Metadata of the article that will be visualized in OnlineFirst

1	Article Title	**Fine-grained multi-view clustering with robust multi-prototypes representation**	
2	Article Sub-Title		
3	Article Copyright - Year	**Springer Science+Business Media, LLC, part of Springer Nature 2022** **(This will be the copyright line in the final PDF)**	
4	Journal Name	Applied Intelligence	
5	Corresponding Author	Family Name	**Yin**
6		Particle	
7		Given Name	**Hongwei**
8		Suffix	
9		Organization	Huzhou University
10		Division	School of Information Engineering and Zhejiang Province Key Laboratory of Smart Management, Application of Modern Agricultural Resources
11		Address	Huzhou, 313000, China
12		e-mail	02713@zjhu.edu.cn
13	Corresponding Author	Family Name	**Hu**
14		Particle	
15		Given Name	**Wenjun**
16		Suffix	
17		Organization	Huzhou University
18		Division	School of Information Engineering and Zhejiang Province Key Laboratory of Smart Management, Application of Modern Agricultural Resources
19		Address	Huzhou, 313000, China
20		e-mail	hoowenjun@foxmail.com
21	Author	Family Name	**Wang**
22		Particle	
23		Given Name	**Guixiang**
24		Suffix	
25		Organization	Huzhou University
26		Division	School of Information Engineering and Zhejiang Province Key Laboratory of Smart Management, Application of Modern Agricultural Resources

27		Address	Huzhou, 313000, China
28		e-mail	gxxgwang@163.com
29	Author	Family Name	**Zhang**
30		Particle	
31		Given Name	**Zhao**
32		Suffix	
33		Organization	Hefei University of Technology
34		Division	School of Computer Science and Information Engineering, Key Laboratory of Knowledge Engineering with Big Data (Ministry of Education)
35		Address	Hefei, 230009, China
36		e-mail	cszzhang@gmail.com

37	Schedule	Received	
38		Revised	
39		Accepted	14 June 2022

40	Abstract	Multi-view clustering is a hot research topic that improves clustering performance by leveraging complementary information from multiple views. Recently, many multi-view clustering methods have been proposed. Most of them take the entire sample space as a fusion object and treat the local structures within each view equally. This paradigm is considered coarse-grained information fusion. However, in many real-world applications, different local structures with strong or weak clustering capacities could coexist in one view. To fully exploit valuable information of local structures, it is necessary to distinguish these local structures with different clustering capacities. In this paper, we propose a novel fine-grained multi-view clustering method. First, the sample space of each view is divided into many sub-clusters by using multi-prototypes representation. Second, the robustness of the multi-prototypes representation is enhanced by reducing the overlap between sub-clusters, which can reduce the effect of noise data. Finally, each sub-cluster's contribution weights are automatically assigned based on its clustering capacity. In addition, the robust multi-prototypes representation, the fine-grained multi-view fusion, and the clustering process are integrated into a unified framework. An effective alternating optimization algorithm is adopted to solve the objective function. Extensive experiments on two toy datasets and several real-world datasets prove that our method outperforms the traditional methods in clustering accuracy.

41	Keywords separated by ' - '	Multi-view clustering - Fine-grained fusion - Multi-prototypes representation - Sub-cluster structure

42 Foot note information

Applied Intelligence
https://doi.org/10.1007/s10489-022-03898-2

Fine-grained multi-view clustering with robust multi-prototypes representation

Q1 Hongwei Yin[1] · Guixiang Wang[1] · Wenjun Hu[1] · Zhao Zhang[2]

Accepted: 14 June 2022

Abstract

Multi-view clustering is a hot research topic that improves clustering performance by leveraging complementary information from multiple views. Recently, many multi-view clustering methods have been proposed. Most of them take the entire sample space as a fusion object and treat the local structures within each view equally. This paradigm is considered coarse-grained information fusion. However, in many real-world applications, different local structures with strong or weak clustering capacities could coexist in one view. To fully exploit valuable information of local structures, it is necessary to distinguish these local structures with different clustering capacities. In this paper, we propose a novel fine-grained multi-view clustering method. First, the sample space of each view is divided into many sub-clusters by using multi-prototypes representation. Second, the robustness of the multi-prototypes representation is enhanced by reducing the overlap between sub-clusters, which can reduce the effect of noise data. Finally, each sub-cluster's contribution weights are automatically assigned based on its clustering capacity. In addition, the robust multi-prototypes representation, the fine-grained multi-view fusion, and the clustering process are integrated into a unified framework. An effective alternating optimization algorithm is adopted to solve the objective function. Extensive experiments on two toy datasets and several real-world datasets prove that our method outperforms the traditional methods in clustering accuracy.

Keywords Multi-view clustering · Fine-grained fusion · Multi-prototypes representation · Sub-cluster structure

1 Introduction

At present, multi-view clustering is a hot research topic in machine learning and pattern recognition and has been widely used in practice [1–5]. Compared with single-view clustering, multi-view clustering enables more accurate data partitioning by exploiting the complementary information from multiple views. Over the past decades, many multi-view clustering algorithms have been proposed based on different data representations. These algorithms can be roughly divided into four categories including multi-view subspace clustering, multi-view nonnegative matrix factorization, graph-based multi-view clustering, and deep multi-view clustering.

Subspace clustering expresses each sample with a linear combination of the samples, namely, self-representation. Multi-view subspace clustering assumes that multiple views originate from one shared latent representation, which can reveal the common latent structure shared by different self-representations [6–9]. Another research direction focuses on multi-view clustering using joint nonnegative matrix decomposition (NMF) due to its low computational cost and excellent performance in latent factor extraction. These NMF-based methods enforce the nonnegative embeddings of different views to a consensus indicator matrix, which directly reveals the consistent clustering result [10–12].

✉ Hongwei Yin
02713@zjhu.edu.cn

✉ Wenjun Hu
hoowenjun@foxmail.com

Guixiang Wang
gxxgwang@163.com

Zhao Zhang
cszzhang@gmail.com

[1] School of Information Engineering and Zhejiang Province Key Laboratory of Smart Management, Application of Modern Agricultural Resources, Huzhou University, Huzhou, 313000, China

[2] School of Computer Science and Information Engineering, Key Laboratory of Knowledge Engineering with Big Data (Ministry of Education), Hefei University of Technology, Hefei, 230009, China

Graph-based methods are particularly suitable for data with non-Gaussian or non-spherical distributions. Considering each sample as a node in an undirected graph, each edge between two nodes is weighted by a suitably defined distance measure. According to specific criteria for each particular algorithm, connected sub-graphs of various shapes and sizes can be detected. Graph-based multi-view clustering methods typically find a unified graph by fusing the input graphs of all views first. Then, an additional clustering algorithm, such as k-means, is carried out on the unified graph adjacency matrix to obtain a consistent clustering result [13–15].

To address data with a non-linear structure, deep learning-based methods have more advantages. A deep neural network (DNN) composed of multiple nonlinear transformations can learn feature representation better than the traditional shallow model. Inspired by the success of deep single-view clustering in recent years [16–18], many researchers integrate deep learning into multi-view clustering tasks, such as the popular autoencoder and generative adversarial networks. These methods design a view-specific encoder to extract the high-level information in each view, where the fusion mechanism based on the fusion of latent representations has not been substantially improved [19–22].

In fact, following the principle of consistency and complementarity, these traditional multi-view clustering methods achieve impressive results. In the fusion phase, almost all traditional methods treat the entire sample space of each view as a whole and assign a contribution weight to each view. In terms of complementarity, the contribution weights should be learned automatically to describe the different clustering capacities of each view [5, 23–26]. This fusion paradigm is considered coarse-grained information fusion. However, in many real-world applications, the clustering capacities of different local structures within each view are not ideally consistent. Different local structures with strong or weak clustering capacity could coexist in one view. The coarse-grained fusion paradigm takes the entire sample space of each view as the fusion object, which cannot distinguish these local structures with different clustering capacities. As a result, the valuable information of local structures in a poor view cannot be fully used to improve the final multi-view clustering performance. Additionally, misleading information of local structures in a quality view can reduce the final clustering performance. Therefore, to address this issue and improve the performance of multi-view clustering, it is necessary to refine the granularity of the fusion object from the entire sample space to the local structure within each view.

Obviously, there are two natural and easily obtained fine-grained local structures, samples, and clusters, that can be used as fusion objects. Note that when samples are used as fusion objects, it is difficult to accurately obtain the clustering capacity of an individual sample [27]. When clusters are used as fusion objects, it is difficult to capture non-spherical clusters in the original data [28]. In this paper, a novel fine-grained multi-view clustering method (FgMVC) is proposed. As shown in Fig. 1, instead of natural samples and clusters, the data of each view are split into many small sub-clusters based on multi-prototypes representation. Note that each sub-cluster has its own prototype. Then, each view is represented as multiple prototypes and a corresponding connection probability matrix. By reducing the overlap between different sub-clusters, the robustness of multi-prototypes representation is enhanced, which can reduce the effect of noise data. Finally, a structured consensus probability matrix is learned by fusing the complementary information from multiple sub-clusters of different views. The main contributions of this paper are summarized as follows:

1) This paper proposes a novel fine-grained multi-view clustering method, where the fusion objects are refined into sub-clusters within each view. Instead of natural samples and clusters, we first split the sample space of each view into many small sub-clusters based on multi-prototypes representation. Therefore, non-spherical clusters in the original data can be adequately fitted by utilizing multi-prototypes representation. In addition, the contribution weight is automatically assigned according to the clustering capacity of each sub-cluster, which can better strengthen the complementarity between multiple views.
2) Fine-grained multi-fusion mechanisms cause two problems. First, the effect of noise samples on clustering is amplified. Second, the boundary between sub-clusters is not clear. To solve these problems, we minimize the degree of overlap between two sub-clusters by maximizing the distance between prototypes of different sub-clusters. This low-overlap multi-prototypes representation has strong robustness because reducing the overlap between sub-clusters can judge the division of samples more clearly. Moreover, the robust multi-prototypes representation, the multi-view information fusion, and the clustering process are integrated into a unified framework.
3) An effective alternating optimization algorithm is adopted to solve the objective function. In addition, the computational complexity of the algorithm is analyzed in detail, and the convergence of the algorithm is proved theoretically and numerically.
4) To verify the effectiveness of our method, we conduct experiments on two synthetic datasets and six real-world datasets. Compared with several typical coarse-grained multi-view clustering methods, our method

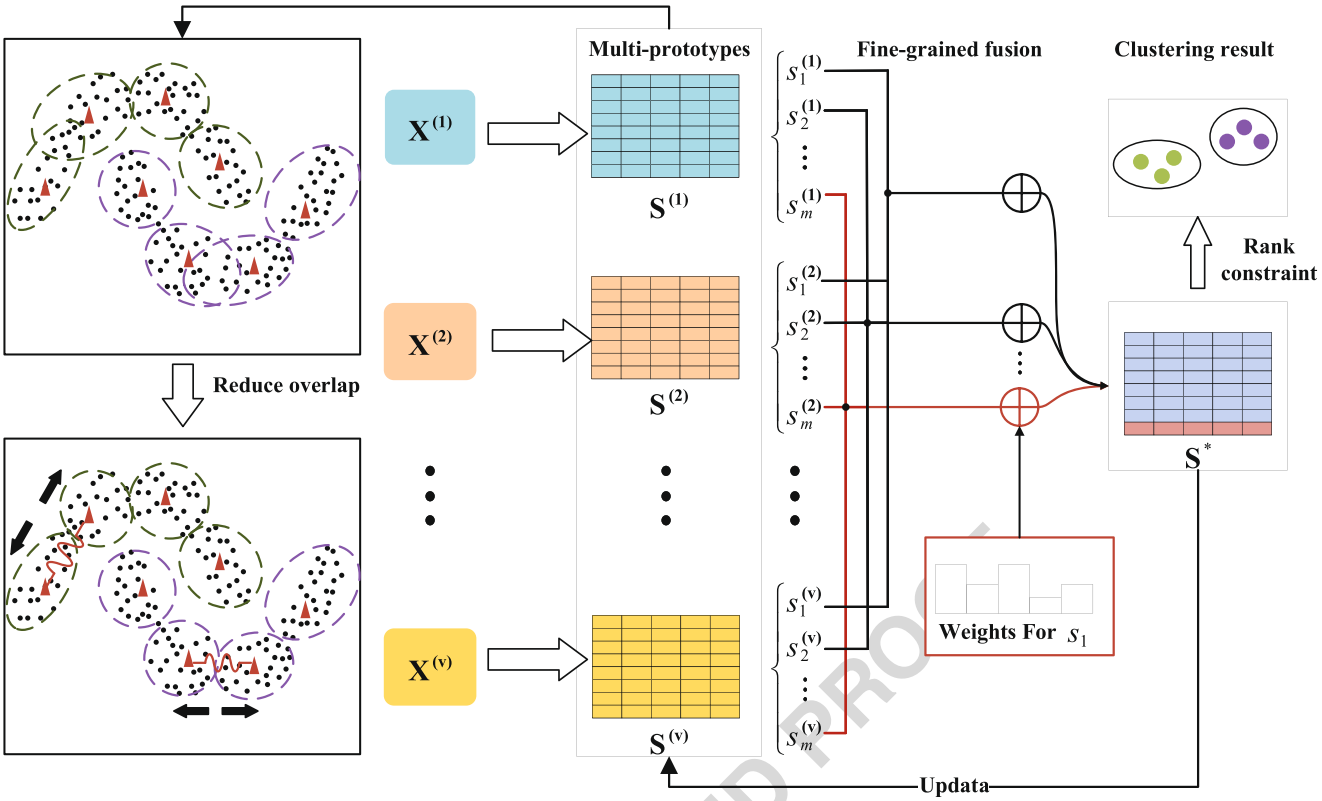


Fig. 1 The main framework of FgMVC. First, the sample space of each view is divided into m sub-clusters by using multi-prototypes representation, where the sub-cluster structure is represented by a connection probability matrix $\boldsymbol{S}$. For each view, the distance between prototypes of different sub-clusters is maximized to reduce the overlap degree between sub-clusters. Reducing the overlap between sub-clusters can enhance the differentiation between sub-clusters, which can obtain robust multi-prototypes representation. Second, each sub-cluster in each view is automatically assigned a contribution weight $w_j^{(p)}$ according to its clustering capacity. Finally, the consensus connection probability matrix $\boldsymbol{S}^*$ is obtained by fusing multiple probability connection matrices of different views, and the clustering is realized by adding a rank constraint to the matrix $\boldsymbol{S}^*$

shows superior performance on multi-view clustering tasks.

The rest of this paper is organized as follows. Section 2 briefly introduces the related work about multi-view clustering, K-multiple-means, and AKM^{3}C. Section 3 describes our proposed fine-grained multi-view clustering and its optimization algorithm in detail. In Section 4, we provide the corresponding theoretical analysis of our algorithm. Section 5 reports the experimental results on two synthetic and six real-world datasets. Finally, we conclude this paper in Section 6.

In this paper, all matrices are written in bold uppercase, all vectors are written in bold lowercase and scalars are written in lowercase. For a matrix $\boldsymbol{M}$, m_{ij} is the element in the i-th row and the j-th column. $\boldsymbol{m}_j$ is the j-th column vector. The Frobenius norm of $\boldsymbol{M}$ is expressed by $\|\boldsymbol{M}\|_F$. The trace of $\boldsymbol{M}$ is expressed by $Tr(\boldsymbol{M})$. For a vector $\boldsymbol{u}$, the ℓ_2-norm of $\boldsymbol{u}$ is expressed by $\|\boldsymbol{u}\|_2$. Vector $\mathbf{1}$ denotes a column vector with all the elements as one. For vector $\boldsymbol{u}$ and matrix $\boldsymbol{M}$, $\boldsymbol{u} \geq 0$ and $\boldsymbol{M} \geq 0$ represent that all the elements of $\boldsymbol{u}$ and $\boldsymbol{M}$ are equal to or larger than zero. The main notations are summarized in Table 1.

2 Related work

2.1 Multi-view clustering

In recent years, a large number of multi-view clustering methods have been proposed, most of which adopt a coarse-grained fusion paradigm. Whether based on a graph model, nonnegative embedding, or subspace learning, the fusion object in these methods is the entire sample space of each view. In [13], a sparse representation method is first used to construct the similarity-induced graph matrix of each view. Then, it proposes a multiple graph fusion model, in which each view is weighted automatically. To make use of the duality between samples and features of multi-view data, Huang et al. [14] improved the graph fusion model into a bipartite graph fusion model. Under the joint nonnegative matrix decomposition framework [29], a

Table 1 Main notations

n	Number of samples
c	Number of clusters
k	Number of neighbors
m	Number of sub-clusters
v	Number of views
$d^{(p)}$	Dimension of features in p-th view
$\mathcal{X} = \{\boldsymbol{X}^{(p)}\}_{p=1}^{v}$	Multi-view data
$\boldsymbol{X}^{(p)} \in \mathbb{R}^{n \times d^{(p)}}$	The p-th view data matrix
$\boldsymbol{A}^{(p)} \in \mathbb{R}^{m \times d^{(p)}}$	The p-th view prototype matrix
$\boldsymbol{S}^{(p)} \in \mathbb{R}^{n \times m}$	The p-th view connection probability matrix
$\|\boldsymbol{X}\|_F$	Frobenius norm of matrix $\boldsymbol{X}$
$Tr(\boldsymbol{X})$	Matrix trace of matrix $\boldsymbol{X}$
$\boldsymbol{I}$	Identity matrix

multi-view clustering method by integrating nonnegative embedding and spectral embedding is proposed in [11], which inherits the advantages of both graph-based and matrix decomposition methods.

When the clustering capacities of different local structures are inconsistent, the coarse-grained multi-view clustering methods cannot achieve the optimal complementary fusion. In [30], a groupware multi-view fusion approach is proposed. In this approach, nonnegative spectral clustering is utilized to divide samples into groups, where a group can be identified as the intermediate representation between clusters and individual samples. However, the group discovery process and clustering process in this method are two separate steps. In [28], Zhao et al. assigned a weight to each cluster within each view. The weight of a cluster is learned by making the cluster weight proportional to the average distance between the cluster and other clusters. When the number of natural clusters is small or the original data structure is non-spherical, this method does not work well. In [27], Yu et al. assumed that some samples may be corrupted or lost in partial views. A fine-grained similarity fusion for multi-view spectral clustering is proposed by automatically assigning weights to different views between two samples. Different from these above methods, we refine the fusion object from the entire sample space of each view into sub-clusters within each view by using multi-prototypes representation [31]. Additionally, the distance between different prototypes within each view is maximized to make the clustering capacity gap between sub-clusters more significant. Moreover, the robust multi-prototypes representation, multi-view sub-cluster fusion, and clustering process are integrated into a unified framework.

2.2 K-multiple-means

Given a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$, K-multiple-means (KMM) [31] first splits the data space into m sub-clusters. Each sub-cluster has a prototype. Then, a Laplacian rank constraint is imposed on the connection probability matrix to obtain the specified c clusters. The objective function of KMM is as follows:

$$\begin{aligned} &\min_{\boldsymbol{A},\boldsymbol{S}} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij} \left\|\boldsymbol{x}_i - \boldsymbol{a}_j\right\|_2^2 + \gamma \|\boldsymbol{S}\|_F^2 \\ &s.t.\ \boldsymbol{S}\boldsymbol{1} = \boldsymbol{1},\ \boldsymbol{S} \geq 0,\ \boldsymbol{S} \in \Omega. \end{aligned} \tag{1}$$

where $\boldsymbol{a}_j$ is the prototype of the j-th sub-cluster, and s_{ij} is the connection probability between the i-th sample and the j-th prototype. Note that the j-th column vector $\boldsymbol{s}_j$ of the matrix $\boldsymbol{S}$ indicates the connection probability of each sample to the j-th sub-cluster, and $\boldsymbol{S}$ is a connection probability matrix-induced bipartite graph. With a Laplacian rank constraint $\boldsymbol{S} \in \Omega$, the bipartite graph is guaranteed to have c connected components. The regularization parameter γ controls the sparsity of the bipartite graph, which avoids the trivial solution to $\boldsymbol{S}$.

Compared with the traditional K-means, KMM greatly improves the capacity of K-means to capture non-spherical clusters by using multi-prototypes representation. The work on multi-view clustering based on multi-prototypes representation is relatively limited. Inspired by KMM, this paper extends multi-prototypes representation to multi-view clustering tasks. In addition, we improve this method to make it more suitable for multi-view clustering tasks.

2.3 AKM^{3}C: adaptive K-multiple-means for multi-view clustering

As a multi-view extension method of KMM, AKM^{3}C can also capture non-spherical clusters by utilizing multi-prototypes representation. To distinguish the importance of different views, AKM^{3}C uses the multi-view combination weights strategy to automatically assign a weight to each view so that complementary information of different views can be fused and consistent results can be obtained. Its objective function is:

$$\begin{aligned} &\min_{\boldsymbol{A}^{(p)},\boldsymbol{S}} \sum_{p=1}^{v}\sum_{i=1}^{n}\sum_{j=1}^{m} w_p s_{ij} \left\|\boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)}\right\|_2^2 + \gamma \|\boldsymbol{S}\|_F^2 \\ &s.t.\ \boldsymbol{S}\boldsymbol{1} = \boldsymbol{1},\ \boldsymbol{S} \geq 0,\ \boldsymbol{S} \in \Omega \end{aligned} \tag{2}$$

where w_p is the weight of the p-th view and $\boldsymbol{a}_j^{(p)}$ is the j-th prototype of the p-th view. To obtain consistent results, AKM^{3}C takes $\boldsymbol{S}$ as a shared connection probability matrix, where s_{ij} is the connection probability from the i-th sample to the j-th prototype, and $\boldsymbol{S} \in \Omega$ denotes that the bipartite graph has c connected components.

AKM^{3}C extends KMM to multi-view clustered tasks. It treats the entire sample space of each view as a fusion object. The contribution weights are adaptively assigned based on the clustering capacity of each view, where each sub-cluster in one view has the same contribution

weight. AKM^{3}C belongs to the coarse-grained fusion paradigm, which cannot distinguish sub-cluster structures with different clustering capacities. Moreover, the multi-prototypes representation used by AKM^{3}C causes the problem of overlapping between sub-clusters. To solve these problems, we propose a novel fine-grained multi-view clustering method that treats each sub-cluster as a fusion object. The contribution weights are allocated adaptively based on the clustering capacity of each sub-cluster. Therefore, the valuable information of sub-cluster structures in a poor view can be fully used to improve the clustering performance. The misleading information of sub-cluster structures in a quality view will be weakened. To solve the problem of overlapping sub-clusters, we increase the distance between prototypes to obtain robust multi-prototypes representation. To our knowledge, this work is the first multi-view clustering method using a fine-grained fusion paradigm.

3 Fine-grained multi-view clustering with robust multi-prototypes representation

3.1 Model formulation

Given a multi-view dataset $\{\boldsymbol{X}^{(p)}\}_{p=1}^{v}$, where the p-th view is denoted as $\boldsymbol{X}^{(p)} \in \mathbb{R}^{n\times d^{(p)}}$. As in K-multiple-means, the entire sample space of each view is split into m sub-clusters, where each sub-cluster has its own prototype. Then, in the p-th view, m prototypes form a prototype matrix, e.g., $\boldsymbol{A}^{(p)} = \left[\boldsymbol{a}_1^{(p)}, \boldsymbol{a}_2^{(p)}, \ldots, \boldsymbol{a}_m^{(p)}\right]$. The multi-prototypes representation problem of multi-view data is written as:

$$\begin{aligned} &\min_{\boldsymbol{A}^{(p)},\boldsymbol{S}^{(p)}} \sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\sum_{i=1}^{n} s_{ij}^{(p)} \left\|\boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)}\right\|_2^2} + \gamma \left\|\boldsymbol{S}^{(p)}\right\|_F^2 \\ &s.t.\ \boldsymbol{S}^{(p)}\mathbf{1} = \mathbf{1},\ \boldsymbol{S}^{(p)} \geq 0 \end{aligned} \tag{3}$$

where $s_{ij}^{(p)}$ represents the connection probability between the i-th sample and the j-th prototype in p-th view, which is negatively correlated with the distance from the i-th sample to the j-th prototype in the p-th view. γ is a regularization parameter and controls the sparsity of the connection probability matrix. In (3), the multi-prototypes representation of each view can be obtained independently. Because the overlap between sub-clusters is reduced, the discriminability between sub-clusters can be enhanced, which can improve the robustness of multi-prototypes representation. To enhance the discriminability in clustering capacity between sub-clusters, we attempt to reduce their overlap by maximizing the distance between prototypes. The robust multi-prototypes representation is as follows:

$$\begin{aligned} &\min_{\boldsymbol{A}^{(p)},\boldsymbol{S}^{(p)}} \sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\sum_{i=1}^{n} s_{ij}^{(p)} \frac{\left\|\boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)} - \boldsymbol{a}_j^{(p)}\right\|_2^2}} + \gamma \left\|\boldsymbol{S}^{(p)}\right\|_F^2 \\ &s.t.\ \boldsymbol{S}^{(p)}\mathbf{1} = \mathbf{1},\ \boldsymbol{S}^{(p)} \geq 0 \end{aligned} \tag{4}$$

where the greater the distance $\|\boldsymbol{a}_t^{(p)} - \boldsymbol{a}_j^{(p)}\|_2^2$ between two prototypes $\boldsymbol{a}_t^{(p)}$ and $\boldsymbol{a}_j^{(p)}$ is, the less overlap there is between two sub-clusters. Based on the robust multi-prototypes representation, different sub-clusters within each view are separated well. For multi-view clustering tasks, such improvement can help to distinguish sub-clusters with different clustering capacities.

In the connection probability matrices $\{\boldsymbol{S}^{(p)}\}_{p=1}^{v}$, the column vectors represent the sub-cluster distribution within each view, respectively. Specifically, the j-th column $\boldsymbol{s}_j^{(p)}$ of matrix $\boldsymbol{S}^{(p)}$ indicates the possibility of each sample belonging to the j-th sub-cluster in the p-th view. To achieve fine-grained multi-view fusion, the column vectors of the corresponding sub-cluster from different views are unified into a consensus connection probability vector $\boldsymbol{s}_j^*$. For each sub-cluster, the consensus connection probability vector $\boldsymbol{s}_j^*$ is obtained as follows:

$$\min \sum_{p=1}^{v}\left\|\boldsymbol{s}_j^{(p)} - \boldsymbol{s}_j^*\right\|_2^2 \ s.t.\ \mathbf{1}^T\boldsymbol{s}_j^* = 1,\ \boldsymbol{s}_j^* \geq 0 \tag{5}$$

Indeed, the explicit sub-cluster distribution of each view is not the final result we need. In our model, the key is to obtain the consensus connection probability vector of each sub-cluster. Therefore, instead of a linear model combination, we jointly learn the multi-prototypes representation and the consensus connection probability vector by replacing $\boldsymbol{s}_j^{(p)}$ with $\boldsymbol{s}_j^*$ as follows:

$$\begin{aligned} &\min_{\boldsymbol{A}^{(p)},\boldsymbol{S}^*} \sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\frac{\sum_{i=1}^{n} s_{ij}^*\left\|\boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)} - \boldsymbol{a}_j^{(p)}\right\|_2^2}} + \gamma \left\|\boldsymbol{S}^*\right\|_F^2 \\ &s.t.\ \boldsymbol{S}^*\mathbf{1} = \mathbf{1},\ \boldsymbol{S}^* \geq 0 \end{aligned} \tag{6}$$

where $\boldsymbol{S}^*$ is the consensus connection probability matrix. Let $\Gamma(\boldsymbol{x},\boldsymbol{a}) = \frac{\sum_{i=1}^{n} s_{ij}^*\left\|\boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)} - \boldsymbol{a}_j^{(p)}\right\|_2^2}$, the Lagrange function of (6) is:

$$\sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\Gamma(\boldsymbol{x},\boldsymbol{a})} + \gamma\left\|\boldsymbol{S}^*\right\|_F^2 + \Phi(\eta,\boldsymbol{S}^*) \tag{7}$$

where $\Phi(\eta,\boldsymbol{S}^*)$ is the formalized term derived from the constraint term, and η is the Lagrange multiplier. We take

the derivative of S^* in (7), and set the value to zero. Then, the following equation can be obtained:

$$\sum_{p=1}^{v}\sum_{j=1}^{m}\frac{\partial\sqrt{\Gamma(\boldsymbol{x},\boldsymbol{a})}}{\partial S^*}+\frac{\partial\left\|S^*\right\|_F^2}{\partial S^*}+\frac{\partial\Phi(\eta,S^*)}{\partial S^*}=0 \tag{8}$$

Obviously, $\frac{\partial\sqrt{\Gamma(x,a)}}{\partial S^*}=\frac{1}{2\sqrt{\Gamma(x,a)}}\frac{\partial\Gamma(x,a)}{\partial S^*}$, and (8) is rewritten as:

$$\sum_{p=1}^{v}\sum_{j=1}^{m}\frac{1}{2\sqrt{\Gamma(\boldsymbol{x},\boldsymbol{a})}}\frac{\partial\Gamma(\boldsymbol{x},\boldsymbol{a})}{\partial S^*}+\frac{\partial\left\|S^*\right\|_F^2}{\partial S^*}+\frac{\partial\Phi(\eta,S^*)}{\partial S^*}=0 \tag{9}$$

Let $w_j^{(p)}=\frac{1}{2\sqrt{\Gamma(\boldsymbol{x},\boldsymbol{a})}}$, and (8) is equivalent to

$$\sum_{p=1}^{v}\sum_{j=1}^{m}w_j^{(p)}\frac{\partial\Gamma(\boldsymbol{x},\boldsymbol{a})}{\partial S^*}+\frac{\partial\left\|S^*\right\|_F^2}{\partial S^*}+\frac{\partial\Phi(\eta,S^*)}{\partial S^*}=0 \tag{10}$$

Note that $\Gamma(\boldsymbol{x},\boldsymbol{a})=\frac{\sum_{i=1}^{n}s_{ij}^*\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}$; therefore, the following equation can be obtained:

$$w_j^{(p)}=\frac{1}{2}\sqrt{\frac{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{i=1}^{n}s_{ij}^*\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}} \tag{11}$$

where $w_j^{(p)}$ is set to a fixed value, and (10) is also the solution to (12). Therefore, the (6) can be transformed into (12).

$$\begin{aligned}&\min_{\boldsymbol{A}^{(p)},S^*}\sum_{p=1}^{v}\sum_{j=1}^{m}w_j^{(p)}\frac{\sum_{i=1}^{n}s_{ij}^*\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}+\gamma\left\|S^*\right\|_F^2\\&s.t.\ S^*\mathbf{1}=\mathbf{1},\ S^*\geq 0\end{aligned} \tag{12}$$

Note that $w_j^{(p)}$ actually describes the clustering capacity of each sub-cluster. $\sum_{t=1,t\neq j}^{m}\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2$ is the sum of distances between the j-th prototype and other prototypes in the p-th view. When this value is large, it indicates that the j-th sub-cluster is far away from other sub-clusters. $\sum_{i=1}^{n}s_{ij}^*\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2$ is the sum of distances between the j-th prototype with samples in the p-th view. When this value is small, it indicates that samples belonging to the j-th sub-cluster are close to the j-th prototype.

For each sub-cluster, the former measures how much it overlaps with other sub-clusters, while the latter measures the degree of compactness within itself. The less overlap a sub-cluster has, the stronger its clustering capacity is. Additionally, the more compact a sub-cluster is, the stronger its clustering capacity is. Using this data-driven weighting scheme, each sub-cluster can be automatically weighted based on its clustering capacity without introducing any additional parameters.

The consensus connection probability matrix S^* is an induced bipartite graph representing the connectivity between samples and prototypes. We aim to partition all samples into c natural clusters, which is equivalent to obtaining c connected components of the bipartite graph. Inspired by [31], we define a symmetric adjacency matrix $\boldsymbol{P}=\begin{bmatrix}\boldsymbol{0} & S^*\\ S^{*T} & \boldsymbol{0}\end{bmatrix}$ and a diagonal degree matrix $\boldsymbol{D}$ with $d_{ii}=\sum_{j=1}^{n+m}p_{ij}$. Then, $\boldsymbol{L}_{S^*}=\boldsymbol{I}-\boldsymbol{D}^{-1/2}\boldsymbol{P}\boldsymbol{D}^{-1/2}$ is the normalized Laplacian matrix associated with S^*.

Theorem 1 [32] *The multiplicity c of the eigenvalue 0 of the normalized Laplacian matrix $\boldsymbol{L}_{S^*}$ is equal to the number of connected components in the graph associated with S^*.*

According to Theorem 1, when the rank of the normalized Laplacian matrix is equal to $(n+m)-c$, the associated graph S^* has c connected components. Therefore, the c connected components can be obtained directly from the induced bipartite graph by imposing a rank constraint on its Laplacian matrix. The objective function of multi-view clustering is as follows:

$$\begin{aligned}&\min_{\boldsymbol{A}^{(p)},S^*}\sum_{p=1}^{v}\sum_{j=1}^{m}w_j^{(p)}\frac{\sum_{i=1}^{n}s_{ij}^*\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}+\gamma\left\|S^*\right\|_F^2\\&s.t.\ S^*\mathbf{1}=\mathbf{1},\ S^*\geq 0,\ rank(\boldsymbol{L}_{S^*})=(n+m)-c\end{aligned} \tag{13}$$

where $\boldsymbol{L}_{S^*}$ is a semi-positive definite matrix with all eigenvalues not less than zero. The rank constraint is transformed to minimize the first c smallest eigenvalues of $\boldsymbol{L}_{S^*}$. Denote $\sigma_i(\boldsymbol{L}_{S^*})$ as the i-th minimum eigenvalue of the matrix $\boldsymbol{L}_{S^*}$. The objective function in (13) can be rewritten as follows:

$$\begin{aligned}&\min_{\boldsymbol{A}^{(p)},S^*}\sum_{p=1}^{v}\sum_{j=1}^{m}w_j^{(p)}\frac{\sum_{i=1}^{n}s_{ij}^*\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}+\gamma\left\|S^*\right\|_F^2\\&+\lambda\textstyle\sum_{l=1}^{c}\sigma_l(\boldsymbol{L}_{S^*})\\&s.t.\ S^*\mathbf{1}=\mathbf{1},\ S^*\geq 0\end{aligned} \tag{14}$$

According to Ky Fan's theorem [33], the eigenvalue minimization problem is equivalent to a matrix trace function as follows:

$$\sum_{l=1}^{c}\sigma_l(\boldsymbol{L}_{S^*})=\min_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}}Tr(\boldsymbol{F}^T\boldsymbol{L}_{S^*}\boldsymbol{F}), \tag{15}$$

where $\boldsymbol{F}\in\mathbb{R}^{(n+m)\times c}$ is an orthogonal matrix. Therefore, the objective function is further rewritten as follows:

$$\begin{aligned}&\min_{\boldsymbol{A}^{(p)},S^*,\boldsymbol{F}}\sum_{p=1}^{v}\sum_{j=1}^{m}w_j^{(p)}\frac{\sum_{i=1}^{n}s_{ij}^*\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}\\&+\gamma\left\|S^*\right\|_F^2+\lambda Tr(\boldsymbol{F}^T\boldsymbol{L}_{S^*}\boldsymbol{F})\\&s.t.\ S^*\mathbf{1}=\mathbf{1},\ S^*\geq 0,\ \boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}\end{aligned} \tag{16}$$

3.2 Optimization algorithm

In (16), the objective function contains four types of unknown variables including $\boldsymbol{F}$, $\boldsymbol{S}^*$, $\{\boldsymbol{w}^{(p)}\}_{p=1}^v$, and $\{\boldsymbol{A}^{(p)}\}_{p=1}^v$. Since the objective function is a non-convex optimization problem, we adopt an alternating iterative algorithm to solve it. Note that the optimization processes of variables $\boldsymbol{F}$ and $\boldsymbol{S}^*$ belong to one subprocedure. They are iteratively optimized until the Laplacian rank constraint condition is satisfied, i.e., $\sum_{l=1}^{c}\sigma_l(\boldsymbol{L}_{\boldsymbol{S}^*})=0$ and $\sum_{l=1}^{c+1}\sigma_l(\boldsymbol{L}_{\boldsymbol{S}^*})>0$.

1) Fix $\boldsymbol{S}^*$, $\{\boldsymbol{w}^{(p)}\}_{p=1}^v$, $\{\boldsymbol{A}^{(p)}\}_{p=1}^v$, and update $\boldsymbol{F}$. When all other variables except $\boldsymbol{F}$ are fixed, the optimization problem in (16) is transformed into:

$$\min_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}} Tr(\boldsymbol{F}^T\boldsymbol{L}_{\boldsymbol{S}^*}\boldsymbol{F}) \tag{17}$$

As mentioned in Section 3.1, the normalized Laplacian matrix $\boldsymbol{L}_{\boldsymbol{S}^*}=\boldsymbol{I}-\boldsymbol{D}^{-1/2}\boldsymbol{P}\boldsymbol{D}^{-1/2}$. By substituting $\boldsymbol{L}_{\boldsymbol{S}^*}$, (17) is rewritten as:

$$\max_{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}} Tr(\boldsymbol{F}^T\boldsymbol{D}^{-\frac{1}{2}}\begin{bmatrix}\boldsymbol{S}^*\\ \boldsymbol{S}^{*T}\end{bmatrix}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{F}). \tag{18}$$

where the block matrices $\boldsymbol{F}\in\mathbb{R}^{(n+m)\times c}$ and $\boldsymbol{D}\in\mathbb{R}^{(n+m)\times(n+m)}$ are decomposed into the following form:

$$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{V}\end{bmatrix},\ \boldsymbol{D}=\begin{bmatrix}\boldsymbol{D}_U & \\ & \boldsymbol{D}_V\end{bmatrix}, \tag{19}$$

Then, the (18) can be further rewritten as:

$$\max_{\boldsymbol{U}^T\boldsymbol{U}+\boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{I}} Tr(\boldsymbol{U}^T\boldsymbol{D}_{\boldsymbol{U}}^{-\frac{1}{2}}\boldsymbol{S}^*\boldsymbol{D}_{\boldsymbol{V}}^{-\frac{1}{2}}\boldsymbol{V}), \tag{20}$$

According to Theorem 2, the above problem has a closed-form solution.

Theorem 2 [31] *For the following problem:*

$$\max_{\boldsymbol{X}^T\boldsymbol{X}+\boldsymbol{Y}^T\boldsymbol{Y}=\boldsymbol{I}} Tr(\boldsymbol{X}^T\boldsymbol{Q}\boldsymbol{Y}), \tag{21}$$

where $\boldsymbol{Q}\in\mathbb{R}^{n\times m}$, $\boldsymbol{X}\in\mathbb{R}^{n\times c}$, and $\boldsymbol{Y}\in\mathbb{R}^{m\times c}$. A closed-form solution is $\boldsymbol{X}=\frac{\sqrt{2}}{2}\boldsymbol{U}_1$ and $\boldsymbol{Y}=\frac{\sqrt{2}}{2}\boldsymbol{V}_1$. $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ are the first c left and right singular vectors of matrix $\boldsymbol{Q}$, respectively.

2) Fix $\boldsymbol{F}$, $\{\boldsymbol{w}^{(p)}\}_{p=1}^v$, $\{\boldsymbol{A}^{(p)}\}_{p=1}^v$, and update $\boldsymbol{S}^*$. When all other variables are fixed except $\boldsymbol{S}^*$, the objective function in (16) becomes:

$$\begin{aligned}\min_{\boldsymbol{A}^{(p)},\boldsymbol{S}^*,\boldsymbol{F}} &\sum_{i=1}^{n}\sum_{j=1}^{m}s_{ij}^*\sum_{p=1}^{v}w_j^{(p)}\frac{\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}\\ &+\gamma\left\|\boldsymbol{S}^*\right\|_F^2+\lambda Tr(\boldsymbol{F}^T\boldsymbol{L}_{\boldsymbol{S}^*}\boldsymbol{F})\\ s.t.\ &\boldsymbol{S}^*\boldsymbol{1}=\boldsymbol{1},\ \boldsymbol{S}^*\geq 0\end{aligned} \tag{22}$$

Let $\boldsymbol{f}_i$ denote the i-th row of $\boldsymbol{F}$ and d_i denote the i-th diagonal element of $\boldsymbol{D}$. The matrix trace in the objective function can be expanded as follows:

$$Tr(\boldsymbol{F}^T\boldsymbol{L}_{\boldsymbol{S}^*}\boldsymbol{F})=\sum_{i=1}^{n}\sum_{j=1}^{m}s_{ij}^*\|\frac{\boldsymbol{f}_i}{\sqrt{d_i}}-\frac{\boldsymbol{f}_{(n+j)}}{\sqrt{d_{(n+j)}}}\|_2^2 \tag{23}$$

For simplicity, let $m_j^{(p)}=\sum_{t=1,t\neq j}^{m}\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2$ and $h_{ij}=\|\frac{\boldsymbol{f}_i}{\sqrt{d_i}}-\frac{\boldsymbol{f}_{(n+j)}}{\sqrt{d_{(n+j)}}}\|_2^2$. Since (16) is independent among different i, each row vector of $\boldsymbol{S}^*$ can be solved as follows:

$$\begin{aligned}\min_{\boldsymbol{s}_i^*}&\sum_{j=1}^{m}\left(s_{ij}^*\sum_{p=1}^{v}\frac{w_j^{(p)}}{m_j^{(p)}}\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2+\gamma(s_{ij}^*)^2+\lambda s_{ij}^*h_{ij}\right)\\ s.t.\ &\boldsymbol{s}_i^*\geq 0,\ \boldsymbol{s}_i^*\boldsymbol{1}=\boldsymbol{1}\end{aligned} \tag{24}$$

Let $\widetilde{h}_{ij}=\sum_{p=1}^{v}\frac{w_j^{(p)}}{m_j^{(p)}}\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2+\lambda h_{ij}$; this problem can be written in vector form as in (25). Similar to [34], this problem can be solved by a simple and efficient iterative algorithm, which has a closed-form solution.

$$\min_{\boldsymbol{s}_i^*\geq 0,\boldsymbol{s}_i^*\boldsymbol{1}=\boldsymbol{1}}\left\|\boldsymbol{s}_i^*+\frac{\widetilde{\boldsymbol{h}}_i}{2\gamma}\right\|_2^2 \tag{25}$$

3) Fix $\boldsymbol{F}$, $\boldsymbol{S}^*$, $\{\boldsymbol{A}^{(p)}\}_{p=1}^v$, and update $\{\boldsymbol{w}^{(p)}\}_{p=1}^v$. According to (11), the contribution weight for each sub-cluster in each view is obtained automatically without introducing any additional parameter. This weight reveals the clustering capacity of each sub-cluster.

4) Fix $\boldsymbol{F}$, $\boldsymbol{S}^*$, $\{\boldsymbol{w}^{(p)}\}_{p=1}^v$, and update $\{\boldsymbol{A}^{(p)}\}_{p=1}^v$. When all other variables are fixed, the prototype of each sub-cluster in each view can be updated by solving the following problem:

$$\min_{\boldsymbol{a}_j^{(p)}}\sum_{i=1}^{n}s_{ij}^*\frac{w_j^{(p)}}{m_j^{(p)}}\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2 \tag{26}$$

By taking the derivative of $a_j^{(p)}$ and setting it to zero, the j-th prototype in the p-th view can be updated as:

$$\boldsymbol{a}_j^{(p)}=\frac{\sum_{i=1}^{n}s_{ij}^*\boldsymbol{x}_i^{(p)}}{\sum_{i=1}^{n}s_{ij}^*} \tag{27}$$

When the prototypes in the p-th view are updated, the auxiliary variable $m_j^{(p)}$ should be updated according to:

$$m_j^{(p)}=\sum_{t=1,t\neq j}^{m}\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2 \tag{28}$$

Algorithm 1 Fine-grained multi-view clustering with robust multi-prototypes representation (FgMVC).

Input: Multi-view dataset $\{\boldsymbol{X}^{(p)}\}_{p=1}^{v}$, number of clusters c, number of sub-clusters m, tradeoff parameters γ and λ.
Output: The clustering assignment.
1: Initialize m prototypes for each view randomly;
2: Initialize weights of sub-clusters in each view to $1/v$;
repeat
3: Initialize $\boldsymbol{S}^*$ by solving (32);
while the rank constraint (i.e., $\sum_{l=1}^{c}\sigma_l(\boldsymbol{L}_{\boldsymbol{S}^*}) = 0$ and $\sum_{l=1}^{c+1}\sigma_l(\boldsymbol{L}_{\boldsymbol{S}^*}) > 0$) is not satisfied **do**
4: Fix $\boldsymbol{S}^*$, $\{\boldsymbol{w}^{(p)}\}_{p=1}^{v}$, $\{\boldsymbol{A}^{(p)}\}_{p=1}^{v}$, and update $\boldsymbol{F}$ by solving (20);
5: Fix $\boldsymbol{F}$, $\{\boldsymbol{w}^{(p)}\}_{p=1}^{v}$, $\{\boldsymbol{A}^{(p)}\}_{p=1}^{v}$, and update $\boldsymbol{S}^*$ by solving (25);
end while
6: Fix $\boldsymbol{F}$, $\boldsymbol{S}^*$, $\{\boldsymbol{A}^{(p)}\}_{p=1}^{v}$, $m_j^{(p)}$, and update $w_j^{(p)}$ by (11);
7: Fix $\boldsymbol{F}$, $\boldsymbol{S}^*$, $w_j^{(p)}$, $m_j^{(p)}$, and update $\{\boldsymbol{A}^{(p)}\}_{p=1}^{v}$ by (27);
8: Fix $\boldsymbol{F}$, $\boldsymbol{S}^*$, $\{\boldsymbol{A}^{(p)}\}_{p=1}^{v}$, $w_j^{(p)}$, and update $m_j^{(p)}$ by (28);
until convergence
Based on connectivity of consensus graph matrix $\boldsymbol{S}^*$, the clustering assignment is obtainded directly.

These variables are optimized by alternate iteration until convergence, and the local optimal solution to the objective function is obtained. Based on the connectivity of graph matrix $\boldsymbol{S}^*$, we can obtain the clustering label of each sample directly. To make these update rules clear, the whole optimization procedure is summarized in Algorithm 1.

4 Theoretical analysis

4.1 Initialization analysis

The initialization of $\boldsymbol{S}^*$ has an important effect on the optimization algorithm. Elements of matrix $\boldsymbol{S}^*$ represent the probability of connection between the samples and the prototypes. Note that the connection probability is related to the distance between samples and prototypes. The farther the sample is from the prototype of the sub-cluster, the smaller the probability of connection between them.

Denote a vector $\boldsymbol{d}_i \in \mathbb{R}^{n\times 1}$ with the j-th element as $d_{ij} = \sum_{p=1}^{v} s_{ij}^* \frac{w_j^{(p)}}{m_j^{(p)}} \|\boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)}\|_2^2$. Since each row vector $\boldsymbol{s}_i^*$ of $\boldsymbol{S}^*$ can be solved independently, the initialization problem of $\boldsymbol{S}^*$ can be decomposed into n sub-problems as follows:

$$\min_{\boldsymbol{s}_i^*} \sum_{j=1}^{m}\left(s_{ij}^* \sum_{p=1}^{v}\frac{w_j^{(p)}}{m_j^{(p)}}\|\boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)}\|_2^2 + \gamma (s_{ij}^*)^2\right) \quad s.t.\ \boldsymbol{s}_i^* \geq 0,\ \boldsymbol{s}_i^*\mathbf{1} = \mathbf{1} \tag{29}$$

By simple mathematical derivation, the above equation can be rewritten as:

$$\min_{\boldsymbol{s}_i^*\geq 0, \boldsymbol{s}_i^{*T}\mathbf{1}=1} \left\|\boldsymbol{s}_i^* + \frac{\boldsymbol{d}_i}{2\gamma}\right\|_2^2 \tag{30}$$

The Lagrangian function of the above problem is

$$L(\boldsymbol{s}_i^*, \beta, \theta_i) = \frac{1}{2}\left\|\boldsymbol{s}_i^* + \frac{\boldsymbol{d}_i}{2\gamma_i}\right\| - \beta(\boldsymbol{s}_i^{*T}\mathbf{1} - 1) - \boldsymbol{\theta}_i^T\boldsymbol{s}_i^* \tag{31}$$

where $\beta \geq 0$ and $\boldsymbol{\theta}_i \geq 0$ are Lagrange multipliers. Following the KKT condition, we can obtain the optimal solution as follows:

$$s_{ij}^* = (-\frac{d_{ij}^x}{2\gamma} + \beta)_+ \tag{32}$$

where the regularization parameter γ is difficult to determine. We adopt an effective method in [35] to determine the value of γ.

4.2 Computational complexity

In Algorithm 1, the main time cost consists of five parts: The first part is the initialization of $\boldsymbol{S}^*$ (Step 3). The second part is to update $\boldsymbol{F}$ and $\boldsymbol{S}^*$ alternately (Steps 4-5). The third part updates the weight of each sub-cluster in each view (Step 6). The fourth part updates the prototype of each sub-cluster in each view (Step 7). The fifth part updates the auxiliary variables (Step 8).

The time complexity of the first part is $O(nmd)$, where $d = \sum_{p=1}^{v} d^{(p)}$ is the sum of the feature dimensions, n is the number of samples, and m is the number of sub-clusters. The time complexity of the second part is $O((m^2n + nmc)t_1)$, where c is the number of clusters and t_1 is the iteratio number of the sub-procedure. In the third part, the time complexity required to update the weight of each sub-cluster in each view is $O(nmdv)$, where v is the number of views. The time complexity required to update the prototypes is $O(mv)$. Finally, the time complexity required to update the auxiliary variables is $O(m^2v)$. Therefore, the total time complexity required by Algorithm 1 is $O(((m^2n+nmc)t_1 + (nd + m)mv)t)$, where t is the total number of algorithm iterations.

To improve the efficiency of the algorithm, we learned a sparse $\boldsymbol{S}^*$, which only needs to update the connection probability between each sample and k nearest prototypes. When updating other variables through sparse $\boldsymbol{S}^*$, the time complexity can also be significantly reduced.

4.3 Convergence analysis

According to the following lemma [36], we prove that the proposed algorithm converges to a local optimal solution. When other variables are fixed except $\boldsymbol{S}^*$, the updated variable $\widetilde{\boldsymbol{S}}^*$ will decrease the objective function value in (16) in each iteration.

Lemma 1 *For any positive real number a and b, the following inequality holds:*

$$\sqrt{a}-\frac{a}{2\sqrt{b}} \leq \sqrt{b}-\frac{b}{2\sqrt{b}} \tag{33}$$

For simplicity, let $\widetilde{\boldsymbol{Q}}_j^{(p)}=\frac{\sum_{i=1}^{n}\tilde{s}_{ij}^{*}\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}$, and let $\boldsymbol{Q}_j^{(p)}=\frac{\sum_{i=1}^{n}s_{ij}^{*}\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}{\sum_{t=1,t\neq j}^{m}\left\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2}$. After the sub-procedure of updating $\boldsymbol{S}^*$, we can obtain the following inequality:

$$\sum_{p=1}^{v}\sum_{j=1}^{m}\frac{\widetilde{\boldsymbol{Q}}_j^{(v)}}{\sqrt{\boldsymbol{Q}_j^{(v)}}}+\gamma\|\widetilde{\boldsymbol{S}}^*\|_F^2 \leq \sum_{p=1}^{v}\sum_{j=1}^{m}\frac{\boldsymbol{Q}_j^{(v)}}{\sqrt{\boldsymbol{Q}_j^{(v)}}}+\gamma\left\|\boldsymbol{S}^*\right\|_F^2 \tag{34}$$

Then, according to Lemma 1, the following inequality can be obtained:

$$\begin{aligned}&\sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\widetilde{\boldsymbol{Q}}_j^{(p)}}-\sum_{p=1}^{v}\sum_{j=1}^{m}\frac{\widetilde{\boldsymbol{Q}}_j^{(p)}}{2\sqrt{\boldsymbol{Q}_j^{(p)}}}\\ &\leq \sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\boldsymbol{Q}_j^{(p)}}-\sum_{p=1}^{v}\sum_{j=1}^{m}\frac{\boldsymbol{Q}_j^{(p)}}{2\sqrt{\boldsymbol{Q}_j^{(p)}}}\end{aligned} \tag{35}$$

Combining (34) and (35), the following inequality can be easily obtained:

$$\sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\widetilde{\boldsymbol{Q}}_j^{(p)}}+\gamma\|\widetilde{\boldsymbol{S}}^*\|_F^2 \leq \sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\boldsymbol{Q}_j^{(p)}}+\gamma\left\|\boldsymbol{S}^*\right\|_F^2 \tag{36}$$

The above inequality is expanded as follows:

$$\begin{aligned}&\sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\frac{\sum_{i=1}^{n}\tilde{s}_{ij}^{*}\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2}{\sum_{t=1,t\neq j}^{m}\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2}}+\gamma\|\widetilde{\boldsymbol{S}}^*\|_F^2\\ &\leq \sum_{p=1}^{v}\sum_{j=1}^{m}\sqrt{\frac{\sum_{i=1}^{n}s_{ij}^{*}\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2}{\sum_{t=1,t\neq j}^{m}\|\boldsymbol{a}_t^{(p)}-\boldsymbol{a}_j^{(p)}\|_2^2}}+\gamma\|\boldsymbol{S}^*\|_F^2\end{aligned} \tag{37}$$

Therefore, the algorithm proposed in this paper can converge to a locally optimal solution.

4.4 Connection to other multi-view clustering

Connection to coarse-grained multi-view clustering Coarse-grained multi-view clustering methods take the view as the fusion object, which assigns a weight $w^{(p)}$ to each view to distinguish the contribution of different views. Then the local structures in each view will have the same weight. Similarly, the FgMVC method assigns a weight $w_j^{(p)}$ to each sub-cluster to distinguish the contribution of each sub-cluster. When sub-clusters within all views have the same weight, it means that the FgMVC will treat each view equally. When sub-clusters in the p-th views have the same weight, then the weight of each sub-cluster in the p-th view will satisfy $w_1^{(p)}=w_2^{(p)}=...=w_m^{(p)}=w^{(p)}$, which means that the overall structure of each view is given a weight. At the moment, the fusion object changes from the sub-cluster to the entire view.

Connection to multi-view K-means Without considering the overlapping problem of prototypes, when $\gamma \to \infty$, problem (38) is closely related to multi-view K-means, where the constraint $S^* \in \Omega$ represents that the consensus bipartite graph has c connected components.

$$\min_{\substack{\boldsymbol{A}^{(p)},\boldsymbol{S}^*\geq 0,\\ \boldsymbol{S}^*\boldsymbol{1}=\boldsymbol{1},\boldsymbol{S}^*\in\Omega}} \sum_{p=1}^{v}\sum_{i=1}^{n}\sum_{j=1}^{m}s_{ij}^{*}w_j^{(p)}\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2+\gamma\left\|\boldsymbol{S}^*\right\|_F^2 \tag{38}$$

Proof When $\boldsymbol{S}^*$ satisfies the rank constraint, $\lambda Tr(F^T L_{S^*} F)$ becomes zero. At this point, the consensus bipartite graph has c connected components, and the connection probability matrix of the l-th connected component can be expressed as $S_l^* \in R^{n_l \times m_l}$, where n_l and m_l represent the number of samples and the number of prototypes of the l-th connected component respectively. When $\gamma \to \infty$, in order to ensure the minimum value of problem (38), the following problems need to be minimized:

$$\min_{\boldsymbol{S}_l^*\geq 0,\boldsymbol{S}_l^*\boldsymbol{1}=\boldsymbol{1}}\left\|\boldsymbol{S}_l^*\right\|_F^2 \tag{39}$$

The optimal solution to problem (39) is that all elements of $\boldsymbol{S}_l^*$ are $1/m_l$. Therefore, when $\gamma \to \infty$, only samples and prototypes within the same cluster can be connected.

When the number of samples and the number of prototypes in each connected component are fixed, problem (40) becomes

$$\min_{\boldsymbol{Z}^*}\sum_{p=1}^{v}\sum_{i=1}^{n}\frac{1}{\sum_{t=1}^{m}z_{it}^*}\sum_{j=1}^{m}w_j^{(p)}z_{ij}^*\left\|\boldsymbol{x}_i^{(p)}-\boldsymbol{a}_j^{(p)}\right\|_2^2 \tag{40}$$

where $\boldsymbol{Z}^*$ represents the partition of samples and prototypes. When the i-th sample is connected to the j-th prototype, $z_{ij}^*=1$; otherwise, $z_{ij}^*=0$. The equation $\sum_{t=1}^{m}z_{it}^*$ is used to calculate the number of prototypes in the l-th connected

component. Then the optimal solution to the prototype in question (41) can be obtained:

$$\boldsymbol{a}_j^{(p)} = \frac{\sum_{i=1}^{n} z_{ij}^* \boldsymbol{x}_i^{(p)}}{\sum_{i=1}^{n} z_{ij}^*} = \frac{1}{n_l} \sum_{i=1}^{n} z_{ij}^* \boldsymbol{x}_i^{(p)} \tag{41}$$

In this case, the all prototypes in the l-th connected component are the same center point $\tilde{\boldsymbol{a}}_l^{(p)}$, and $\tilde{\boldsymbol{a}}_l^{(p)}$ is the mean of all samples in the l-th connected component, so $\tilde{\boldsymbol{a}}_l^{(p)}$ is equivalent to the clustering center of the l-th connected component. Therefore, for c connected components, there are c clustering centers. All sub-clusters in the same connected component have the same weight; then, problem (42) can be written as

$$\min_{\boldsymbol{Y} \in \mathbb{R}^{n \times c}, \tilde{\boldsymbol{A}}^{(p)} \in \mathbb{R}^{c \times d^{(p)}}} \sum_{p=1}^{v} \sum_{i=1}^{n} \sum_{l=1}^{c} w_l^{(p)} y_{il} \left\| \boldsymbol{x}_i^{(p)} - \tilde{\boldsymbol{a}}_l^{(p)} \right\|_2^2 \tag{42}$$

where $\boldsymbol{Y}$ is the membership matrix. When the i-th sample belongs to the l-th cluster, $y_{il} = 1$; otherwise, $y_{il} = 0$. $w_l^{(p)}$ represents the weight of the l-th cluster in the p-th view. At this point, problem (42) is the extension of K-means to multi-view clustering. Therefor, when $\gamma \to \infty$, to minimize the problem (38), it can be obtained that the prototypes in each connected component are the same, which is equivalent to only one prototype for each connected component. Therefore, FgMVC is closely related to multi-view K-means. □

5 Experiments

In this section, we conduct experiments on two synthetic and six real-world datasets. The synthetic multi-view datasets (Toy_1 and Toy_2) are generated in 3-dimensional space and are collected from their front view, top view, and side view. The real-world datasets include Handwritten,[1] Wiki,[2] 100Leaves,[3] COIL20,[4] ORL,[5] and Prokaryotic Phyla.[6] The information of datasets is shown in Table 2. Three widely used cluster evaluation metrics are utilized, including Accuracy (ACC), Normalized Mutual Information (NMI), and Purity. All experiments are conducted in the MATLAB R2020a environment with an Inter Core i5-9400 CPU (2.90GHz and 8GB RAM).

5.1 Baselines

To verify the effectiveness of FgMVC, we compare both the single-view and multi-view clustering methods. MVASM and AKM^{3}C are closely related to FgMVC. They all belong to the extension of K-means. The other multi-view clustering methods are the most advanced at present. CorSC, AACC, and MVSC are multi-view methods based on spectral clustering. AMGL is a graph-based multi-view clustering method.

1) **K-means:** [37] Based on Euclidean distance, K-means assigns each sample to the nearest prototype. Then, the prototypes and samples partition are updated iteratively.
2) **Spectral Clustering (SC):** [38] This method conducts separately on each view, where the best results are recorded.
3) **K-multiple-means (KMM):** [31] KMM is an extension of classic K-means. Compared with the K-means, KMM has a better ability to capture non-spherical clusters.
4) **Co-regularized multi-view spectral clustering (CorSC):** [39] To make the corresponding samples in each view have the same cluster membership, it learns a common spectral embedding by co-regularizing different spectral embeddings of different views.
5) **Affinity Aggregation Spectral Clustering (AASC):** [40] The spectral clustering algorithm is extended to sets with multiple affinities, in order to be more immune to invalid affinities and irrelevant features by seeking the best combination of affinity matrices.
6) **Multi-view Spectral Clustering (MVSC):** [41] The local manifold fusion is used to integrate heterogeneous features and the bipartite graph is used to approximate the similarity graph to improve efficiency.
7) **Auto-weighted Multiple Graph Learning (AMGL):** [42] Weights of different views are adaptively obtained,

Table 2 Statistics of the datasets

Datasets	Samples	Views	Clusters	Attribute
Toy_1	3000	3	3	–
Toy_2	5000	3	5	–
Handwritten	2000	6	10	digit image
Wiki	2866	2	10	text+image
100Leaves	1600	3	100	plant image
COIL20	1440	3	20	object image
ORL	400	4	40	face image
Prokaryotic Phyla	551	3	4	text

[1] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[2] https://github.com/jmpu/multiview_cluster/blob/master/Wiki_textimage/Wiki_textimage.mat

[3] https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set

[4] http://www.cs.columbia.edu/CAVE/research/coil-20.htm

[5] http://www.uk.research.att.com/facedatabase.html

[6] https://github.com/mbrbic/Multi-view-LRSSC/blob/master/datasets/prokaryotic.mat

and different Laplacian matrices are learned from different views to calculate the embedding matrix in the subspace.

8) **Multi-View clustering with Adaptive Sparse Memberships and Weight Allocation (MVASM):** [24] A common membership matrix with appropriate sparsity is constructed on different views, and the centroid matrix of each view and its corresponding weights are learned.
9) **Adaptive K-Multiple-Means for Multi-view Clustering (AKM^{3}C):** [25] To capture the heterogeneous features of each view, complex data structures are modeled by utilizing multiple sub-cluster centers. The weight for each view is learned automatically by combining the weight strategy. Finally, an optimal shared bipartite graph is constructed by fusing complementary information from different views.

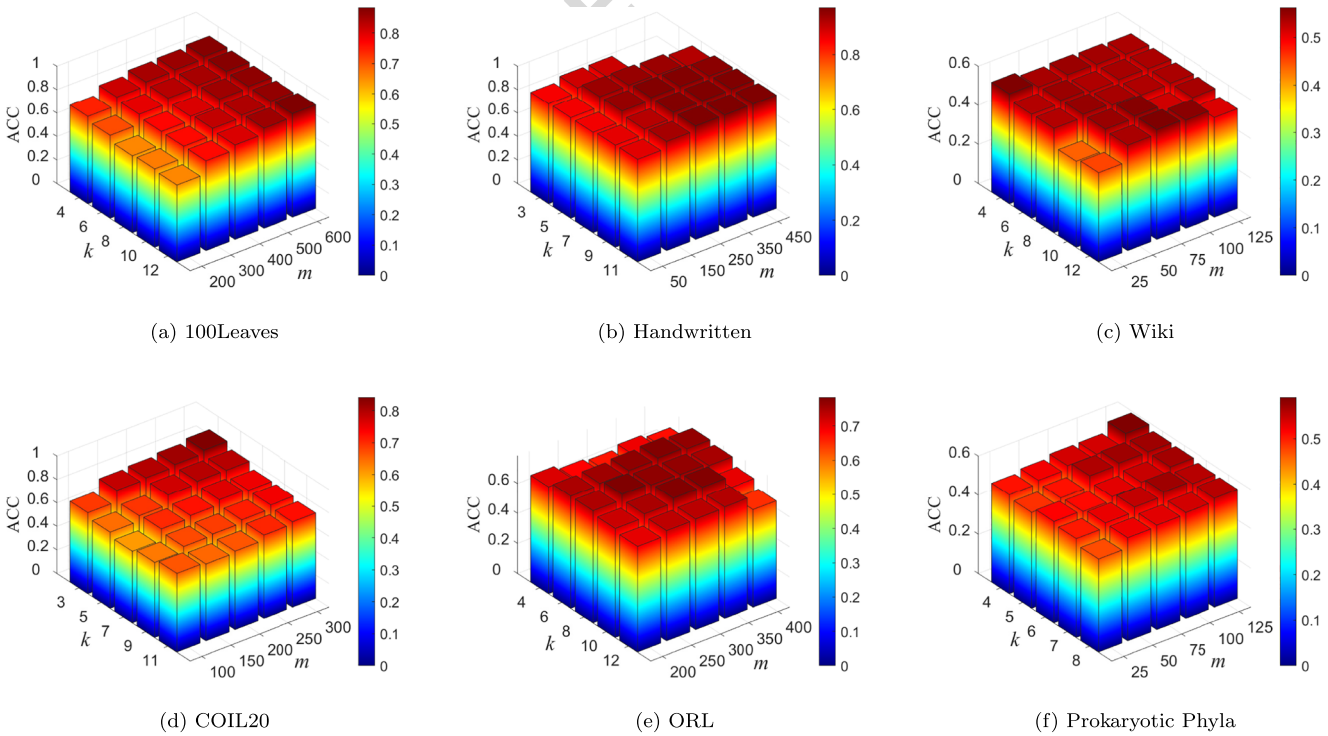
5.2 Parameter selection

In this paper, FgMVC has four important parameters, including m, k, γ and λ. Parameter m is the number of sub-clusters. To search the optimal m for each dataset, its value is set to $m_0 c$, where m_0 is a predefined constant and c is the number of clusters. In Fig. 2, the value of m has a great influence on the clustering results. When m value is set too small, non-convex structures in the original data cannot be fitted adequately. When m value is set too large, the computational complexity between the samples and prototypes will increase.

Specifically, a larger m increases the dimension of the matrix $\boldsymbol{S}^*$, which participates in the updating of variables $\{\boldsymbol{w}^{(p)}\}_{p=1}^{v}$, $\{\boldsymbol{A}^{(p)}\}_{p=1}^{v}$, and $\boldsymbol{F}$. A sparse connection probability matrix $\boldsymbol{S}^*$ is obtained to reduce the computational complexity, which only needs to update the connection probability between each sample and k nearest prototypes. Therefore, an appropriate k should reduce the amount of computation related to $\boldsymbol{S}^*$.

Parameter k is the number of neighbor prototypes. For simplicity, k is usually set to a fixed constant. In this paper, the value of k is searched within a fixed interval $[3, 15]$. In Fig. 2, on the Handwritten, Wiki, COIL20, ORL, and Prokaryotic Phyla datasets, the optimal values of k are 4, 7, 4, 3, 8, 4, respectively. Indeed, the value of k determines the sparsity of the connected probability matrix $\boldsymbol{S}^*$. When k is set too small, many important connection relations are eliminated. Many redundant connected components would occur in $\boldsymbol{S}^*$. Since the connection probability between k prototypes and each sample should be calculated, an excessively large k will lead to high computational complexity.

The parameter γ controls the sparsity of the bipartite graph, and its setting is adaptive. For parameter λ, $\lambda = \gamma$ is initially set according to [15]. If the connected component of



(a) 100Leaves (b) Handwritten (c) Wiki

(d) COIL20 (e) ORL (f) Prokaryotic Phyla

Fig. 2 The experimental results of FgMVC on six real-world datasets with different combinations of the number of neighbor prototypes k and the number of sub-clusters m

the bipartite graph is greater than the number of clusters c, we divide λ by 2. If the connected component of the bipartite graph is less than c, we multiply λ by 2.

5.3 Experiments on toy datasets

Two synthetic multi-view datasets (Toy_1 and Toy_2) are generated in three-dimensional space. Each dataset has three different views: front view, top view, and side view. Tables 3 and 4 show the construction details of Toy_1 and Toy_2 datasets. Toy_1 and Toy_2 are generated by using the dot product $\otimes$ of $\sin(t)$ and $\cos(t)$ functions, where t ranges from 0 to π in steps of $\pi/999$. Toy_1 consists of 5 clusters, each of which generates 1000 samples. Toy_2 consists of 3 clusters, each of which generates 1000 samples. As shown in Figs. 3a and 4a, Toy_1 dataset is convex and Toy_2 dataset is non-convex.

In Fig. 3b, there are intersections between different clusters. In Fig. 3c and d, there are covers between clusters. In Fig. 3e, single-view K-means cannot achieve satisfactory clustering. As a multi-view expanded version of K-means, MVASM achieves better clustering which is shown in Fig. 3f. In Figs. 3g and 4g, compared with MVASM, the clustering results of AKM^{3}C are obviously improved, especially on the non-convex dataset. This phenomenon proves that the effectiveness of multi-prototypes representation in dealing with non-convex structures. In Figs. 3h and 4h, compared with AKM^{3}C, our FgMVC both achieves impressive clustering results on convex and non-convex datasets by using robust multi-prototype representations.

5.4 Experiments on real-world datasets

In CorSC, the co-regularization parameter λ is varied from 0.01 to 0.05 with a step size of 0.01. In AMGL, there is no free parameter. In AASC, the number of neighbors is set as 10. In MVSC, the parameter r is searched in logarithmic $log_{10}r$ from 0.1 to 2 with a step size of 0.2. In MVASM, the parameter q is varied from 1 to 6 with a step size of 0.5, and parameter γ searches for the optimal value in $\{0.001, 0.10, 10, 100\}$. In AKM^{3}C, the number of neighbor prototypes k is set as 5 according to [25], and the number of sub-clusters m is searched within the range $[\sqrt{nc}, 2\sqrt{nc}]$.

Compared with other methods, the clustering results in terms of ACC, NMI and purity are shown in Tables 5, 6 and 7 respectively, where the mean and standard deviation are reported over 30 runs. The best clustering results are identified in bold font. For single-view K-means, KMM, and SC, we conduct separately on each view, where the best results are recorded. According to the experimental results in Tables 5–7, we can draw the following conclusions:

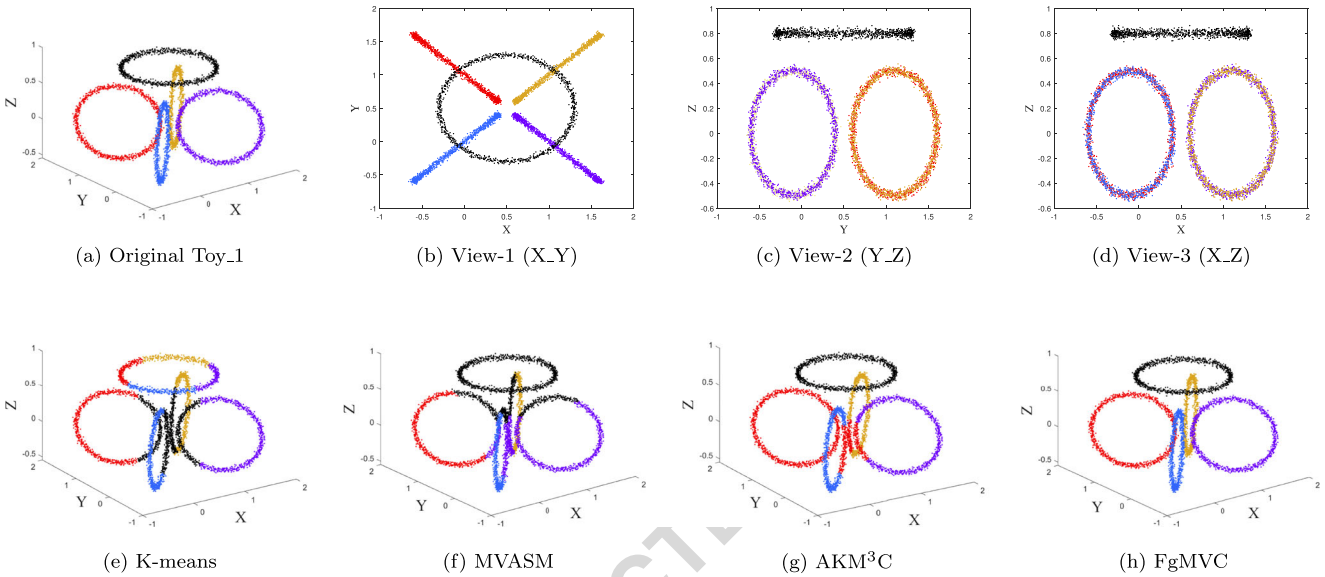
1) AASC, MVSC, and CorSC are based on spectral clustering methods (SC-based). MVASM, AKM^{3}C, and FgMVC are K-means-based clustering methods. It can be found that the clustering results of SC-based methods are better than MVASM on most datasets. However, AKM^{3}C achieves the same clustering effect as SC-based methods. AKM^{3}C captures non-spherical clusters in data by utilizing multi-prototypes representation. Compared with AKM^{3}C, our FgMVC can obtain better clustering results. Our method automatically allocates contribution weights based on the clustering capacity of sub-clusters to achieve fine-grained information fusion. To solve the problem of overlapping sub-clusters caused by using multi-prototypes representation, we increase the distance between prototypes to obtain robust multi-prototypes representation.
2) Compared with average weighted multi-view clustering methods (e.g., CorSC), auto-weighted multi-view clustering can achieve better clustering results on most datasets. For example, on the 100Leaves dataset, compared with the CorSC, MVSC improves by 4.97%, 2.54%, and 4.00% in ACC, NMI, and Purity, respectively. In the auto-weighted multi-view clustering methods, the clustering performance of our FgMVC is better.
3) Our FgMVC does not outperform all the traditional clustering methods on the ORL and Prokaryotic Phyla datasets. On the ORL dataset, our method is slightly weaker than MVSC on NMI score. However, the NMI score of our method approaches the highest score. On the Prokaryotic Phyla dataset, our method is weaker than AASC on NMI score. However, On ACC and Purity scores, our method achieves better results.

Table 3 The construction method of synthetic dataset Toy_1

Category of the samples	Three dimensions of the samples
Red cluster	$(\cos(t) \otimes \cos(t) - 0.6,\ \sin(t) \otimes \sin(t) + 0.6,\ \sin(t) \otimes \cos(t))$
Blue cluster	$(\sin(t) \otimes \cos(t) - 0.1,\ \sin(t) \otimes \cos(t) - 0.1,\ \cos(t) \otimes \cos(t) - 0.5)$
Purple cluster	$(\cos(t) \otimes \cos(t)) + 0.6,\ \sin(t) \otimes \sin(t) - 0.6,\ \sin(t) \otimes \cos(t))$
Yellow cluster	$(\sin(t) \otimes \cos(t)) + 1.1,\ \sin(t) \otimes \cos(t) - 1.1,\ \cos(t) \otimes \cos(t) - 0.5)$
Black cluster	$(1.6 \times \cos(t) \otimes \cos(t)) - 0.3,\ 1.6 \times \sin(t) \otimes \cos(t) + 0.5,\ 0.8)$

Table 4 The construction method of synthetic dataset Toy_2

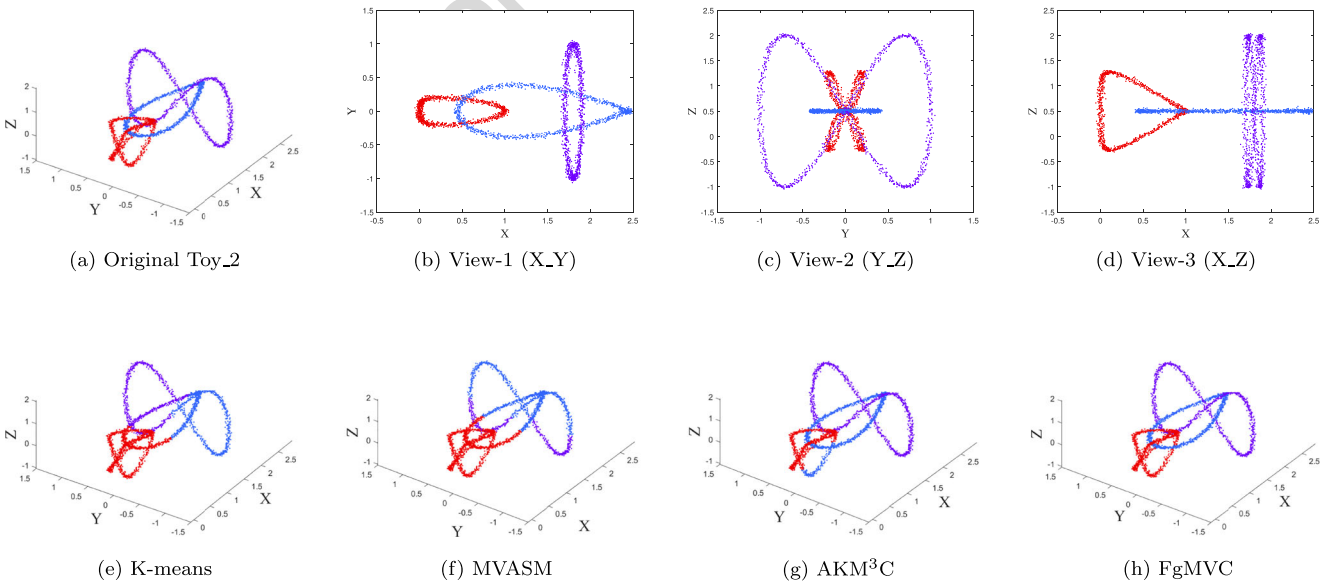
Category of the samples	Three dimensions of the samples
Red cluster	$(\sin(t) \otimes \sin(t) \otimes \sin(t) \otimes \sin(t) + 0.2,\ 0.4 \times \sin(t) \otimes \cos(t),\ 2 \times in(t) \otimes \cos(t) \otimes \cos(t) + 0.5)$
Blue cluster	$(2 \times \sin(t) \otimes \sin(t) + 0.65,\ \sin(t) \otimes \cos(t) \otimes \cos(t),\ 0.01 \times \sin(t) \otimes \sin(t) \otimes \cos(t) + 0.5)$
Purple cluster	$(0.1 \times \cos(t) + 2,\ \sin(t),\ 3 \times \sin(t) \otimes \cos(t) + 0.5)$



(a) Original Toy_1 (b) View-1 (X_Y) (c) View-2 (Y_Z) (d) View-3 (X_Z)

(e) K-means (f) MVASM (g) AKM^{3}C (h) FgMVC

Fig. 3 The clustering results on Toy_1 dataset. Sample points of the same color belong to the same cluster



(a) Original Toy_2 (b) View-1 (X_Y) (c) View-2 (Y_Z) (d) View-3 (X_Z)

(e) K-means (f) MVASM (g) AKM^{3}C (h) FgMVC

Fig. 4 The clustering results on Toy_2 dataset. Sample points of the same color belong to the same cluster

Table 5 Clustering performance comparison in terms of ACC on six real-world datasets

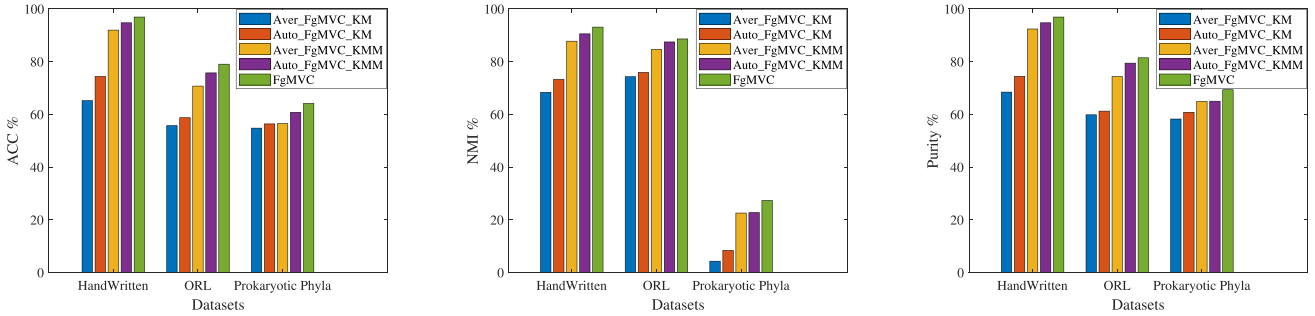
ACC(%)	Handwritten	Wiki	100Leaves	COIL20	ORL	Prokaryotic Phyla
K-means(BestView)	68.80(±6.25)	54.07(±2.04)	54.76(±1.73)	55.61(±4.84)	66.88(±3.90)	57.17(±9.90)
SC(BestView)	69.21(±5.28)	53.99(±2.29)	54.80(±1.96)	70.83(±3.43)	70.97(±2.29)	45.14(±0.96)
KMM(BestView)	77.74(±6.07)	55.86(±2.04)	51.65(±1.98)	66.61(±4.17)	65.21(±3.77)	53.40(±1.87)
AASC	82.35(±0.00)	49.27(±0.50)	87.28(±1.44)	78.14(±1.67)	76.59(±2.48)	63.81(±0.32)
AMGL	82.81(±8.16)	48.57(±3.90)	76.21(±2.96)	81.24(±3.71)	71.68(±3.43)	54.45(±2.74)
MVSC	78.14(±6.44)	54.12(±2.77)	83.20(±2.51)	68.71(±7.83)	74.36(±3.99)	55.64(±3.58)
CorSC	85.35(±7.23)	54.07(±1.74)	79.26(±2.46)	72.84(±2.77)	76.09(±3.88)	54.05(±1.17)
MVASM	72.92(±6.26)	55.88(±9.34)	39.62(±2.56)	63.47(±4.73)	54.23(±7.96)	57.83(±2.76)
AKM^{3}C	95.93(±1.71)	56.85(±2.09)	79.25(±2.62)	80.31(±2.50)	69.24(±3.82)	60.80(±7.51)
FgMVC	**96.93(±0.20)**	**58.20(±1.54)**	**88.42(±1.52)**	**83.08(±1.95)**	**79.06(±3.25)**	**64.17(±9.66)**

Table 6 Clustering performance comparison in terms of NMI on six real-world datasets

NMI(%)	Handwritten	Wiki	100Leaves	COIL20	ORL	Prokaryotic Phyla
K-means(BestView)	70.11(±2.42)	52.49(±0.67)	78.15(±0.80)	71.69(±2.72)	84.58(±2.00)	28.92(±8.47)
SC(BestView)	64.14(±2.76)	52.79(±0.94)	76.38(±0.87)	80.86(±1.71)	85.09(±0.83)	15.72(±1.32)
KMM(BestView)	77.11(±3.85)	52.09(±1.03)	74.76(±1.07)	79.10(±2.90)	83.57(±2.35)	24.75(±1.88)
AASC	86.50(±0.00)	45.63(±0.42)	95.11(±0.62)	88.43(±0.74)	87.95(±1.12)	**35.01(±0.12)**
AMGL	84.52(±5.53)	46.22(±2.77)	88.31(±1.91)	91.27(±2.51)	89.09(±1.75)	2.17(±0.54)
MVSC	83.40(±3.64)	46.99(±1.25)	94.21(±0.70)	85.31(±3.91)	**89.34(±1.75)**	32.62(±1.81)
CorSC	79.60(±3.86)	48.85(±0.42)	91.88(±0.85)	82.35(±1.60)	89.19(±1.79)	23.88(±0.76)
MVASM	74.72(±3.50)	52.80(±11.09)	64.57(±2.04)	76.75(±2.02)	73.03(±10.59)	29.08(±1.71)
AKM^{3}C	91.93(±1.46)	52.51(±1.11)	92.88(±0.94)	90.40(±1.22)	83.38(±2.26)	32.04(±5.27)
FgMVC	**93.12(±0.38)**	**53.42(±1.90)**	**95.65(±0.56)**	**91.90(±0.87)**	88.62(±2.78)	27.36(±13.89)

Table 7 Clustering performance comparison in terms of Purity on six real-world datasets

Purity(%)	Handwritten	Wiki	100Leaves	COIL20	ORL	Prokaryotic Phyla
K-means(BestView)	71.87(±5.31)	60.99(±1.26)	59.45(±1.29)	59.69(±4.17)	72.23(±3.17)	66.43(±6.74)
SC(BestView)	71.17(±4.37)	60.77(±1.34)	57.77(±1.80)	74.05(±2.63)	73.98(±1.84)	57.89(±0.13)
KMM(BestView)	80.31(±5.13)	60.65(±0.98)	56.10(±1.84)	71.69(±3.60)	71.32(±3.12)	60.52(±1.81)
AASC	85.75(±0.00)	56.23(±0.44)	89.94(±1.06)	82.48(±1.56)	80.54(±1.90)	65.99(±0.32)
AMGL	83.46(±6.73)	48.18(±3.18)	80.71(±2.34)	85.89(±3.26)	79.22(±2.24)	57.42(±0.27)
MVSC	80.82(±5.36)	59.64(±2.06)	85.78(±2.15)	72.59(±7.20)	76.83(±3.79)	66.50(±2.10)
CorSC	86.21(±5.84)	59.46(±0.61)	82.48(±1.98)	75.23(±2.48)	79.86(±3.33)	63.35(±0.48)
MVASM	75.88(±4.88)	61.54(±10.18)	39.06(±2.48)	67.17(±4.45)	57.46(±8.23)	64.80(±1.15)
AKM^{3}C	95.93(±1.71)	60.63(±1.54)	83.37(±1.96)	82.99(±2.24)	73.77(±2.76)	68.60(±2.47)
FgMVC	**96.93(±0.20)**	**62.00(±1.65)**	**90.50(±1.24)**	**86.10(±1.85)**	**81.47(±2.85)**	**69.36(±7.28)**



(a) ACC (b) NMI (c) Purity

Fig. 5 Ablation study on three real-world datasets

5.5 Ablation study

As shown in Fig. 5, to verify the importance of the fine-grained fusion mechanism and the importance of reducing the overlap of sub-cluster structures, we conduct ablation studies. Four baselines were generated through ablation studies. These four baselines are special cases of our FgMVC method, which can be divided into two types, namely FgMVC_KM and FgMVC_KM.

FgMVC_KM is a multi-view method based on K-means, and its objective function is shown in (43). When FgMVC_KM treats clusters in different views equally, the weight $w_j^{(p)} = 1/v$, as shown in Aver_FgMVC_KM in Fig. 5. When FgMVC_KM automatically assigns contribution weights to clusters in different views, the weight $w_j^{(p)} = 1/\sqrt{\sum_i s_{ij}^* \left\| \boldsymbol{x}_i^{(p)} - \boldsymbol{v}_j^{(p)} \right\|_2^2}$, as shown in Auto_FgMVC_KM in Fig. 5.

$$\min_{\boldsymbol{U}\mathbf{1}=\mathbf{1},\boldsymbol{U}\geq 0} \sum_{p=1}^{v}\sum_{i=1}^{n}\sum_{j=1}^{c} w_j^{(p)} u_{ij} \left\| \boldsymbol{x}_i^{(p)} - \boldsymbol{v}_j^{(p)} \right\|_2^2 + \gamma \left\| \boldsymbol{U} \right\|_F^2 \quad (43)$$

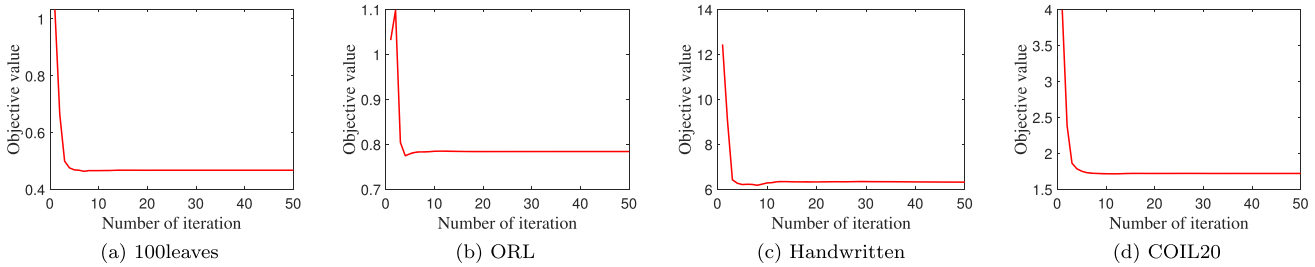
where $v_j^{(p)}$ is the j-th cluster in the p-th view, and $w_j^{(p)}$ is the weight of the j-th cluster in the p-th view. U is the consensus membership degree matrix, and the membership degree from the i-th sample to the j-th cluster is u_{ij}. Other variables have the same meaning as the variables in the objective function of FgMVC.

FgMVC_KMM is a multi-view method based on multi-prototypes representation, and its objective function is shown in (44). When FgMVC_KMM treats the sub-clusters in different views equally, then the weight $w_j^{(p)} = 1/v$, as shown in Aver_FgMVC_KMM in Fig. 5. When FgMVC_KMM automatically assigns contribution weights to sub-clusters in different views, the weight $w_j^{(p)} = 1/\sqrt{\sum_i s_{ij}^* \left\| \boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)} \right\|_2^2}$, as shown in Auto_FgMVC_KMM in Fig. 5.

$$\min_{\boldsymbol{S}^*\mathbf{1}=\mathbf{1},\boldsymbol{S}^*\geq 0,\ \boldsymbol{S}^*\in\Omega} \sum_{p=1}^{v}\sum_{i=1}^{n}\sum_{j=1}^{m} w_j^{(p)} s_{ij}^* \left\| \boldsymbol{x}_i^{(p)} - \boldsymbol{a}_j^{(p)} \right\|_2^2 + \gamma \left\| \boldsymbol{S}^* \right\|_F^2 \quad (44)$$

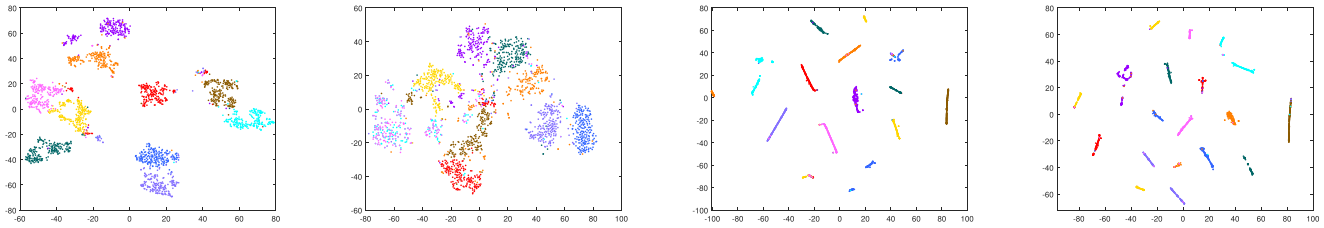
where the variables in (44) have the same meaning as those in FgMVC. Through the experimental results in Fig. 5, we can draw the following conclusions:

1) Comparing the average weight clustering methods (e.g., Aver_FgMVC_KM and Aver_FgMVC_KMM) with the auto-weight clustering methods (e.g., Auto_FgMVC_KM and Auto_FgMVC_KMM), we find that the auto-weight clustering methods can obtain better clustering results. Therefore, we conclude that compared with the coarse-grained fusion mechanism, the fine-grained fusion mechanism can obtain more comprehensive complementary information by acquiring more detailed fusion objects.



(a) 100leaves (b) ORL (c) Handwritten (d) COIL20

Fig. 6 Convergence curve of FgMVC on four real-world datasets



(a) View-1 (b) View-2 (c) AKM^{3}C (d) FgMVC

Fig. 7 Visualization of the clustering results of Handwritten dataset with t-SNE

2) Comparing the single prototype representation methods (e.g., Aver_FgMVC_KM and Auto_FgMVC_KM) and the multi-prototypes representation methods (e.g., Aver_FgMVC_KMM and Auto_FgMVC_KMM), we find that the multi-prototypes representation methods can achieve better clustering results. Because the multi-prototypes representation methods can find non-spherical clusters in the original data, which can better fit the data.
3) Compared with Auto_FgMVC_KMM, our FgMVC reduces the overlap between the sub-cluster structures, thus resulting in a better multi-prototypes representation. It can be concluded from the clustering result in Fig. 5 that our FgMVC can be better improved by taking into account the reduction in the overlap between sub-cluster structures.

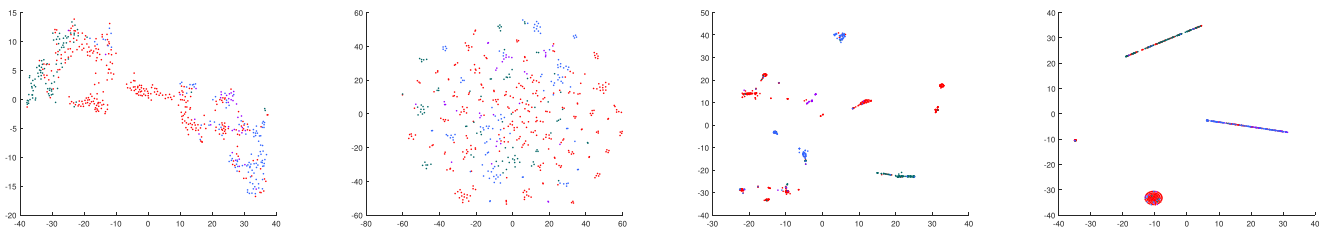
5.6 Convergence study

We not only theoretically analyze the convergence of the algorithm but also collect the objective function values on four real-world datasets in the process of iteration. As shown in Fig. 6, the algorithm converges rapidly within 10 iterations, which proves that the algorithm has good convergence. In Fig. 6, the X-axis represents the number of iterations, and the Y-axis represents the value of the objective function.

5.7 Clustering visualization

The visual clustering results of two real-wolrd datasets (e.g., Handwritten and Prokaryotic Phyla) are shown in Figs. 7 and 8 by using t-SNE. t-SNE is a classical non-linear dimensionality reduction algorithm that is used to visualize clustering results [43, 44]. In Fig. 7a and b, we visualize two views of the Handwritten dataset with t-SNE. Figure 8a and b show two views of the Prokaryotic Phyla dataset visualized with t-SNE. In Figs. 7c and 8c, we conducted t-SNE for the connection probability matrix $\boldsymbol{S}^*$ obtained by AKM^{3}C. In Figs. 7d and 8d, we conducted t-SNE for the connection probability matrix $\boldsymbol{S}^*$ obtained by our FgMVC. The points with the same color belong to the same cluster. We emphasize that the color corresponds to the truth label.

Figure 7c and d show that AKM^{3}C and FgMVC obtain clear boundaries between clusters. However, some clusters are not pure, where a small number of samples with clustering errors can be observed. This phenomenon also corresponds to the clustering results obtained by AKM^{3}C and FgMVC in Tables 5–7. In Fig. 8a and b, there is a problem of data imbalance in the Prokaryotic Phyla dataset. Moreover, there is no clear boundary between clusters within each view, which is why most clustering methods in Tables 5–7 do not obtain satisfactory clustering results. In Fig. 8c and d, it can be seen that compared with AKM^{3}C, FgMVC obtain a clearer boundary between clusters.



(a) View-1 (b) View-2 (c) AKM^{3}C (d) FgMVC

Fig. 8 Visualization of the clustering results of Prokaryotic Phyla dataset with t-SNE

6 Conclusion

In this paper, a novel fine-grained multi-view clustering method with robust multi-prototypes representation (FgMVC) is proposed. Unlike most current multi-view clustering methods, we refine the granularity of fusion objects from the entire sample space to the local sub-cluster within each view. Specifically, the sample space of each view is divided into many sub-clusters based on multi-prototypes representation, where the contribution weights are automatically assigned to sub-clusters based on their clustering capacities. Thus, the influence of valuable information in each view is reinforced, and the influence of misleading information is reduced. To reduce the effect of noisy data and judge the partition of samples more clearly, the robust multi-prototypes representation is obtained by maximizing the distance between the prototypes of different sub-clusters within each view. Finally, the robust multi-prototypes representation, the fine-grained multi-view fusion, and the clustering process are integrated into a unified framework. We further analyze the initialization, computational complexity, and convergence. It can be observed from the experimental results that our method performs better than the comparison algorithms on most datasets. This phenomenon proves that our strategy of refining the fusion granularity in multi-view clustering is effective. In the next step, we will consider the combination of deep autoencoder to further improve the performance of fine-grained multi-view clustering.

Acknowledgments This work has been partially supported by grants from the National Natural Science Foundation of China (62072151), the Key Project supported by the Joint Funds of the National Natural Science Foundation of China (U20A20228), the Zhejiang Basic Public Welfare Research Project (LGN18F020002), the Natural Science Foundation of Zhejiang Province (LR20F020002), the Anhui Provincial Natural Science Fund for Distinguished Young Scholars (2008085J30), the Fundamental Research Funds for Central Universities of China (JZ2019HGPA0102), the Huzhou Public Welfare Applied Research Project (2021GZ05), and the Huzhou University Graduate Scientific Research Innovation Project (2022KYCX44).

References

1. Xiao H, Chen Y, Shi X (2021) Knowledge graph embedding based on multi-view clustering framework. IEEE Trans Knowl Data Eng 33(2):585–596
2. Wang Q, Chen M, Nie F, Li X (2020) Detecting coherent groups in crowd scenes by multiview clustering. IEEE Trans Pattern Anal Mach Intell 42(1):46–58
3. Yu H, Tang J, Wang G, Gao X (2021) A novel multi-view clustering method for unknown mapping relationships between cross-view samples. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery and data mining, pp 2075–2083
4. Tian J, Zhao J, Zheng C (2021) Clustering of cancer data based on Stiefel manifold for multiple views. BMC Bioinforma 22(1):268
5. Liu X, Li M, Tang C, Xia J, Xiong J, Liu L, Kloft M, Zhu E (2021) Efficient and effective regularized incomplete multi-view clustering. IEEE Trans Pattern Anal Mach Intell 43(8):2634–2646
6. Lv J, Kang Z, Wang B, Ji L, Xu Z (2021) Multi-view subspace clustering via partition fusion. Inf Sci 560:410–423
7. Zhang C, Fu H, Hu Q, Cao X, Xie Y, Tao D, Xu D (2020) Generalized latent multi-view subspace clustering. IEEE Trans Pattern Anal Mach Intell 42(1):86–99
8. Yan H, Liu S, Yu PS (2019) From joint feature selection and self-representation learning to robust multi-view subspace clustering. In: Proceedings of the 19th IEEE international conference on data mining, ICDM, pp 1414–1419
9. Wang H, Wang Y, Zhang Z, Fu X, Zhuo L, Xu M, Wang M (2021) Kernelized multiview subspace analysis by self-weighted learning. IEEE Trans Multimed 23:3828–3840
10. Feng L, Liu W, Meng X, Zhang Y (2021) Re-weighted multi-view clustering via triplex regularized non-negative matrix factorization. Neurocomputing 464:352–363
11. Hu Z, Nie F, Wang R, Li X (2020) Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding. Inform Fus 55:251–259
12. Zong L, Zhang X, Liu X (2018) Multi-view clustering on unmapped data via constrained nonnegative matrix factorization. Neural Netw 108:155–171
13. Wang H, Yang Y, Liu B (2020) GMC: graph-based multi-view clustering. IEEE Trans Knowl Data Eng 32(6):1116–1129
14. Huang S, Xu Z, Tsang I, Kang Z (2020) Auto-weighted multi-view co-clustering with bipartite graphs. Inf Sci 512:18–30
15. Yin H, Hu W, Zhang Z, Lou J, Miao M (2021) Incremental multi-view spectral clustering with sparse and connected graph learning. Neural Netw 144:260–270
16. Xie J, Girshick R, Farhadi A (2016) Unsupervised deep embedding for clustering analysis. In: Proceedings of the 33nd international conference on machine learning, ICML, vol 48, pp 478–487
17. Dizaji KG, Herandi A, Deng C, Cai W, Huang H (2017) Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: Proceedings of the IEEE international conference on computer vision, ICCV, pp 5747–5756
18. Tu W, Zhou S, Liu X, Guo X, Cai Z, Zhu E, Cheng J (2021) Deep fusion clustering network. In: Proceedings of the 35th AAAI conference on artificial intelligence, AAAI, pp 9978–9987
19. Xie Y, Lin B, Qu Y, Li C, Zhang W, Ma L, Wen Y, Tao D (2021) Joint deep multi-view learning for image clustering. IEEE Trans Knowl Data Eng 33(11):3594–3606
20. Wen J, Wu Z, Zhang Z, Fei L, Zhang B, Xu Y (2021) Structural deep incomplete multi-view clustering network. In: Proceedings of the 30th ACM international conference on information and knowledge management, CIKM, pp 3538–3542
21. Wen J, Zhang Z, Zhang Z, Wu Z, Fei L, Xu Y, Zhang B (2020) Dimc-net: deep incomplete multi-view clustering network. In: Proceedings of the 28th ACM international conference on multimedia, ACM MM, pp 3753–3761
22. Li Z, Wang Q, Tao Z, Gao Q, Yang Z (2019) Deep adversarial multi-view clustering network. In: Proceedings of the 28th international joint conference on artificial intelligence, IJCAI, pp 2952–2958
23. Wen J, Zhang Z, Zhang Z, Fei L, Wang M (2021) Generalized incomplete multiview clustering with flexible locality structure diffusion. IEEE Trans Cybern 51(1):101–114
24. Han J, Xu J, Nie F, Li X (2020) Multi-view k-means clustering with adaptive sparse memberships and weight allocation. IEEE Trans Knowl Data Eng, 1–1

25. Hu Y, Song Z, Wang B, Gao J, Sun Y, Yin B (2021) Akm^{3}c: adaptive k-multiple-means for multi-view clustering. IEEE Trans Circ Syst Vid Technol 31(11):4214–4226
26. Yin H, Hu W, Li F, Lou J (2021) One-step multi-view spectral clustering by learning common and specific nonnegative embeddings. Int J Mach Learn Cybern 12(7):2121–2134
27. Yu X, Liu H, Wu Y, Zhang C (2021) Fine-grained similarity fusion for multi-view spectral clustering. Inf Sci 568:350–368
28. Zhao Q, Zong L, Zhang X, Liu X, Yu H (2020) Multi-view clustering via clusterwise weights learning. Knowl-Based Syst 193:105459
29. Gao J, Han J, Liu J, Wang C (2013) Multi-view clustering via joint nonnegative matrix factorization. In: Proceedings of the 13th SIAM international conference on data mining, ICDM, pp 252–260
30. Xue Z, Li G, Wang S, Zhang C, Zhang W, Huang Q (2015) GOMES: a group-aware multi-view fusion approach towards real-world image clustering. In: Proceedings of the IEEE international conference on multimedia and expo, ICME, pp 1–6
31. Nie F, Wang C-L, Li X (2019) K-multiple-means: a multiple-means clustering method with specified K clusters. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, KDD, pp 959–967
32. Li Z, Nie F, Chang X, Nie L, Zhang H, Yang Y (2018) Rank-constrained spectral clustering with flexible embedding. IEEE Trans Neural Netw Learn Syst 29(12):6073–6082
33. Fan K (1949) On a theorem of weyl concerning eigenvalues of linear transformations I. Proc Natl Acad Sci USA 35(11):652
34. Nie F, Wang X, Jordan MI, Huang H (2016) The constrained laplacian rank algorithm for graph-based clustering. In: Proceedings of the 30th AAAI conference on artificial intelligence, AAAI, pp 1969–1976
35. Nie F, Wang X, 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, KDD, pp 977–986
36. Nie F, Huang H, Cai X, Ding CHQ (2010) Efficient and robust feature selection via joint 2,1-norms minimization. In: Proceedings of the 24th annual conference on neural information processing systems, NIPS, pp 1813–1821
37. MacQueen J et al (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, vol 1, pp 281–297
38. Ng A, Jordan M, Weiss Y (2001) On spectral clustering: analysis and an algorithm. Adv Neural Inform Process Syst 14:849–856
39. Kumar A, Rai P, III HD (2011) Co-regularized multi-view spectral clustering. In: Proceedings of the 24th international conference on neural information processing systems, NeurIPS, pp 1413–1421
40. Huang H-C, Chuang Y-Y, Chen C-S (2012) Affinity aggregation for spectral clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR, pp 773–780
41. Li Y, Nie F, Huang H, Huang J (2015) Large-scale multi-view spectral clustering via bipartite graph. In: Proceedings of the 29th AAAI conference on artificial intelligence, AAAI, pp 2750–2756
42. Nie F, Li J, Li X (2016) Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification. In: Proceedings of the 25th international joint conference on artificial intelligence, IJCAI, pp 1881–1887
43. Rauber PE, Falcão AX, Telea AC (2016) Visualizing time-dependent data using dynamic t-sne. In: Proceedings of 18th eurographics conference on visualization, pp 73–77
44. Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9:11

JrnlID 10489_ArtID 3898_Proof#1 - 25/06/2022

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES:

Q1. Authors biography and photos are desired. Please provide if necessary.
Q2. Please check affiliations if captured and presented correctly.
Q3. City was provided 1 and 2. Please check if captured and presented correctly.