.
while not convergent do
    Update $A_v^r$ by solving Eq. (6).
    Update $Z_v^r$ by solving Eq. (8).
    Update $\boldsymbol{\beta}$ by solving Eq. (11).
end while
Construct $\overline{Z}$ according to Eq. (4).
Return The right singular vectors $H$ of the concatenated matrix $\overline{Z}$.
```

Analysis and Extensions

Computational Complexity: The computational complexity of our model during optimization involves three parts. Supposing the r-th choice number is m_r, the first step to update A_v^r requires $\mathcal{O}\left(d_v m_r^2\right)$. Updating Z_v^r needs $\mathcal{O}\left(n m_r^2\right)$, and updating $\boldsymbol{\beta}$ requires $\mathcal{O}\left(d_v m_r n\right)$. For all of the views and choices, our model achieves total $\mathcal{O}\left(\sum_{v=1}^V \sum_{r=1}^R (d_v m_r^2 + m_r^2 n + d_v m_r n)\right)$ computational complexity during each iteration. The post-processing of SVD and k-means are also linear complexity w.r.t n. Consequently, we achieve a linear computational complexity with respect to the number of instances, making it suitable for large-scale clustering tasks.

Convergence: This paper proposes an alternative optimization algorithm by updating one variable with other variables fixed. Since each sub-optimization can achieve its optimal solution, according to (Bezdek and Hathaway 2003), our algorithm is guaranteed to decrease monotonously during the iteration and converges to a local minimum.

Single-view extension: The proposed algorithm can be easily extended to the single-view scenario by setting $v = 1$, which means it can still generate diverse and informative anchor graphs for a single view. Under the setting of single-view clustering, our proposed method could provide a flexible framework to explore single-view information from different perspectives to form "pseudo" multi-view clustering.

Experiment

Datasets and Baselines

Table 4 lists ten widely used public MVC datasets, including yaleA[1], MSRCV1 (Winn and Jojic 2005), Flower17[2], UCI-Digit[3], Caltech101[4], Reuters, VGGFace[5], CIFAR100[6], EMNIST[7], YTF20 (Huang, Wang, and Lai 2022).

No.	Dataset	Sample	View	Cluster
1	yaleA	165	3	15
2	MSRCV1	210	6	7
3	Flower17	1360	7	17
4	UCI-Digit	2000	3	10
5	Caltech101	9144	5	102
6	Reuters	18758	5	5
7	VGGFace	36287	4	100
8	CIFAR100	60000	4	99
9	YTF20	63896	4	20
10	EMNIST	280000	4	9

Table 4: Description of Datasets.

Ten state-of-the-art MVC methods are compared, including Multi-view k-means Clustering on Big Data **(RMKM)** (Cai, Nie, and Huang 2013), Parameter-free Auto-weighted Multiple Graph Learning **(AMGL)** (Nie et al. 2016), Flexible Multi-View Representation Learning for Subspace Clustering **(FMR)** (Li et al. 2019), Partition Level Multi-view Subspace Clustering **(PMSC)** (Kang et al. 2020a), Binary Multi-View Clustering **(BMVC)** (Zhang et al. 2018), Large-scale Multi-view Subspace Clustering in Linear Time **(LMVSC)** (Kang et al. 2020b), Scalable multi-view subspace clustering with unified anchors **(SMVSC)** (Sun et al. 2021), Fast Multi-View Clustering via Nonnegative and Orthogonal Factorization **(FMCNOF)** (Yang et al. 2020), Fast Parameter-free Multi-view Subspace Clustering with Consensus Anchor Guidance **(FPMVS)** (Wang et al. 2021), Multi-view clustering: a Scalable and Parameter-free Bipartite Graph Fusion Method **(SFMC)** (Li et al. 2020). Note that BMVC, LMVSC, SMVSC, FMCNOF, FPMVS-CAG, and SFMC are designed for large-scale MVC scenarios.

Experiment Setup

For a fair comparison, all the involved hyper-parameters are carefully tuned following the original settings. For our proposed model, α varies in $[10^{-5}, 10^{-1}, 10^{1}, 10^{3}]$, λ varies in $[10^{1}, 10^{3}, 10^{5}]$. The number of anchor choices R is simply set to 4, which means that the number of anchors ranges from k to $4k$, where k is the number of clusters. To mitigate the randomness caused by k-means in the initialization and the post-processing, we repeated 50 times k-means for all compared methods to report the average results. We apply four metrics, including Accuracy (ACC), Normalized Mutual Information (NMI), Purity, and Fscore, to fully assess the clustering quality. Larger values indicate better clustering performance. To assess the efficiency of all approaches, we record the running times of the main algorithm. The reported results in the following section are the optimal ACC across all parameters and its corresponding NMI, Purity, and running time. Besides, our experiments are conducted on a computer with MATLAB R2020b and an Intel Core i7-7820X CPU and 64GB RAM, MATLAB 2020b (64-bit). Our code is released at https://github.com/Jeaninezpp/FDAGF.

Clustering Performance

The clustering performance of ten compared methods and ours on eight benchmark dataset are reported in Table 2 and Table 3. In Table 3, we primarily discuss approaches dedicated to large-scale data since the other compared algorithms typically suffer from "out-of-memory" or need extensive running time. We have the following observations:

- In Table 2, our proposed model achieves the best performance on small-size and medium-size datasets no matter what the evaluation metric is. Especially, our model outperforms the second best method by large margins with 13.03%, 10.95%, 10.37%, and 2.35% of ACC on the four datasets, respectively.
- Furthermore, Table 3 reports clustering performance of seven large-scale oriented methods on six large-scale datasets. We can observe that the clustering performance of the proposed FDAGF is superior to the state-of-the-art competitors w.r.t ACC, demonstrating its strong multi-view clustering capabilities. Moreover, our model achieves considerable improvements on other metrics.
- From these two tables, we can conclude that our algorithm achieves the best or comparable performance compared with the full-graph MVC methods and anchor-graph MVC methods. Especially, our model outperforms LMVSC by 44.24%, 18.10%, 10.37% and 15.55% accuracy on four small and medium size datasets, and by 11.65%, 8.32%, 2.79%, 1.53%, 2.52% and 7.66% on the large-scale datasets. The significant performance difference between ours and LMVSC demonstrates the effectiveness of our flexible anchor fusion strategy.

Furthermore, we restore the full graph S_v^r of each Z_v^r to visualize the effect of flexible anchor graph fusion. Taking dataset uci-digit as an example, Figure 2(a) is the restored full graph S_v^r of each anchor choice over each view. Figure

[1] http://vision.ucsd.edu/content/yale-face-database

[2] https://www.robots.ox.ac.uk/vgg/data/flowers/17/

[3] https://jamesmccaffrey.wordpress.com/2020/10/26/

[4] https://authors.library.caltech.edu/7694/

[5] https://www.robots.ox.ac.uk/vgg/data/vgg_face/

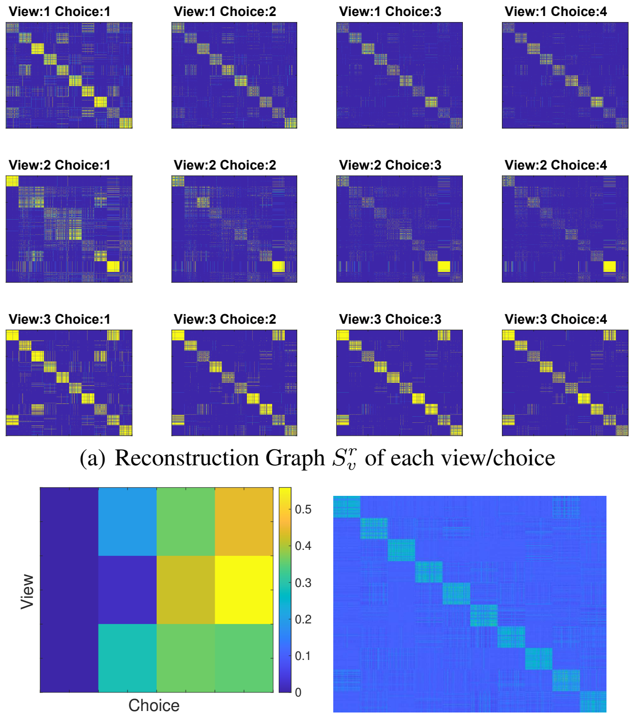
[6] http://www.cs.toronto.edu/kriz/cifar.html

[7] https://www.nist.gov/itl/products-and-services/emnist-dataset

Dataset	Metric	RMKM	AMGL	FMR	PMSC	BMVC	LMVSC	SMVSC	FMCNOF	FPMVS	SFMC	Proposed
yaleA	ACC	0.5939	0.6341	0.6634	0.7606	0.3455	0.4485	0.7152	0.2303	0.7492	0.4242	**0.8909**
	NMI	0.6791	0.6988	0.7582	0.8626	0.4307	0.5126	0.8039	0.3023	0.8161	0.5057	**0.9161**
	Purity	0.6182	0.6573	0.6832	0.7873	0.3758	0.5939	0.7307	0.2303	0.7530	0.4242	**0.9273**
	Fscore	0.4766	0.4122	0.5762	0.7043	0.1919	0.3010	0.6450	0.1842	0.6886	0.2753	**0.8399**
MSRCV1	ACC	0.7143	0.7644	0.7748	0.4745	0.2667	0.7190	0.7051	0.4714	0.7195	0.6048	**0.9000**
	NMI	0.6303	0.7765	0.6948	0.3429	0.0829	0.6077	0.6201	0.3842	0.6569	0.6023	**0.8142**
	Purity	0.7476	0.8045	0.7901	0.4991	0.2714	0.7190	0.7151	0.5048	0.7233	0.6286	**0.9000**
	Fscore	0.5998	0.7028	0.6676	0.3405	0.1601	0.5558	0.5931	0.3385	0.6155	0.5243	**0.8128**
Flower17	ACC	0.2324	0.0970	0.3343	0.2082	0.2699	0.3375	0.2713	0.1743	0.2599	0.0757	**0.4412**
	NMI	0.2207	0.1025	0.3065	0.1913	0.2562	0.3675	0.2578	0.1468	0.2581	0.0787	**0.4319**
	Purity	0.2449	0.1076	0.3474	0.2220	0.2941	0.3985	0.2788	0.1757	0.2638	0.1029	**0.4963**
	Fscore	0.1435	0.1149	0.2009	0.1233	0.1661	0.2255	0.1753	0.1393	0.1729	0.1094	**0.2928**
uci-digit	ACC	0.9115	0.7776	0.7247	0.7092	0.7115	0.7795	0.8261	0.5485	0.8189	0.7635	**0.9350**
	NMI	0.8448	0.8388	0.7033	0.7240	0.6924	0.6735	0.7784	0.5717	0.7876	0.8333	**0.8670**
	Purity	0.9115	0.8071	0.7530	0.7397	0.7290	0.7795	0.8336	0.5625	0.8208	0.7685	**0.9350**
	Fscore	0.8361	0.7503	0.6455	0.6417	0.6250	0.6338	0.7558	0.5040	0.7525	0.7528	**0.8743**

Table 2: Clustering performance comparison (ACC, NMI, Purity, Fscore) of ten MVC methods on four small-size or medium-size datasets. The top performance is highlighted in red bold and the second best is in underlined blue.

2(b) is the visualization of the diverse weights $\boldsymbol{\beta}$. Each block represents the importance of the corresponding choice in the view. Figure 2(c) is the restored full graph by computing $\overline{Z}^\top \overline{Z}$. We can observe the structure on each view and each choice is varied in Figure 2(a). However, after our flexible fusion strategy, the fusion graph reconstructed by $\overline{S} = \overline{Z}^\top \overline{Z}$ shows more clear clustering structure than in Fig. 2(a) and therefore better clustering performance.



(a) Reconstruction Graph S_v^r of each view/choice

(b) Visualization of $\boldsymbol{\beta}$

(c) Fusion Graph $\overline{S}$

Figure 2: Visualization of anchor graph fusion on uci-digit.

Running Time

The total time to obtain the final optimum results consists of the tuning time of the hyper-parameters and the running time under the optimum parameters. Due to space constraints, we visualize the running time w.r.t the optimum parameters on eight datasets in Fig. 3 . As mentioned in computational complexity, our model achieves linear computational complexity. Experimental results further show that our model requires a much shorter running time compared with most compared baselines. Although our model costs more running time than BMVC, LMVSC, SMVSC and FMCNOF on large-scale datasets, our model does not require tuning the number of anchors to achieve optimal results. In brief, our FDAGF does not incur significant tuning time for finding the optimal number of anchors and yet attains superior performance with comparable running time, demonstrating the advantages of our methods regarding time and performance.

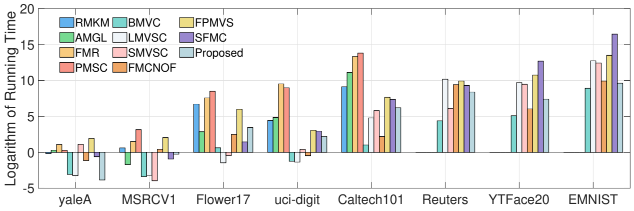


Figure 3: Comparison of relative logarithm running time of ten methods on eight datasets. The y axis is scaled by $\log$ to mitigate the gap between methods. Missing bars indicate that the method suffered an “out-of-memory” error.

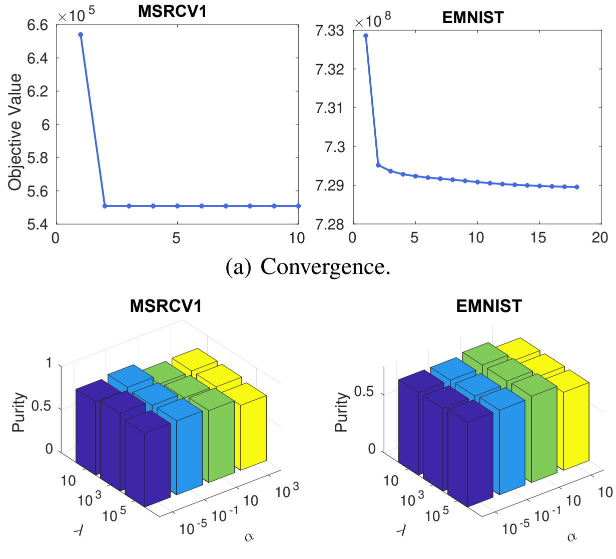
Convergence and Sensitivity

Figure 4(a) depicts the evolution of objective function during the iteration on MSRCV1 and EMNIST. Clearly, from Figure 4(a), the objective monotonically decreases and quickly converges within 20 iterations, which experimentally veri-

Dataset	Metric	BMVC	LMVSC	SMVSC	FMCNOF	FPMVS	SFMC	Proposed
Caltech101	ACC	0.2588	0.1553	0.1427	0.1507	0.2591	0.0876	**0.2718**
	NMI	0.4213	0.3196	0.2822	0.1309	0.3438	0.0002	**0.4479**
	Purity	**0.3915**	0.1949	0.2707	0.1618	0.3389	0.0876	0.3463
	Fscore	**0.2997**	0.1039	0.1217	0.0773	0.2009	0.0548	0.2442
Reuters	ACC	0.5285	0.5287	0.5778	0.3683	0.4458	0.2513	**0.6119**
	NMI	0.2651	0.3316	0.3641	0.1706	0.2850	0.1280	**0.4292**
	Purity	0.5647	0.5803	0.6366	0.4042	0.5358	0.3540	**0.6898**
	Fscore	0.4435	0.4433	0.4682	0.3521	0.3906	0.3390	**0.4754**
VGGFace	ACC	0.0617	0.0609	0.079	0.0198	0.0347	0.0602	**0.0888**
	NMI	0.1426	0.1192	0.1480	0.0581	0.1233	0.0091	**0.1627**
	Purity	0.0715	0.0702	0.0855	0.0350	0.0633	0.0209	**0.1298**
	Fscore	0.0277	0.0247	0.0345	0.0239	0.0314	0.0212	**0.0385**
CIFAR100	ACC	0.0832	0.0953	0.0834	0.0366	0.0729	0.0118	**0.1106**
	NMI	0.1505	0.1540	0.1440	0.0704	0.1362	0.0053	**0.1836**
	Purity	0.0933	0.1090	0.0893	0.0368	0.0764	0.0126	**0.1864**
	Fscore	0.0437	0.0369	0.0447	0.0256	0.0378	0.0198	**0.0448**
YouTubeFace20	ACC	0.5739	0.6726	0.6713	0.3861	0.6308	0.3554	**0.6978**
	NMI	0.7065	0.7678	0.7836	0.4545	0.7430	0.3982	**0.7894**
	Purity	0.6276	0.7340	0.7240	0.4034	0.6492	0.3632	**0.7812**
	Fscore	0.4904	**0.6243**	0.6168	0.2584	0.5781	0.2633	0.6009
EMNIST_digits	ACC	0.6899	0.6574	0.5705	0.3650	0.6224	N/A	**0.7340**
	NMI	**0.7008**	0.5986	0.5250	0.2836	0.5347	N/A	0.6543
	Purity	0.7138	0.6615	0.5763	0.3690	0.6230	N/A	**0.7478**
	Fscore	**0.6138**	0.5515	0.4696	0.2564	0.4929	N/A	0.5888

Table 3: Clustering results of seven large-scale oriented methods on six large-scale datasets. The best result is highlighted in red bold and the second best in underlined blue. "N/A" refers to suffering from an "out-of-memory" error on our device.

fies the convergence of sour proposed model. To investigate the parameter sensitivity study of the involved parameter λ and α, Fig. 4(b) plots the results by grid search on MSRCV1 and EMNIST. We can observe that performance is relatively stable over a given range of parameters.



(a) Convergence.

(b) Parameter sensitivity.

Figure 4: The convergence and parameter sensitivity experiments on dataset MSRCV1 and EMNIST.

Conclusion

In this paper, we propose a flexible and diverse anchor graph fusion framework to alleviate the expenditures of time and space for large-scale multi-view clustering. We consider optimizing a set of anchor graphs with diverse sizes instead of manually searching for the optimal parameter of anchor numbers. Besides, the diverse weights of multi-choice over multi-view are automatically adjusted, which increases the significance of the high-quality anchor graph in hybrid fusion. Then a novel hybrid multi-size anchor graph fusion paradigm is presented to fuse the various size of anchor graphs. Additionally, our proposed method establishes a simple but effective connection between single-view clustering and multi-view clustering. Extensive experiments verify the effectiveness and efficiency of our proposed FDAGF.

Acknowledgments

This work was supported by the National Key R&D Program of China 2020AAA0107100 and the National Natural Science Foundation of China(project no. 61872377, 61922088, 61976196 and 61872371).

References

Affeldt, S.; Labiod, L.; and Nadif, M. 2020. Spectral clustering via ensemble deep autoencoder learning (SC-EDAE). *Pattern Recognition*, 108: 107522.

Bezdek, J. C.; and Hathaway, R. J. 2003. Convergence of Alternating Optimization. *Neural Parallel Sci. Comput.*, 11(4): 351–368.

Cai, X.; Nie, F.; and Huang, H. 2013. Multi-view k-means clustering on big data. In *Twenty-Third International Joint conference on artificial intelligence*. Citeseer.

Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–594.

Huang, D.; Wang, C.-D.; and Lai, J.-H. 2022. Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. *arXiv preprint arXiv:2203.11572*.

Kang, Z.; Zhao, X.; Peng, C.; Zhu, H.; Zhou, J. T.; Peng, X.; Chen, W.; and Xu, Z. 2020a. Partition level multiview subspace clustering. *Neural Networks*, 122: 279–288.

Kang, Z.; Zhou, W.; Zhao, Z.; Shao, J.; Han, M.; and Xu, Z. 2020b. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4412–4419.

Li, L.; Wang, S.; Liu, X.; Zhu, E.; Shen, L.; Li, K.; and Li, K. 2022. Local Sample-Weighted Multiple Kernel Clustering With Consensus Discriminative Graph. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.

Li, R.; Zhang, C.; Hu, Q.; Zhu, P.; and Wang, Z. 2019. Flexible Multi-View Representation Learning for Subspace Clustering. In *IJCAI*, 2916–2922.

Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2020. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1): 330–344.

Li, Y.; Nie, F.; Huang, H.; and Huang, J. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Liu, J.; Wang, C.; Gao, J.; and Han, J. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM international conference on data mining*, 252–260. Society for Industrial and Applied Mathematics.

Liu, S.; Wang, S.; Zhang, P.; Xu, K.; Liu, X.; Zhang, C.; and Gao, F. 2022a. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7576–7584.

Liu, W.; He, J.; and Chang, S.-F. 2010. Large graph construction for scalable semi-supervised learning. In *ICML*.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Tang, C.; Yin, J.; Shen, D.; Wang, H.; and Gao, W. 2018. Late fusion incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 41(10): 2410–2423.

Liu, Y.; Yang, X.; Zhou, S.; and Liu, X. 2022b. Simple Contrastive Graph Clustering. *arXiv preprint arXiv:2205.07865*.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Nie, F.; Li, J.; Li, X.; et al. 2016. Parameter-free auto-weighted multiple graph learning: a framework for multi-view clustering and semi-supervised classification. In *IJCAI*, 1881–1887.

Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 977–986.

Sun, M.; Zhang, P.; Wang, S.; Zhou, S.; Tu, W.; Liu, X.; Zhu, E.; and Wang, C. 2021. Scalable multi-view subspace clustering with unified anchors. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3528–3536.

Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; and Zhu, E. 2021. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31: 556–568.

Winn, J. M.; and Jojic, N. 2005. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, 756–763. IEEE Computer Society.

Yang, B.; Zhang, X.; Nie, F.; Wang, F.; Yu, W.; and Wang, R. 2020. Fast multi-view clustering via nonnegative and orthogonal factorization. *IEEE Transactions on Image Processing*, 30: 2575–2586.

Yang, X.; Hu, X.; Zhou, S.; Liu, X.; and Zhu, E. 2022. Interpolation-Based Contrastive Learning for Few-Label Semi-Supervised Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–12.

Yang, Y.; and Wang, H. 2018. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2): 83–107.

Zhang, C.; Wang, S.; Liu, J.; Zhou, S.; Zhang, P.; Liu, X.; Zhu, E.; and Zhang, C. 2021a. Multi-view clustering via deep matrix factorization and partition alignment. In *Proceedings of the 29th ACM International Conference on Multimedia*, 4156–4164.

Zhang, J.; Li, L.; Wang, S.; Liu, J.; Liu, Y.; Liu, X.; and Zhu, E. 2022a. Multiple Kernel Clustering with Dual Noise Minimization. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, 3440–3450. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392037.

Zhang, P.; Liu, X.; Xiong, J.; Zhou, S.; Zhao, W.; Zhu, E.; and Cai, Z. 2022b. Consensus One-Step Multi-View Subspace Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 34(10): 4676–4689.

Zhang, T.; Liu, X.; Gong, L.; Wang, S.; Niu, X.; and Shen, L. 2021b. Late Fusion Multiple Kernel Clustering with Local Kernel Alignment Maximization. *IEEE Transactions on Multimedia*, 1–1.

Zhang, T.; Liu, X.; Zhu, E.; Zhou, S.; and Dong, Z. 2022c. Efficient Anchor Learning-based Multi-view Clustering–A Late Fusion Method. In *Proceedings of the 30th ACM International Conference on Multimedia*, 3685–3693.

Zhang, Z.; Liu, L.; Shen, F.; Shen, H. T.; and Shao, L. 2018. Binary multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1774–1782.

Zhao, H.; Ding, Z.; and Fu, Y. 2017. Multi-view clustering via deep matrix factorization. In *Thirty-first AAAI conference on artificial intelligence*.