



Multiple kernel-based anchor graph coupled low-rank tensor learning for incomplete multi-view clustering

Senhong Wang¹ · Jiangzhong Cao¹ · Fangyuan Lei² · Jianjian Jiang² · Qingyun Dai² · Bingo Wing-Kuen Ling¹

Accepted: 6 May 2022 / Published online: 2 June 2022
© The Author(s) 2022

Abstract

Incomplete Multi-View Clustering (IMVC) attempts to give an optimal clustering solution for incomplete multi-view data that suffer from missing instances in certain views. However, most existing IMVC methods still have various drawbacks in practical applications, such as arbitrary incomplete scenarios cannot be handled; the computational cost is relatively high; most valuable nonlinear relations among samples are often ignored; complementary information among views is not sufficiently exploited. To address the above issues, in this paper, we present a novel and flexible unified graph learning framework, called Multiple Kernel-based Anchor Graph coupled low-rank Tensor learning for Incomplete Multi-View Clustering (MKAGT_IMVC), whose goal is to adaptively learn the optimal unified similarity matrix from all incomplete views. Specifically, according to the characteristics of incomplete multi-view data, MKAGT_IMVC innovatively improves an anchor selection strategy. Then, a novel cross-view anchor graph fusion mechanism is introduced to construct multiple fused complete anchor graphs, which captures more the intra-view and inter-view nonlinear relations. Moreover, a graph learning model combining low-rank tensor constraint and consensus graph constraint is developed, where all fused complete anchor graphs are regarded as prior knowledge to initialize this model. Extensive experiments conducted on eight incomplete multi-view datasets clearly show that our method delivers superior performance relative to some state-of-the-art methods in terms of clustering ability and time-consuming.

Keywords Incomplete multi-view clustering · Anchor graph · Multiple kernel · Low-rank tensor constraint · Consensus graph constraint

1 Introduction

In many practical applications, since the information of data is usually collected from various sensors or diverse processing methods, this data characterized by multiple perspectives is regarded as multi-view data in machine learning communities [1]. For instance, a specific piece of news can be reported simultaneously by multiple news organizations [2]; an image can be described with multiple visual features, such as GIST, PHOG, LBP, etc. [3]. Generally speaking, multi-view data is more comprehensive than an individual view because multiple views contain complementary and consistent information between views [4]. Thus, this means that making efficient use of

information between views can elevate the performance of various machine learning.

Multi-View Clustering (MVC) which can more accurately partition the sample data into corresponding groups by efficiently utilizing the complementarity and consistency information among multiple views, has become an important research topic due to the wide application of multi-view data. Most available literature reports may be roughly classified into three types based on the involved technologies, including matrix factorization-based methods [5, 6], graph-based methods [7–12], and deep learning-based methods [13, 14]. It should be noted that the aforementioned MVC methods usually need to be satisfied with a basic assumption that all instances are fully observed in each view. In many real-world applications, however, each view may lose some instances due to various reasons, such as human error or sensor failure. For example, as for a document, the texts and images are usually regarded as two views, while some documents may be missing texts or images [15]; the magnetic resonance images and blood test results can both be used

✉ Jiangzhong Cao
cjz510@gdut.edu.cn

to diagnose diseases, while some patients may have only one of them [16]. Thus, these data are generally referred to as incomplete multi-view data. The missing instances not only lead to a significant decrease in the information quality of each view but also make it more difficult to mine the information of complementarity and consistency between views. These factors make the conventional MVC methods unavoidably significantly degrade or even fail for mining the incomplete multi-view data, especially data with a large sample incomplete rate [17]. Since incomplete multi-view data is important and complex, Incomplete Multi-View Clustering (IMVC) has received widespread attention in recent years. To address above issues, a variety of IMVC methods have been proposed under different theoretical frameworks [18]. Roughly speaking, in terms of the techniques involved, existing IMVC methods can be categorized into three categories: matrix factorization-based methods, deep learning-based methods, and graph-based methods.

The matrix factorization-based methods [17, 19, 20] aim to learn the optimal low-dimensional consistency representation directly from multiple incomplete views via the matrix factorization technique. For example, as a pioneering work dealing with missing instances in all views, Li et al. [17] proposed Partial multi-View Clustering (PVC), which adopts non-negative matrix factorization (NMF) and l_1 norm regularization to learn the potentially consistent representation from two views. Unfortunately, PVC can only handle the incomplete multi-view data of two views and ignores the difference in the quality of the information of each view. By combining weighted semi-NMF and $l_{2,1}$ norm regularization, Hu and Chen [19] further introduced Doubly Aligned Incomplete Multi-view Clustering (DAIMC) to learn the optimal unified latent feature matrix from all views. Nevertheless, DAIMC fails to explore the underlying relations between instances in each view. Wen et al. [20] provided a more effective and flexible IMVC framework, called Generalized Incomplete Multi-view Clustering with Flexible Locality Structure Diffusion (GIMC_FLSD), which integrates the local structure preserving and the individual representation learning to obtain the optimal consistent representation. The original data generally presents the nonlinear structure, while GIMC_FLSD directly encodes these data in a linear space, resulting in failure to capture more nonlinear relations among samples.

The deep learning-based methods [21–26] build a deep neural network to adaptively learn the optimal consistent representation in nonlinear space. For example, inspired by the idea of Generative Adversarial Networks (GAN), [21–23] leverage the common representation generated by GAN to infer the missing data and learn the optimal consistent representation simultaneously. Lin et al. [24] adopted contrastive learning and dual prediction mechanism to

mutually boost the consistency learning and data recovery. However, the above deep methods require a large number of aligned samples that appear in all views to guarantee good performance. To further improve the effectiveness and flexibility of deep method, [25] and [26] integrate the view-specific encoders, graph embedding, fusion layer, and view-specific decoders into a joint pre-training network to deal with the incomplete multi-view data. Nevertheless, the above two methods cannot directly execute the clustering task in an end-to-end manner.

The graph-based methods [15, 16, 27–34] focus on finding the optimal fusion graph from different graphs calculated by all views, where these graphs cover the relationships among all samples. According to the different initialization construction strategies of the graph, we roughly summarize them as the available instance-based method, imputation-based method, and anchor-based method. From the perspective of the available instance-based method, [15] and [16] directly use the available instances of each view to construct a low-dimensional similarity graph and exploit the index matrix recording the index information for the corresponding view to restore the low-dimensional similarity graph to a complete similarity graph. Then the complete similarity graph is integrated into the optimization framework of consistent representation learning. However, this kind of method generally has a high computational cost, limiting their practical applications. As the representative imputation-based methods, [27–31] impute each incomplete graph and learn the optimal consensus clustering matrix simultaneously. To further explore the hidden information and interpretability of the missing views, Wen et al. [32] proposed IMVTSC-MVI that recovers the missing instances rather than the missing graph. However, the imputation-based method may introduce noise, which cannot guarantee the improvement of clustering performance. As for the anchor-based method, recent references [33] and [34] proposed to select samples appearing in all views as anchor points which are used to build the anchor graph of the corresponding view. Although the above two methods can reduce the computational complexity, especially on large-scale dataset, they require a certain amount of samples that appear in all views, which makes them inflexible.

To sum up, although most of the aforementioned methods have obtained state-of-the-art performances on incomplete multi-view clustering tasks, they still have following several limitations in practical applications. First, many methods are inflexible, only suitable for special incomplete scenarios, and usually have relatively high computational cost. Second, the underlying and significant nonlinear relations among samples are ignored. Third, the complementary information among all views has not been sufficiently exploited. To address these issues discussed above, in this article, we propose a novel and flexible

unified graph learning framework, called Multiple Kernel-based Anchor Graph coupled low-rank Tensor learning for Incomplete Multi-View Clustering (MKAGT_IMVC). Specifically, as shown in Fig. 1, MKAGT_IMVC mainly includes the following three key steps. Firstly, we adopt the improved DAS to efficiently generate a batch of representative anchor points from incomplete multi-view data. The selected anchor points are exploited to build a complete anchor graph for each view and directly coalesces all complete anchor graphs to form the fused complete anchor graph, which can greatly increase the flexibility to deal with any incomplete situations and effectively reduce computational cost. Secondly, under the cross-view anchor graph fusion mechanism, given multiple pre-defined kernel functions, the proposed method can obtain a corresponding number of fused complete anchor graphs which record a large number of valuable nonlinear relationships among samples. Thirdly, we develop a graph learning model based on subspace representation, in which all the previous fused complete anchor graphs are regarded as prior knowledge to initialize this model. To guarantee that the complementary and consistent information of multiple views is fully explored, low-rank tensor constraint and consensus graph constraint are jointly imposed on this graph learning model. Therefore, MKAGT_IMVC has the potential to adaptively learn the optimal unified similarity matrix by solving the above-mentioned graph learning model through the Alternating Direction Method of Multiplier (ADMM).

The main contributions and novelty of this article are listed as follows:

1. We innovatively exploit a simple and efficient anchor selection approach to generate a batch of representative anchors from incomplete multi-view data, where these anchors are used to construct view-wise complete anchor graphs.
2. Instead of combining view-wise similarity graphs via imputing missing samples, we provide a novel cross-view anchor graph fusion paradigm to construct the fused complete anchor graph as prior knowledge for incomplete multi-view data, where the anchor graph fusion mechanism not only well captures more nonlinear relations between samples, but also adequately excavates the intra-view and inter-view information.
3. We present a novel and effective unified graph learning framework (MKAGT_IMVC) that incorporates some prior knowledge into the subspace representation based graph learning model for incomplete multi-view clustering, where the prior knowledge can be used as the initial input for the model. Moreover, this model integrates low-rank tensor constraint and consensus graph constraint to ensure the optimal unified similarity matrix.
4. We conduct extensive experiments that clearly show our algorithm delivers superior performance relative to some state-of-the-art methods in terms of clustering ability and time-consuming.

2 Notations and related works

In this section, we first introduce the notations used throughout this paper. Then, we briefly analyze several closely related works of the proposed method.

2.1 Notations

In this paper, we use italic lowercase letters, bold italic lowercase letters, bold italic capital letters, and bold calligraphy letters to denote scalars, vectors, matrices, and tensors, respectively (i.e., a , \mathbf{a} , \mathbf{A} , \mathcal{A}). We denote $\mathbf{I} \in \mathbb{R}^{n \times n}$ as the identity matrix with an compatible size. We denote $\mathbf{1}$ as the all-one vector with an appropriate length. For convenience, we give some basic notations and their descriptions in Table 1.

2.2 Related works

2.2.1 Anchor-based Partial Multi-view Clustering (APMC)

APMC [33] utilizes anchor points to construct intra-view and inter-view similarity to address the IMVC problem. Specifically, APMC mainly consists of two steps, i.e., constructing the fused similarity graph by anchor points and obtaining the cluster indicators via spectral clustering. APMC first selects the common instances showing in all views to build an instance-to-anchor graph for each view and then integrates all instance-to-anchor graphs to create the unified similarity matrix \mathbf{S} . Then, APMC conducts

Table 1 Basic notations and their descriptions

Notations	Descriptions
$\mathbf{A}, \mathbf{a}, a$	matrix, vector, scalar
$\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$	three-order tensor
$\mathcal{A}^{(k)} = \mathcal{A}(:, :, k)$	the k -th frontal slice of tensor \mathcal{A}
$\mathcal{A}_f = \text{fft}(\mathcal{A}, [1, 3])$	fast Fourier transform along the third dimension
$\mathcal{A} * \mathcal{B}$	the t-product between tensor \mathcal{A} and tensor \mathcal{B}
$\ \mathcal{A}\ _{\otimes}$	the nuclear norm of tensor \mathcal{A}
$\ \mathbf{A}\ _F, \ \mathbf{A}\ _1$	Frobenius norm and l_1 -norm of the matrix \mathbf{A}
$\ \mathbf{A}\ _{\infty}, \ \mathbf{A}\ _*$	infinity norm and nuclear norm of the matrix \mathbf{A}
$\mathcal{A}^T, \mathbf{A}^T, \mathbf{a}^T$	the transpose of tensor \mathcal{A} , matrix \mathbf{A} , and vector \mathbf{a}
$\text{Tr}(\mathbf{A})$	the track of the matrix \mathbf{A}
$\text{rank}(\mathbf{A})$	the rank of the matrix \mathbf{A}

spectral clustering on the unified similarity matrix S . The model of spectral clustering is written as:

$$\max_{F^T F = I} \text{Tr}(F^T S F), \quad (1)$$

where $F \in \mathbb{R}^{n \times c}$ and c is the cluster indicator matrix and the number of clusters, respectively. Since the unified similarity matrix S has the property of the anchor graph, the cluster indicator matrix F can be derived by performing Singular Value Decomposition [34]. Finally, APMC executes k -means on F to get the clustering results. More details can be found in [33].

Although APMC can reduce the computational complexity, especially on the large-scale dataset, it requires a certain amount of samples that appear in all views, which makes them inflexible. In addition, performing spectral clustering directly on the unified similarity matrix S cannot guarantee adaptively obtaining optimal clustering results.

2.2.2 Low-rank representation based multi-view subspace clustering

For this paper, in fields of low-rank representation (LRR) based multi-view subspace clustering, we mainly refer to the graph learning based methods that learn a unified graph from multiple views by imposing a variety of different constraints, e.g., low-rank constraint, sparse constraint, and consensus constraint, etc. Consequently, given a multi-view dataset $\{X^{(v)} \in \mathbb{R}^{d_v \times n}\}_{v=1}^V$, the general framework of LRR based multi-view subspace clustering can be formulated as follows [35, 36]:

$$\begin{aligned} \min_{S^{(v)}, E^{(v)}, S^*} & \left\{ \begin{array}{l} \sum_{v=1}^V \|S^{(v)}\|_* + \alpha \sum_{v=1}^V \Psi(E^{(v)}) \\ + \beta \sum_{v=1}^V \text{Cons}(S^{(v)}, S^*) \end{array} \right\} \\ \text{s.t.} & \quad X^{(v)} = X^{(v)} S^{(v)} + E^{(v)}, \\ & \quad S^{(v)} \geq 0, \text{diag}(S^{(v)}) = 0, (S^{(v)})^T \mathbf{1} = \mathbf{1}, \end{aligned} \quad (2)$$

where $\alpha > 0$ and $\beta > 0$ are the trade-off parameters designed to balance the value of the corresponding term, $E^{(v)} \in \mathbb{R}^{d_v \times n}$ denotes the reconstruction error matrix in the v -th view, $S^{(v)} \in \mathbb{R}^{n \times n}$ and $S^* \in \mathbb{R}^{n \times n}$ denotes the subspace representation matrix in the v -th view and the consistent representation among multiple subspace representation matrices, respectively. When $S^{(v)}$ is imposed the following constraints: $S^{(v)} \geq 0$, $\text{diag}(S^{(v)}) = 0$, $(S^{(v)})^T \mathbf{1} = \mathbf{1}$, $S^{(v)}$ can be regarded as similarity matrix reflecting the distance among instances [36]. $\|S^{(v)}\|_*$ is the regularization term used low-rank representation which can best capture the intrinsic geometric structures in v -th view. $\Psi(E^{(v)})$ simulates different noises with various norm constraints, e.g., $\|E^{(v)}\|_1$, $\|E^{(v)}\|_{2,1}$, and $\|E^{(v)}\|_F^2$. $\text{Cons}(S^{(v)}, S^*)$ is a function which learn the consistent

representation S^* from multiple view-specific subspace representations $S^{(v)}$.

However, there are several drawbacks in the model of (2) to some extent. Firstly, this model requires that all views are complete. In other words, the entries of all input samples should be fully observed from multiple views. Secondly, the real-world datasets usually show the nonlinear structure. Yet the subspace representation matrices are directly encoded on the linear space in this model, which makes the model unable to handle nonlinear data well. Thirdly, this model cannot effectively learn the complementary information from all the views to improve the clustering performance.

3 Preliminaries

3.1 Problem definition

As incomplete multi-view data, each sample not only is characterized by multiple views, but also usually suffers from missing some views. To facilitate the discussion, We define ‘‘instance’’ as the feature representation of samples in corresponding views. Assume that we are given a dataset $X = \{X^{(1)}, X^{(2)}, \dots, X^{(V)}\}$ with V views, where the v -th view is denoted as $X^{(v)} = [x_1^{(v)}, x_2^{(v)}, \dots, x_n^{(v)}] \in \mathbb{R}^{d_v \times n}$ composed of available and missing instances. $x_i^{(v)} \in \mathbb{R}^{d_v}$ is denoted as the i -th instance in the v -th view. Similarly, $Y^{(v)} \in \mathbb{R}^{d_v \times n_v}$ and $y_i^{(v)} \in \mathbb{R}^{d_v}$ are denoted as the available instances matrix and the i -th available instance in the v -th view, respectively. n and n_v separately stands for the number of samples and available instances in the v -th view. d_v is the dimensionality of the feature representation in the v -th view. c represents the number of clusters.

3.2 Tensor nuclear norm

For our method, the tensor nuclear norm is the key component to improve the clustering performance. Therefore, it is necessary to introduce the related definitions to help understand the concept of the tensor nuclear norm. We denote $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ as the identity tensor whose first frontal slice is an identity matrix of size $n_1 \times n_1$ and the remaining frontal slices are zero. The transpose of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is $\mathcal{A}^T \in \mathbb{R}^{n_2 \times n_1 \times n_3}$. A tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is orthogonal which satisfies $\mathcal{A}^T * \mathcal{A} = \mathcal{A} * \mathcal{A}^T = \mathcal{I}$. More details of the tensor nuclear norm can be found in [37, 38].

3.2.1 Tensor Singular Value Decomposition (t-SVD)

Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. The tensor Singular Value Decomposition (t-SVD) of \mathcal{A} is defined as:

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T, \quad (3)$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensor, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor whose every frontal slice is a diagonal matrix.

3.2.2 t-SVD based Tensor Nuclear Norm (t-SVD-TNN)

For the tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. The t-SVD based Tensor Nuclear Norm (t-SVD-TNN) of \mathcal{A} , denoted as $\|\mathcal{A}\|_{\otimes}$, is defined as:

$$\|\mathcal{A}\|_{\otimes} = \sum_{k=1}^{n_3} \|\mathcal{A}_f^{(k)}\|_* = \sum_{k=1}^{n_3} \sum_{i=1}^{\min(n_1, n_2)} |\mathcal{S}_f^{(k)}(i, i)|, \quad (4)$$

where $\sum_{k=1}^{n_3} \|\mathcal{A}_f^{(k)}\|_*$ denotes the sum of singular values of entire the frontal slices of \mathcal{A}_f , $\mathcal{S}_f^{(k)}$ can be obtained by performing Singular Value Decomposition on $\mathcal{A}_f^{(k)}$, i.e., $\mathcal{A}_f^{(k)} = \mathcal{U}_f^{(k)} * \mathcal{S}_f^{(k)} * \mathcal{V}_f^{(k)T}$, and $\mathcal{S}_f^{(k)}(i, i)$ is the i -th singular value of $\mathcal{A}_f^{(k)}$.

4 The proposed method

In this section, we elaborate a novel approach, termed Multiple Kernel-based Anchor Graph coupled low-rank Tensor learning for Incomplete Multi-View Clustering (MKAGT_IMVC), which effectively learns the consistent representation from missing views. Specifically, MKAGT_IMVC mainly consists of three key steps, i.e., generating anchor points for each missing view, constructing multiple fused complete anchor graphs, and developing the representation-based graph learning model. The general framework of our MKAGT_IMVC method is shown in Fig. 1.

4.1 Anchor selection of incomplete multi-view data

Recently, some research findings [12, 33, 34, 39] suggest that the performance of multi-view clustering can be effectively improved by adopting the anchor-based strategy to construct the anchor graphs, where these anchor graphs are regarded as the prior knowledge that can be used as the initial input for multi-view learning. In general, for multi-view data, there are two frequently used methods for anchor selection, i.e., random policy and k -means policy. The random policy adopts random sampling to select a fraction of samples from multi-view data as anchor points. Although the random policy is simple and efficient, it cannot guarantee that the selected anchor points are significantly representative, which leads to great instability in the final results. By contrast, the k -means policy makes the clustering centroids as anchor points, which are more representative than random selection. Specifically, it first

concatenates the features of all views as a joint feature, and then k -means is performed on the joint feature to get the joint centroids of the clustering. Finally, The multiple groups of centroids, which are obtained through splitting the joint centroids by views, are regarded as the anchor points of each view [39]. However, k -means is sensitive to the initialization of origin centroids, resulting in instability like the random sampling. To eliminate the drawback, the k -means requires numerous independent iterative calculations. Intuitively, samples belonging to the same cluster usually have similar feature representations, while the samples not in the same cluster have significant differences in feature representations. Based on this motivation, Li et al. [12] proposed an anchor selection strategy, called directly alternate sampling method (DAS) that makes a compromise between the efficiency of the random policy and the certainty of the k -means policy. The original intention of DAS is to conduct the anchor selection strategy after concatenating the features of all views in the complete multi-view data. However, when a large number of instances are missing in some views, adopting the manner of concatenating first and then anchor points selection causes DAS to inevitably degenerate or even fail. For incomplete multi-view data, the work in [33, 34] proposed to regard the samples appearing in all views as anchor points. Unfortunately, samples appearing in all views are often rare or non-existent. Considering an assumption of multi-view clustering that the cluster structure of all views should be analogous [36]. Inspired by this motivation, we extend the DAS method and apply it to the incomplete multi-view data. To be specific, for all views, we first obtain the available instance matrix $\mathbf{Y}^{(v)} \in \mathbb{R}^{d_v \times n_v}$ by removing the missing instances in $\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$. Then, we can calculate the v -th intuitive score $\mathbf{s}^{(v)} \in \mathbb{R}^{n_v}$ according to the following formula:

$$s_i^{(v)} = \sum_{j=1}^{d_v} Tra(\mathbf{Y}_{ij}^{(v)}), \quad (5)$$

where $Tra(\cdot)$ denotes a non-negative operation for an instance, which each element in the feature representation subtract the smallest element. The motivation of (5) is that the feature representation of instances belonging to the same cluster usually has relatively small differences than the instances not in the same cluster. Therefore, we can get this information naturally, that is, instances of the same cluster have similar scores calculated by summing the feature values. To this end, we choose anchor points based on the scores of each instance for all views. Definitely, we pick the instance with the largest score as the initial anchor point in the v -th view according to the following formula:

$$\arg \max_i s_i^{(v)}, \quad (6)$$

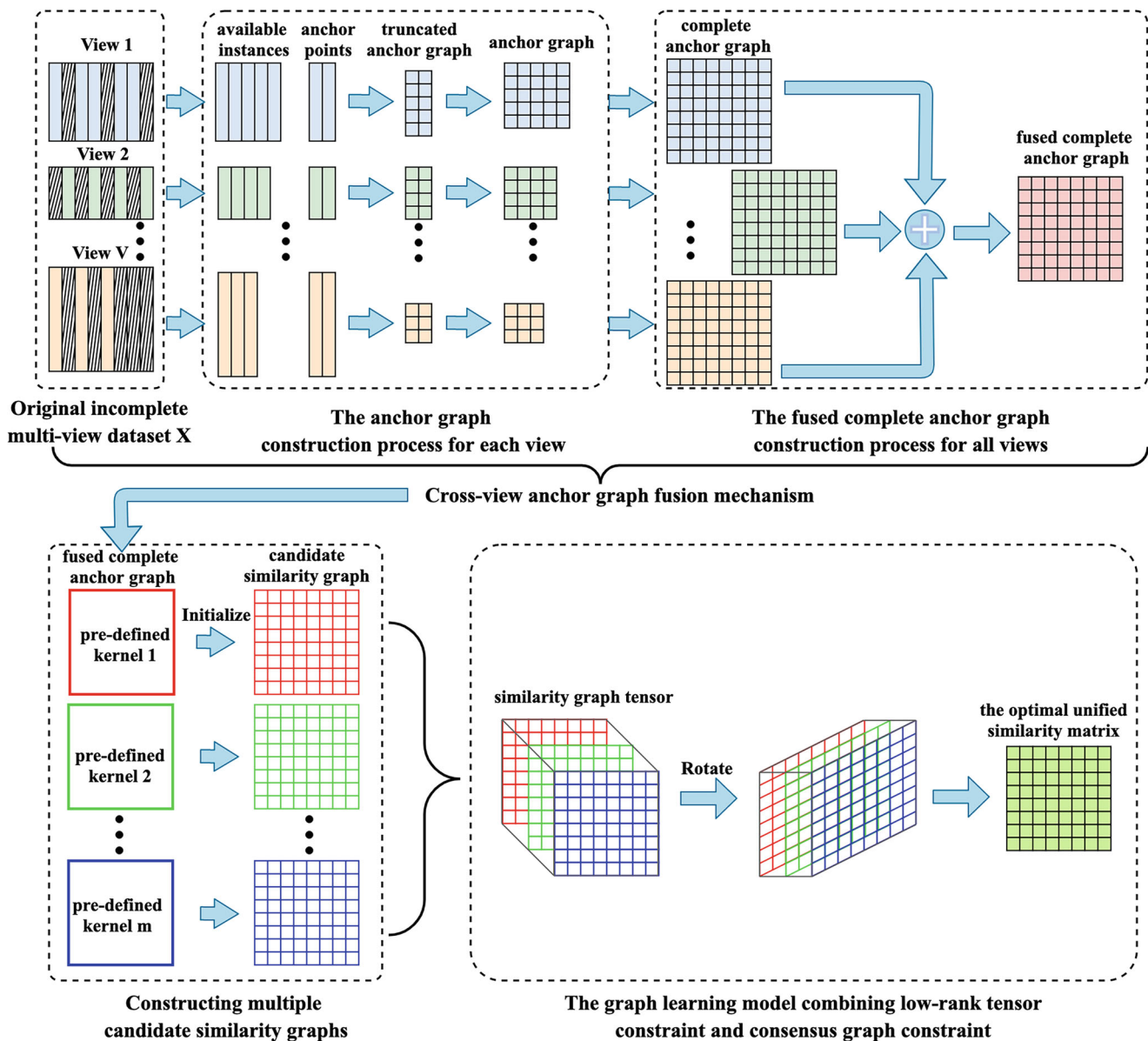


Fig. 1 General framework of our MKAGT_IMVC method. MKAGT_IMVC focuses on adaptively learning the optimal unified similarity matrix from all incomplete views

In order to ensure the anchor points cover all the clusters as much as possible, it is necessary to update the scores in an alternate manner, so that the anchor points can be obtained alternately from each cluster when selecting the anchor points. Therefore, we first standardize the score of every instance by dividing the largest score. The standardized expression is as follows:

$$s_i^{(v)} \leftarrow \frac{s_i^{(v)}}{\max(s^{(v)})}, \quad (i = 1, 2, \dots, n_v), \quad (7)$$

Consequently, the score of the currently selected instance is scaled into 1.

We pick the anchor points under the following principle, that is, select the instance with the largest score as the anchor point. Therefore, the following two key guidelines need to be satisfied to achieve the optimal anchor point selection. Firstly, we need to set the score of the selected instances to zero to avoid repeated selection. Second, By diminishing the extremely low or high scores and fairly exaggerating the medial scores, to ensure that the next anchor point and the currently selected anchor point are from as different clusters as possible [12]. To this end, the expression that updates the score can be defined as the following formula:

$$s_i^{(v)} \leftarrow s_i^{(v)}(1 - s_i^{(v)}), \quad (8)$$

This linear computation is performed l_v times to get l_v anchor points for corresponding view, where l_v is usually greater than the desired cluster number c . Finally, we can obtain the anchor point sets $\{\mathbf{U}^{(v)}\}_{v=1}^V$ from all views, where $\mathbf{U}^{(v)} = [\mathbf{u}_1^{(v)}, \mathbf{u}_2^{(v)}, \dots, \mathbf{u}_{l_v}^{(v)}] \in \mathbb{R}^{d_v \times l_v}$.

4.2 Construction of multiple kernel-based fused complete anchor graphs

Inspired of the kernel trick of multi-view clustering methods [9, 40, 41], the nonlinear samples in the original sample space can be transformed into a high-order feature space with a linear subspace structure by adopting the kernel methods, where the linear subspace structure plays critical roles for addressing the clustering problem. Specifically, for the v -th view, given the available instance matrix $\mathbf{Y}^{(v)} \in \mathbb{R}^{d_v \times n_v}$ and the anchor matrix $\mathbf{U}^{(v)} \in \mathbb{R}^{d_v \times l_v}$. Furthermore, the kernel pool composed of m kernel functions $\{K^{(p)}(\cdot)\}_{p=1}^m$ is given advance. By considering all views and these predefined kernel functions, the similarity between the i -th available instance $\mathbf{y}_i^{(v)}$ and the j -th anchor point $\mathbf{u}_j^{(v)}$ can be expressed as follows:

$$K^{(p)}(\mathbf{y}_i^{(v)}, \mathbf{u}_j^{(v)}) = (\phi^{(p)}(\mathbf{y}_i^{(v)}))^T \phi^{(p)}(\mathbf{u}_j^{(v)}), \tag{9}$$

where $K^{(p)}(\cdot)$ is the p -th predefined kernel function and $\phi^{(p)}(\cdot)$ is the corresponding mapping function without defined explicitly, which maps the instances from original space to the high-order feature space.

Moreover, under differently predefined kernel functions, by calculating the similarity between the available instances and anchor points in each view, we can obtain the element of truncated anchor graph which can be defined as the following formula [33]:

$$\bar{Z}_{ij}^{(pv)} = \frac{K^{(p)}(\mathbf{y}_i^{(v)}, \mathbf{u}_j^{(v)})}{\sum_{k \in \Omega_v} K^{(p)}(\mathbf{y}_i^{(v)}, \mathbf{u}_k^{(v)})}, \quad \forall j \in \Omega_v, \tag{10}$$

where $\Omega_v \subset \{1, 2, \dots, l_v\}$ denotes an index set of k nearest neighbors of $\mathbf{y}_i^{(v)}$ in $\mathbf{U}^{(v)}$ for the v -th view, $\bar{\mathbf{Z}}^{(pv)} \in \mathbb{R}^{n_v \times l_v}$ denotes the truncated anchor graph calculated by the p -th kernel function in the v -th view.

Once we get the truncated anchor graph $\bar{\mathbf{Z}}^{(pv)}$, according to [42], the anchor graph $\bar{\mathbf{W}}^{(pv)}$ between all available instances can be approximated as follows:

$$\bar{\mathbf{W}}^{(pv)} = \bar{\mathbf{Z}}^{(pv)} (\bar{\mathbf{\Lambda}}^{(pv)})^{-1} (\bar{\mathbf{Z}}^{(pv)})^T, \tag{11}$$

where $\bar{\mathbf{\Lambda}}^{(pv)} = \text{diag}((\bar{\mathbf{Z}}^{(pv)})^T \mathbf{1}) \in \mathbb{R}^{l_v \times l_v}$ is the diagonal matrix, $\bar{\mathbf{W}}^{(pv)} \in \mathbb{R}^{n_v \times n_v}$ is an anchor graph calculated with the available instances of the v -th view by the p -th kernel function.

The original incomplete multi-view dataset $\{\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}\}_{v=1}^V$ contains available instances and missing

instances in each view. However, the anchor graph $\bar{\mathbf{W}}^{(pv)}$ in (11) just reflects the relationships between available instances in each view. Moreover, the number of available instances in each view is inconsistent. As a result, it is impossible to directly fuse the multiple views, especially when the number of missing instances is large. To address these issues, some methods adopt imputation strategy to recover the missing graph or instances in each view [27–32], which may bring noise and make it impossible to obtain optimal results. Different from these methods, we exploit the index matrix $\mathbf{H}^{(v)} \in \mathbb{R}^{n \times n_v}$ to adjust $\bar{\mathbf{W}}^{(pv)}$ to a complete anchor graph $\mathbf{W}^{(pv)} \in \mathbb{R}^{n \times n}$ with the same number and order as the original dataset. Thus $\mathbf{W}^{(pv)}$ can be defined as follows:

$$\mathbf{W}^{(pv)} = \mathbf{H}^{(v)} \bar{\mathbf{W}}^{(pv)} (\mathbf{H}^{(v)})^T, \tag{12}$$

where $\mathbf{W}^{(pv)}$ is a complete anchor graph which connects all instances including the available instances and missing instances in the v -th view. $\mathbf{H}^{(v)} \in \mathbb{R}^{n \times n_v}$ is an index matrix which summarizes the available and missing information in the v -th view. Specifically, the elements of index matrix $\mathbf{H}^{(v)}$ can be expressed as the following formula:

$$\mathbf{H}_{ij}^{(v)} = \begin{cases} 1, & \text{if } x_i^{(v)} \text{ is the available instance } y_j^{(v)} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

For an incomplete multi-view dataset, there is a significant imbalance in the number of available instances per sample, especially when the number of views is large. For example, some samples appear in all views, and some only appear in one view. Therefore, by directly fusing all views to construct the fused complete anchor graph $\sum_{v=1}^V \mathbf{W}^{(pv)}$, there is a problem that the representation scales of all samples are different in the affine space. To this end, after obtaining the fused complete anchor graph, we divide the feature representation of each sample by the number of available instances of the corresponding sample to ensure that each sample has the same representation scale in the affine space. Then, the fused complete anchor graph $\mathbf{W}^{(p)} \in \mathbb{R}^{n \times n}$ can be defined as the following formula:

$$\mathbf{W}^{(p)} = \mathbf{A}^{-1} \left(\sum_{v=1}^V \mathbf{W}^{(pv)} \right), \tag{14}$$

where $\mathbf{A} = \text{diag}(\mathbf{h}) \in \mathbb{R}^{n \times n}$ is the diagonal matrix, $\mathbf{h} = [h_1, h_2, \dots, h_n] \in \mathbb{R}^n$ records the number of available instances for all samples. Specifically, \mathbf{h} can be expressed as follows:

$$\mathbf{h} = \sum_{v=1}^V \sum_{j=1}^{n_v} \mathbf{H}_{\cdot j}^{(v)}, \tag{15}$$

where $\mathbf{H}_{\cdot j}^{(v)}$ is the j -th column vector of index matrix $\mathbf{H}^{(v)}$ in the v -th view.

4.3 Construction of unified graph learning model

4.3.1 Low-Rank tensor based multi-view representation learning

For incomplete multi-view dataset, m fused complete anchor graphs $\{\mathbf{W}^{(p)}\}_{p=1}^m$ can be constructed through m different kernel functions. Then, we regard these fused complete anchor graphs as prior knowledge that can be incorporated into the low-rank representation based subspace clustering model in (2). However, this model cannot effectively capture and propagate the complementary information among all views to improve the clustering performance. Borrowing the idea of the t-SVD based tensor nuclear norm [43], we utilize low-rank tensor constraint to extend the matrix nuclear norm. Specifically, we first construct a similarity graph tensor $\mathcal{S} \in \mathbb{R}^{n \times n \times m}$ by stacking m candidate similarity graphs $\{\mathbf{S}^{(p)}\}_{p=1}^m$, where the tensor can be learned by imposing a variety of different constraints, e.g., low-rank tensor constraint, sparse constraint, subspace representation constraint, etc. Then, by performing the rotation operation of the similarity graph tensor, we can better explore the complementary information between these candidate similarity graphs [43]. Therefore, this learning model can be written as:

$$\begin{aligned} \min_{\{\mathbf{S}^{(p)}, \mathbf{E}^{(p)}\}_{p=1}^m} \quad & \lambda_1 \sum_{p=1}^m \|\mathbf{E}^{(p)}\|_1 + \|\mathcal{S}\|_{\otimes} \\ \text{s.t.} \quad & \mathbf{W}^{(p)} = \mathbf{W}^{(p)} \mathbf{S}^{(p)} + \mathbf{E}^{(p)}, \\ & \mathcal{S} = \varphi(\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \dots, \mathbf{S}^{(m)}), \\ & \mathbf{S}^{(p)} \geq 0, \text{diag}(\mathbf{S}^{(p)}) = 0, \mathbf{S}^{(p)} \mathbf{1} = \mathbf{1}, \end{aligned} \quad (16)$$

where $\lambda_1 > 0$ is the trade-off parameter; $\mathbf{W}^{(p)} \in \mathbb{R}^{n \times n}$ is a fused complete anchor graph calculated by the p -th kernel function, in which this matrix indicates feature representation of all samples in the affine space; $\mathbf{S}^{(p)} \in \mathbb{R}^{n \times n}$ denotes the corresponding candidate similarity graph; $\mathbf{E}^{(p)} \in \mathbb{R}^{n \times n}$ represents the corresponding reconstruction error matrix; $\varphi(\cdot)$ is an operator to stack all candidate similarity graphs $\{\mathbf{S}^{(p)}\}_{p=1}^m$ into a third-order tensor $\mathcal{S} \in \mathbb{R}^{n \times n \times m}$; $\mathbf{S}^{(p)} \geq 0$, $\text{diag}(\mathbf{S}^{(p)}) = 0$, $(\mathbf{S}^{(p)}) \mathbf{1} = \mathbf{1}$ ensures that $\mathbf{S}^{(p)}$ has the graph properties reflecting the similarity among all samples.

4.3.2 Consensus representation learning

Generally, the multi-view learning model can achieve great success compared to the single view, mainly from the following two principles [18]: (1) the complementary principle, which means that multi-view data contains complementary information, in other words, different views can describe the data in distinct perspectives; (2) the consensus principle, which means that all views in the multi-view data

have the analogous cluster structure. However, the model in (16) ignores the consensus principle, which may lead to the trivial solution. Inspired by this motivation, we develop the following consensus representation based multi-view learning model for improving the clustering performance:

$$\min_{\{\mathbf{S}^{(p)}\}_{p=1}^m} \sum_{p=1}^m \|\mathbf{S}^{(p)} - \frac{1}{m} \sum_{r=1}^m \mathbf{S}^{(r)}\|_F^2 \quad (17)$$

where $\frac{1}{m} \sum_{r=1}^m \mathbf{S}^{(r)}$ is regarded as the consensus representation by averaging all the candidate similarity graphs.

4.3.3 Objective function of MKAGT_IMVC

To sum up, for incomplete multi-view data, we jointly conduct the low-rank tensor representation based multi-view subspace learning model in (16) and the consensus representation based multi-view learning model in (17) within one unified model, which ensured that the final model not only satisfies the two principles of multi-view learning but also makes the all candidate similarity graphs $\{\mathbf{S}^{(p)}\}_{p=1}^m$ much more close to the optimal subspace structure. Hence, the objective function of MKAGT_IMVC is formulated as follows:

$$\begin{aligned} \min_{\{\mathbf{S}^{(p)}, \mathbf{E}^{(p)}\}_{p=1}^m} \quad & \left(\lambda_1 \sum_{p=1}^m \|\mathbf{E}^{(p)}\|_1 + \|\mathcal{S}\|_{\otimes} \right. \\ & \left. + \lambda_2 \sum_{p=1}^m \|\mathbf{S}^{(p)} - \frac{1}{m} \sum_{r=1}^m \mathbf{S}^{(r)}\|_F^2 \right) \\ \text{s.t.} \quad & \mathbf{W}^{(p)} = \mathbf{W}^{(p)} \mathbf{S}^{(p)} + \mathbf{E}^{(p)}, \\ & \mathbf{S}^{(p)} \geq 0, \text{diag}(\mathbf{S}^{(p)}) = 0, \mathbf{S}^{(p)} \mathbf{1} = \mathbf{1}, \\ & \mathcal{S} = \varphi(\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \dots, \mathbf{S}^{(m)}), \end{aligned} \quad (18)$$

where λ_1 and λ_2 are the trade-off parameters involved to balance the corresponding term. In this final objective function, the first term is the sparsity constraint of the reconstruction error matrix conducted l_1 -norm; the second term is the low-rank tensor constraint of the similarity graph tensor, which not only effectively captures the complementary information but also guarantees that each candidate similarity graph has the optimal low-rank subspace structure; the third term is the consensus constraint on the candidate similarity graphs, which minimizes the disagreement of the cluster structure of all candidate similarity graphs.

4.4 Optimization of proposed method

It is intractable to directly achieve the analytical solution of the final model in (18) since it is a jointly non-convex problem [44] to multiple unknown variables. Therefore, we adopt alternating direction method of multipliers (ADMM) to solve this optimization problem. Following the

principle of ADMM, we introduce several auxiliary variables $\{\mathbf{G}^{(p)}\}_{p=1}^m$ and a constraint $\mathcal{G} = \mathcal{S}$ to make the objective function separable. Then, we can reformulate (18) as follows:

$$\min_{\Theta} \begin{pmatrix} \lambda_1 \sum_{p=1}^m \|\mathbf{E}^{(p)}\|_1 + \|\mathcal{G}\|_{\otimes} \\ + \lambda_2 \sum_{p=1}^m \|\mathbf{S}^{(p)} - \frac{1}{m} \sum_{r=1}^m \mathbf{S}^{(r)}\|_F^2 \end{pmatrix} \tag{19}$$

s.t. $\mathbf{W}^{(p)} = \mathbf{W}^{(p)} \mathbf{S}^{(p)} + \mathbf{E}^{(p)}$,
 $\mathbf{S}^{(p)} \geq 0, \text{diag}(\mathbf{S}^{(p)}) = 0, \mathbf{S}^{(p)} \mathbf{1} = \mathbf{1}$,
 $\mathcal{S} = \varphi(\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \dots, \mathbf{S}^{(m)})$,
 $\mathcal{G} = \varphi(\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \dots, \mathbf{G}^{(m)})$, $\mathbf{S}^{(p)} = \mathbf{G}^{(p)}$,

where $\Theta = \{\mathbf{S}^{(p)}, \mathbf{E}^{(p)}, \mathbf{G}^{(p)}\}_{p=1}^m$ represents a collection of all variables to be optimized.

Moreover, the augmented Lagrangian function of (19) can be defined as follows:

$$L(\Theta) = \min_{\Theta} \begin{pmatrix} \lambda_1 \sum_{p=1}^m \|\mathbf{E}^{(p)}\|_1 + \|\mathcal{G}\|_{\otimes} \\ + \lambda_2 \sum_{p=1}^m \|\mathbf{S}^{(p)} - \frac{1}{m} \sum_{r=1}^m \mathbf{S}^{(r)}\|_F^2 \\ + \frac{\mu}{2} \sum_{p=1}^m \|\mathbf{W}^{(p)} - \mathbf{W}^{(p)} \mathbf{S}^{(p)} - \mathbf{E}^{(p)} + \frac{\mathbf{B}^{(p)}}{\mu}\|_F^2 \\ + \frac{\mu}{2} \sum_{p=1}^m \|\mathbf{S}^{(p)} - \mathbf{G}^{(p)} + \frac{\mathbf{C}^{(p)}}{\mu}\|_F^2 \end{pmatrix}, \tag{20}$$

where $\{\mathbf{B}^{(p)} \in \mathbb{R}^{n \times n}\}_{p=1}^m$ and $\{\mathbf{C}^{(p)} \in \mathbb{R}^{n \times n}\}_{p=1}^m$ represent the Lagrange multipliers, $\mu > 0$ is actually the penalty parameter. Then, under the ADMM framework, we can calculate each variable by iteratively solving the following corresponding sub-problems.

Step 1. Update $\{\mathbf{S}^{(p)}\}_{p=1}^m$: When the other variables are fixed and the terms unrelated to $\mathbf{S}^{(p)}$ are removed, the optimization for each candidate similarity graph $\mathbf{S}^{(p)}$ which is independent can be reformulated to the following sub-problem:

$$\min_{\mathbf{S}^{(p)}} \begin{pmatrix} \lambda_2 \|\mathbf{S}^{(p)} - \frac{1}{m} \sum_{r=1}^m \mathbf{S}^{(r)}\|_F^2 \\ + \frac{\mu}{2} \|\mathbf{W}^{(p)} - \mathbf{W}^{(p)} \mathbf{S}^{(p)} - \mathbf{E}^{(p)} + \frac{\mathbf{B}^{(p)}}{\mu}\|_F^2 \\ + \frac{\mu}{2} \|\mathbf{S}^{(p)} - \mathbf{G}^{(p)} + \frac{\mathbf{C}^{(p)}}{\mu}\|_F^2 \end{pmatrix} \tag{21}$$

s.t. $\mathbf{S}^{(p)} \geq 0, \text{diag}(\mathbf{S}^{(p)}) = 0, \mathbf{S}^{(p)} \mathbf{1} = \mathbf{1}$

By taking the partial derivative of (21) with respect to $\mathbf{S}^{(p)}$ and setting it to zero, we can get the closed-form solution as follows:

$$\widehat{\mathbf{S}}^{(p)} = \left(\frac{2\lambda_2(m-1)^2}{m^2} \mathbf{I} + \mu \mathbf{I} + \mu (\mathbf{W}^{(p)})^T \mathbf{W}^{(p)} \right)^{-1} \left(\frac{2\lambda_2(m-1)}{m^2} \sum_{r=1, r \neq p}^m \mathbf{S}^{(r)} + \mu \mathbf{G}^{(p)} - \mathbf{C}^{(p)} + \mathbf{Q} \right), \tag{22}$$

where $\mathbf{Q}^{(p)} = (\mathbf{W}^{(p)})^T (\mu (\mathbf{W}^{(p)} - \mathbf{E}^{(p)}) + \mathbf{B}^{(p)})$. Then, we can project $\widehat{\mathbf{S}}^{(p)}$ into a constrained

space. The optimal solution of $\mathbf{S}^{(p)}$ can be achieved through the following problem:

$$\mathbf{S}^{(p)} = \arg \min_{\mathbf{S}^{(p)} \geq 0, \text{diag}(\mathbf{S}^{(p)})=0, \mathbf{S}^{(p)} \mathbf{1}=\mathbf{1}} \|\mathbf{S}^{(p)} - \widehat{\mathbf{S}}^{(p)}\|_F^2. \tag{23}$$

Finally, the problem (23) can be solved by an efficiently iterative algorithm presented in [32, 45]:

$$\mathbf{S}_{(i,j)}^{(p)} = \begin{cases} 0, & i = j \\ \left(\widehat{\mathbf{S}}_{(i,j)}^{(p)} + \eta_i^{(p)} \right)_+, & i \neq j \end{cases} \tag{24}$$

where $(\cdot)_+$ is a function to preserve the non-negative elements and transform the negative elements into zero. $\eta_i^{(p)}$ is calculated as:

$$\eta_i^{(p)} = \frac{1}{n-1} - \frac{1}{n-1} \sum_{j=1, j \neq i}^n \widehat{\mathbf{S}}_{(i,j)}^{(p)}. \tag{25}$$

Step 2. Update $\{\mathbf{G}^{(p)}\}_{p=1}^m$: When the other variables are fixed and the terms unrelated to $\mathbf{G}^{(p)}$ are removed, the optimization for the auxiliary variables $\{\mathbf{G}^{(p)}\}_{p=1}^m$ can be reformulated to the following sub-problem:

$$\{\mathbf{G}^{(p)}\}_{p=1}^m = \arg \min_{\{\mathbf{G}^{(p)}\}_{p=1}^m} \|\mathcal{G}\|_{\otimes} + \frac{\mu}{2} \sum_{p=1}^m \|\mathbf{S}^{(p)} - \mathbf{G}^{(p)} + \frac{\mathbf{C}^{(p)}}{\mu}\|_F^2. \tag{26}$$

It is obvious to find that $\{\mathbf{G}^{(p)}\}_{p=1}^m, \{\mathbf{S}^{(p)}\}_{p=1}^m$ and $\{\mathbf{C}^{(p)}\}_{p=1}^m$ can be transformed into the tensor $\mathcal{G} \in \mathbb{R}^{n \times n \times m}, \mathcal{S} \in \mathbb{R}^{n \times n \times m}$ and $\mathcal{C} \in \mathbb{R}^{n \times n \times m}$, respectively. Therefore, problem (26) is equivalent to the following problem:

$$\mathcal{G} = \arg \min_{\mathcal{G}} \|\mathcal{G}\|_{\otimes} + \frac{\mu}{2} \|\mathcal{F} - \mathcal{G}\|_F^2, \tag{27}$$

where $\mathcal{F} = \mathcal{S} + \mathcal{C}/\mu$. According to [43], we first rotate \mathcal{G} from size $n \times n \times m$ to $n \times m \times n$. Through the rotation operation, the proposed model will benefit in three aspects. First of all, in each frontal slice, all samples can be represented by view-specific self-representation coefficients in the Fourier domain. Secondly, since each frontal slice considers the information among all samples and all views, the proposed model can be more effective to capture the complementary information. The third advantage is that the tensor rotation can significantly reduce the computational complexity while performing t-SVD-TNN in the Fourier domain [43].

The minimization problem (27) has the closed-form solution achieved through the tensor tubal-shrinkage operator [32, 46]:

$$\mathcal{G} = \Gamma_{\tilde{\mu}}(\mathcal{F}) = \mathcal{U} * \Gamma_{\tilde{\mu}}(\mathcal{Q}) * \mathcal{V}^T, \tag{28}$$

where $\tilde{\mu} = n/\mu$, $\mathcal{F} = \mathcal{U} * \mathcal{Q} * \mathcal{V}^T$ is achieved by t-SVD operation [47]. $\Gamma_{\tilde{\mu}}(\mathcal{Q}) = \mathcal{Q} * \mathcal{J}$, where $\mathcal{J} \in \mathbb{R}^{n \times m \times n}$ is an f-diagonal tensor whose each diagonal element in the Fourier domain is defined as $\mathcal{J}_f(i, i, k) = \max(1 - \tilde{\mu} / \mathcal{Q}_f^{(k)}(i, i), 0)$.

Step 3. Update $\{E^{(p)}\}_{p=1}^m$: When the other variables are fixed and the terms unrelated to $E^{(p)}$ are removed, the optimization for each reconstruction error matrix $E^{(p)}$ can be reformulated to the following sub-problem:

$$\min_{E^{(p)}} \lambda_1 \|E^{(p)}\|_1 + \frac{\mu}{2} \|W^{(p)} - W^{(p)}S^{(p)} - E^{(p)} + \frac{B^{(p)}}{\mu}\|_F^2. \tag{29}$$

Problem (29) has a closed-form solution as follows:

$$E^{(p)} = \Pi_{\lambda_1/\mu}(W^{(p)} - W^{(p)}S^{(p)} + \frac{B^{(p)}}{\mu}), \tag{30}$$

where Π is the shrinkage operator [48].

Step 4. Update ADMM variables: The Lagrange multipliers $\{B^{(p)}\}_{p=1}^m$, $\{C^{(p)}\}_{p=1}^m$ and penalty parameter μ can be updated as follows:

$$\begin{aligned} B^{(p)} &= B^{(p)} + \mu(W^{(p)} - W^{(p)}S^{(p)} - E^{(p)}) \\ C^{(p)} &= C^{(p)} + \mu(S^{(p)} - G^{(p)}) \\ \mu &= \min(\rho\mu, \mu_{max}), \end{aligned} \tag{31}$$

where ρ and μ_{max} are the constants involved ADMM.

In each iteration, the error of algorithm can be obtained as follows:

$$error = \max \left(\begin{aligned} &\|\psi(\varphi(M^{(p)}))\|_{\infty}, p = 1, \dots, m \\ &\|\psi(\varphi(N^{(p)}))\|_{\infty}, p = 1, \dots, m \end{aligned} \right), \tag{32}$$

where $M^{(p)} = W^{(p)} - W^{(p)}S^{(p)} - E^{(p)}$ and $N^{(p)} = S^{(p)} - G^{(p)}$, $\varphi(\cdot)$ stacks all matrices into a third-order tensor, $\psi(\cdot)$ converts the tensor to a vector. Then, the convergence condition of algorithm can be formulated as follows:

$$abs(error^{(t-1)} - error^{(t)}) < \epsilon, (t > 1) \tag{33}$$

where t denotes the t -th iteration, ϵ is a pre-defined threshold value.

The optimization procedure can be terminated when it is satisfied with the convergence condition (33). After

obtaining the optimal similarity graphs $\{S^{(p)}\}_{p=1}^m$, we use the following formula to achieve the optimal unified similarity matrix S^* .

$$S^* = \frac{1}{m} \sum_{p=1}^m (S^{(p)} + (S^{(p)})^T)/2, \tag{34}$$

Then, MKAGT_IMVC performs the spectral clustering on the optimal unified similarity matrix S^* to get the clustering results. Finally, the optimization steps of MKAGT_IMVC is described in Algorithm 1.

Algorithm 1 The proposed MKAGT_IMVC method for incomplete multi-view clustering.

Input: Incomplete Multi-View data matrices $\{X^{(v)} \in \mathbb{R}^{d_v \times n}\}_{v=1}^V$, index matrices $\{H^{(v)} \in \mathbb{R}^{n \times n_v}\}_{v=1}^V$, anchor points number $\{l_v\}_{v=1}^V$, m kernel functions $\{K^{(p)}(\cdot)\}_{p=1}^m$, cluster number c , trade-off parameters λ_1 and λ_2 .

Output: Assign cluster label to each sample.

Procedure:

- 1: Generate anchor points set $\{U^{(v)} \in \mathbb{R}^{d_v \times l_v}\}_{v=1}^V$ by (6);
 - 2: Construct the fused complete anchor graph $W^{(p)}$ with the same kernel function by (14), where setting the number of nearest neighbors $k = c$;
 - 3: Initialize each candidate similarity graph $S^{(p)} = W^{(p)}$, auxiliary variables $G^{(p)} = S^{(p)}$, reconstruction error matrix $E^{(p)} = 0$, the Lagrange multipliers $B^{(p)} = 0$, $C^{(p)} = 0$, $\forall p \in [1, 2, \dots, m]$; penalty parameter $\mu = 0.01$, $\mu_{max} = 10^6$, $\rho = 1.2$, iteration threshold value $\epsilon = 10^{-4}$;
 - 4: **while** not converged **do**
 - 5: Update $\{S^{(p)}\}_{p=1}^m$ by (24);
 - 6: Update $\{G^{(p)}\}_{p=1}^m$ by (28);
 - 7: Update $\{E^{(p)}\}_{p=1}^m$ by (30);
 - 8: Update $\{B^{(p)}\}_{p=1}^m$, $\{C^{(p)}\}_{p=1}^m$ and μ by (31);
 - 9: Check the convergence condition in (33);
 - 10: **end while**
 - 11: Obtain the optimal unified similarity matrix S^* by (34);
 - 12: Perform the spectral clustering on S^* .
-

4.5 Theoretical convergence analysis

As in the optimization process described in Section 4.4, we can observe that the objective function as formulated by (20) can be separated into three convex sub-problems. Obviously, each sub-problem can find the optimal solution with respect to the corresponding variable. This means that the loss of objective function as formulated by (20) is also monotonically decreasing in the process of alternately

updating all variables. That is, the convergence of the problem (20) can be guaranteed after some iteration steps.

4.6 Computational complexity analysis

As shown in Algorithm 1, the computational complexity of the proposed MKAGT_IMVC mainly consists of three stages, i.e., generating anchor points set, constructing multiple fused complete anchor graphs, solving the jointly objective function iteratively.

In the first stage, we adopt DAS to generate l_v anchor points in the available instances of each incomplete view, which needs to perform l_v iterations, where each iteration is required to execute a linear operation in an n_v dimensional score vector. So, the computational complexity of this stage is $\mathcal{O}(l_v n_v)$ for each view.

In the second stage, we need to construct the truncated anchor graph with (10) and the complete anchor graph with (12), where the main computational complexity is $\mathcal{O}(n_v l_v d_v)$ and $\mathcal{O}(n n_v^2 + n^2 n_v)$ respectively. Considering that $n_v \ll n$, $l_v \ll n$ and $d_v \ll n$ for large-scale incomplete multi-view data, the computational complexity of the fused complete anchor graph with (14) in this stage can be approximated as $\mathcal{O}(\sum_{v=1}^V (m n^2 n_v))$ under m predefined kernel functions.

In the third stage, the joint optimization comprises four key steps, such as solving the following variables $\{\mathbf{S}^{(p)}\}_{p=1}^m$, $\{\mathbf{G}^{(p)}\}_{p=1}^m$, $\{\mathbf{E}^{(p)}\}_{p=1}^m$, $\{\mathbf{B}^{(p)}\}_{p=1}^m$, $\{\mathbf{C}^{(p)}\}_{p=1}^m$, and μ , respectively. In step 1, for updating $\{\mathbf{S}^{(p)}\}_{p=1}^m$, the major computational complexity is the matrix inverse operation in (22), which requires $\mathcal{O}(m n^3)$. In step 2, for updating $\{\mathbf{G}^{(p)}\}_{p=1}^m$, we need to perform the following core operations, such as FFT, t-SVD, and inverse FFT. For a tensor $\mathcal{G} \in \mathbb{R}^{n \times m \times n}$, the corresponding computational complexity of these operations is about $\mathcal{O}(m n^2 \log(n))$, $\mathcal{O}(m^2 n^2)$, and $\mathcal{O}(m n^2 \log(n))$ [40]. Therefore, the total computational complexity of this step is $\mathcal{O}(m n^2 \log(n) + m^2 n^2)$. In step 3, for updating $\{\mathbf{E}^{(p)}\}_{p=1}^m$, the main computational complexity is the matrix multiplication operation in (30), whose computational complexity is $\mathcal{O}(m n^3)$. In step 4, for update ADMM variables, i.e., $\{\mathbf{B}^{(p)}\}_{p=1}^m$, $\{\mathbf{C}^{(p)}\}_{p=1}^m$, and μ , they also mainly contain the matrix addition and multiplication operations in (31), and the computational complexity is $\mathcal{O}(m n^3)$.

According to the above analysis, the total computational complexity of the three stages is $\mathcal{O}(\sum_{v=1}^V (l_v n_v + m n^2 n_v) + t(3m n^3 + m n^2 \log(n) + m^2 n^2))$, where t denotes the number of iterations of the optimization process in the third stage. After the solver converges, we conduct spectral clustering on the optimal unified similarity matrix \mathbf{S}^* , whose computational complexity is usually $\mathcal{O}(n^3)$. Moreover, considering that $n_v \ll n$, $l_v \ll n$, and

V, m, t are usually small in reality. Therefore, the main computational complexity of Algorithm 1 is actually $\mathcal{O}(n^3)$.

5 Experiments and analyses

In this section, we aim to comprehensively evaluate the effectiveness and the superiority of our method (MKAGT_IMVC) through conducting extensive experiments on various benchmark multi-view datasets. Specifically, we compare with some state-of-the-art incomplete multi-view clustering methods in terms of clustering performance and computational efficiency. In addition, we experimentally explore the anchor selection strategy, kernel selection strategy, parameter sensitivity, and convergence property to verify the stability of MKAGT_IMVC. All the experiments are implemented in the MATLAB R2016a and run on a Windows 10 PC machine with Intel(R) Xeon(R) W-2123 CPU @ 3.60-GHz and 32-GB RAM.

5.1 Experimental settings

5.1.1 Description of datasets and incomplete multi-view data construction

We adopt eight widely used real-world multi-view datasets to validate the superiority of the proposed MKAGT_IMVC. Table 2 summarizes the important statistical information of these datasets, where the first three are naturally incomplete and the last five are complete. We briefly give the descriptions of these datasets as follows.

1. **3 Sources Dataset¹ (3Sources)**: 3Sources is a naturally incomplete multi-view text dataset. It consists of 416 distinct news stories annotated with six topical areas, which were reported from three well-known online news organizations, i.e., BBC, Guardian, and Reuters. Of these stories, 53 are reported in one news organizations, 194 are reported in two news organizations, and 169 appear in all three news organizations. The sample incomplete rate of this dataset is 59.38%.
2. **BBCSport Dataset² (BBCSport)**: BBCSport is a naturally incomplete synthetic multi-view text dataset [2]. It consists of 737 sport news articles from the BBC Sport website, where each article was split into two related segments of text and categorized into five topical areas. Of these articles, 193 are reported in one segment and 544 are reported in two segments. The sample incomplete rate of this dataset is 26.19%.

¹<http://erdos.ucd.ie/datasets/3sources.html>

²<http://mlg.ucd.ie/datasets/segment.html>

Table 2 Statistics of eight benchmark multi-view datasets in our experiments

View	3Sources	BBCSport	BBC	Caltech7	Mfeat	LandUse-21	Scene-15	Hdigit
1	BBC (3560)	Segment1 (3183)	Segment1 (4659)	Gabor (48)	Pixel averages (240)	GIST (20)	GIST (20)	Mnist (784)
2	Guardian (3631)	Segment2 (3203)	Segment2 (4633)	Wavelet moments (40)	Fourier coefficients (76)	PHOG (40)	PHOG (40)	Usps (256)
3	Reuters (3068)	-	Segment3 (4665)	CENTRIST (254)	Profile correlations (216)	LBP (59)	LBP (59)	-
4	-	-	Segment4 (4684)	HOG (1984)	Zernike moments (47)	-	-	-
5	-	-	-	GIST (512)	Karhunen coefficients (64)	-	-	-
6	-	-	-	LBP (928)	Morphological (6)	-	-	-
Type	Text	Text	Text	Image	Image	Image	Image	Image
Views	3	2	4	6	6	3	3	2
Clusters	6	5	5	7	10	21	15	10
Samples	416	737	2225	1474	2000	2100	4458	10000

- BBC Dataset³ (BBC)** : BBC is a naturally incomplete synthetic multi-view text dataset [2]. It consists of 2225 stories from the BBC news website, where each story was split into four related segments of text and categorized into five topical areas. Of these stories, 213 are reported in one segment, 744 are reported in two segments, 583 are reported in three segments, and 685 are reported in four segments. The sample incomplete rate of this dataset is 69.21%.
- Caltech101-7 Image Dataset⁴ (Caltech7)**: Caltech7 is a complete multi-view image dataset [49]. It is a subset of the frequently used Caltech101 consisting of 1474 images, where each image is represented by six kinds of features [50]. Caltech7 is composed of seven object categories, i.e., Windsor-Chairs, Stop-Sign, Snoopy, Motorbikes, Garfield, Face, and Dolla-Bills.
- Multiple Features handwritten numerals Dataset⁵ (Mfeat)**: Mfeat is a complete multi-view image dataset. It consists of 2000 handwritten numerals ('0'-'9') from the UCI repository, where each handwritten numeral is described by six feature sets [51]. Mfeat is categorized into ten groups, and each group has 200 samples.
- UC Merced Land Use Dataset⁶ (LandUse-21)**: LandUse-21 is a complete multi-view image dataset. It consists of 2100 land use images selected from aerial orthoimage [52]. where each image is represented by three kinds of features i.e., GIST, PHOG, and LBP [3]. LandUse-21 contains 21 categories, and each category has 100 images.
- 15-Scene Image Dataset⁷ (Scene-15)**: Scene-15 is a complete multi-view image dataset. It consists of 4485 images with indoor and outdoor environments [53], where each image is described by three kinds of features i.e., GIST, PHOG, and LBP [3]. Scene-15 contains 15 natural scene categories, each of which has 200 to 400 images.
- Handwritten digit 2 sources Dataset⁸ (Hdigit)**: Hdigit is a complete multi-view image dataset. It consists of 10000 handwritten digits ('0'-'9') collected from two sources, i.e., MNIST and USPS [11]. Hdigit is categorized into ten groups, and each group has 1000 samples.

Incomplete Multi-view Data Construction: In our experiments, since the 3Sources, BBCSport, and BBC

³<http://mlg.ucd.ie/datasets/segment.html>

⁴http://www.vision.caltech.edu/Image_Datasets/Caltech101/

⁵<http://archive.ics.uci.edu/ml/datasets/Multiple+Features>

⁶<http://weegee.vision.ucmerced.edu/datasets/landuse.html>

⁷https://figshare.com/articles/dataset/15-Scene_Image_Dataset/7007177

⁸<https://cs.nyu.edu/~roweis/data.html>

datasets are naturally incomplete multi-view datasets, we directly conduct various comparative experiments on these three incomplete multi-view datasets. For the other five complete multi-view datasets, we first randomly select $\xi\%$ samples in each dataset as incomplete samples which suffer from missing instances in some views. Then, under such condition that each sample can observe a specific value in at least one view, we randomly generate a binary vector $\mathbf{b} = (b_1, b_2, \dots, b_V)$ for each incomplete sample, where if $b_i = 0$, we set the instance feature in the i -th view of the sample to 0. In this way, we construct five incomplete multi-view datasets, each with different sample incomplete rates $\xi\%$ ($\xi = 70, 90, 100$), to fully assess the effectiveness of the proposed method, where $\xi = 100$ means that all samples are incomplete.

5.1.2 Compared Methods

To comprehensively evaluate the clustering performance of the proposed MKAGT-IMVC, we compare it with the following some state-of-the-art incomplete multi-view clustering methods.

1. **Best Single View (BSV)** [54]: This method populates missing instances with the mean value of the available instances on each view. Then, BSV performs standard spectral clustering [55] on all views independently and reports the best result.
2. **Concat** [54]: This method performs standard spectral clustering on the long dimension features for the final clustering result, where the long dimension features are generated through concatenating the features of all views.
3. **DAIMC** [19]: This method learns a unified feature matrix from all views through combining weighted semi-nonnegative Matrix Factorization and $L_{2,1}$ -Norm regularized regression. Then, DAIMC performs k -means on the unified feature matrix to get the final clustering result.
4. **PIC** [30]: This method converts the instance-value missing into similarity-value missing on each view. Then, PIC learns a consistent Laplacian matrix by adopting the spectral perturbation theory to balance all views. Lastly, PIC performs standard spectral clustering on the consistent Laplacian matrix for the final clustering result.
5. **MKKM-IK-MKC** [27]: This method develops a unified learning framework, which jointly optimizes the multiple kernel imputation and consistent representation learning to achieve better clustering. Then, MKKM-IK-MKC performs k -means on the consistent representation to get the final clustering result.
6. **IMSC_AGL** [15]: This method designs a joint optimization framework, which integrates the optimal similarity graph construction for each view and consistent representation learning for all views. Then, IMSC_AGL performs k -means on the consistent representation to get the final clustering result.
7. **AGC_IMC** [31]: This method develops a unified learning framework, which combines incomplete graphs recovering and consistent representation learning to obtain the optimal consistent representation. Then, AGC_IMC performs k -means on the optimal consistent representation to get the final clustering result.
8. **GIMC_FLSD** [20]: This method develops a matrix factorization model, which integrates individual representation learning for each view and consensus representation learning for all views. Meanwhile, GIMC_FLSD imposes adaptively weighted learning on the model to obtain the optimal consistent representation. Lastly, GIMC_FLSD performs k -means on the optimal consistent representation to get the final clustering result.
9. **APMC** [33]: This method uses the common samples showing all views as anchors to construct the fused similarity matrix. Then, APMC performs standard spectral clustering on the fused similarity matrix for the final clustering result.
10. **IMVTSC-MVI** [32]: This method develops a low-rank representation framework to well recover missing instances appearing in the certain view and learn the optimal similarity graph from all views simultaneously, where the framework is imposed multiple constraints. Then, IMVTSC-MVI performs standard spectral clustering on the optimal similarity graph for the final clustering result.

For all comparison methods, we get the available source codes from the corresponding author's homepage. To make the experiments more fair and conclusive, we run the source codes of comparison methods either using the default parameter values or tuning them as suggested according to the original paper to obtain the best performances. For example, the number of nearest neighbors in the relevant comparison methods is either the default value of the source codes or 10% of the total sample of each dataset.

For the proposed MKAGT-IMVC, we first adopt three commonly used kernel function to construct the view-wise truncated anchor graphs, which are defined as follows:

1. Radial Basis Function (RBF) Kernel:

$$K(\mathbf{y}, \mathbf{u}) = \exp\left(-\gamma\|\mathbf{y} - \mathbf{u}\|^2\right), \quad (35)$$

2. Cosine Similarity Kernel [56]:

$$K(\mathbf{y}, \mathbf{u}) = 1 - \frac{\mathbf{y}^T \mathbf{u}}{\|\mathbf{y}\| \|\mathbf{u}\|}, \quad (36)$$

3. Radial Basis Function (RBF) Chi-Square Kernel [57]:

$$K(\mathbf{y}, \mathbf{u}) = \exp\left(-\gamma\left(\frac{1}{2}\sum_{i=1}^d \frac{(y_i - u_i)^2}{(y_i + u_i)}\right)\right). \quad (37)$$

Without loss of generality, the parameter γ is set to 1 in (35) of the RBF Kernel of and (37) of the RBF Chi-Square Kernel. Furthermore, We empirically set the number of nearest neighbors $k = c$ for creating the view-wise truncated anchor graphs, where the c is the number of clusters of the raw data. For the 3Sources dataset, we set the parameters of the proposed MKAGT_IMVC to $\lambda_1 = 0.001$ and $\lambda_2 = 0.1$ in joint learning model. Meanwhile, for other datasets with different sample incomplete rates, we set these two parameters as $\lambda_1 = 0.001$ and $\lambda_2 = 1$. According to (43), we empirically set the proportion of anchors of 3Sources, BBCSport, BBC, Caltech7, Mfeat, LandUse-21, Scene-15, and Hdigit datasets to entire available instances of the corresponding dataset as $\eta = \{0.58, 0.8, 0.1, 1, 0.4, 1, 0.5, 0.2\}$, separately, where these multi-view datasets have different sample incomplete rates.

Since the k -means is sensitive to the setting of the initial centroid, we perform k -means 20 times for each experiment to eliminate the influence of random initialization, where the maximum number of iterations of k -means is set to 1000. Meanwhile, we randomly generate five different incomplete patterns with the same sample incomplete rate in an above manner and perform all comparison methods on such five patterns to record the average values with standard deviations. Since the use of different data preprocessing schemes and parameter ranges, some results may deviate from the corresponding published information.

5.1.3 Evaluation metrics

Without loss of generality, we adopt three commonly used evaluation metrics to fully assess the performance of different methods as shown in [10]: Clustering Accuracy (ACC), Normalized Mutual Information (NMI), and Purity. These evaluation metrics are calculated in a predetermined formula with clustering labels of all samples and the ground-truth labels of the dataset. For the three metrics, a higher value indicates a better performance. Specifically, the three metrics are defined as follows.

The first metric is ACC which measures clustering accuracy.

$$\text{ACC} = \frac{\sum_{i=1}^n \delta(l_i, \text{map}(c_i))}{n} \quad (38)$$

where l_i and c_i denote the provided ground-truth label and the obtained cluster label of the i -th sample, separately; n is the number of samples; δ is the Dirac delta function

$$\delta(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise.} \end{cases} \quad (39)$$

and $\text{map}(c_i)$ is the optimal permutation mapping function which arranges the cluster label c_i to match the ground-truth label l_i via using the Kuhn-Munkres algorithm.

The second evaluation metric is NMI which assess the quality of the obtained clusters.

$$\text{NMI}(\mathbf{C}, \mathbf{L}) = \frac{MI(\mathbf{C}, \mathbf{L})}{\sqrt{E(\mathbf{C})E(\mathbf{L})}} \quad (40)$$

where the set $\mathbf{C} = \{c_1, c_2, \dots, c_c\}$ and the set $\mathbf{L} = \{l_1, l_2, \dots, l_c\}$ denote the obtained clusters using clustering methods and the provided ground-truth clusters, separately; the constant c is the number of clusters of the raw data; $MI(\mathbf{C}, \mathbf{L})$ denotes the mutual information between \mathbf{C} and \mathbf{L} ; $E(\mathbf{C})$ and $E(\mathbf{L})$ denote the entropy of \mathbf{C} and \mathbf{L} , separately.

Let n_i denotes the number of samples in the obtained cluster c_i ($1 \leq i \leq c$) using clustering methods; and n_j denotes the number of samples belonging to the provided ground-truth cluster l_j ($1 \leq j \leq c$). Then, NMI is rewritten as

$$\text{NMI} = \frac{\sum_{i=1}^c \sum_{j=1}^c n_{i,j} \log\left(\frac{n_{i,j}}{n_i n_j}\right)}{\sqrt{(\sum_{i=1}^c n_i \log \frac{n_i}{n})(\sum_{j=1}^c n_j \log \frac{n_j}{n})}} \quad (41)$$

where $n_{i,j}$ denotes the number of samples in the common parts of c_i and l_j .

The third evaluation metric is Purity which calculates the percentage of correct labels to evaluate the effectiveness of the clustering method.

$$\text{Purity} = \frac{1}{n} \sum_{i=1}^c \max_{1 \leq j \leq c} |c_i \cap l_j| \quad (42)$$

5.2 Clustering results and analysis

To evaluate the effectiveness and superiority of the proposed MKAGT_IMVC, we adopt a variety of incomplete multi-view clustering methods to conduct a series of comparative experiments on eight incomplete multi-view datasets, including three naturally incomplete datasets and five randomly constructed incomplete datasets. Tables 3 and 4 present the clustering results of all comparison methods on multi-view datasets with different sample incomplete rates mentioned above in terms of ACC, NMI, Purity, and CPU running time, respectively. In addition, the CPU running time measurement refers to starting from the original features of the input multi-view dataset and ending with at output of the cluster label. From these experimental results, we can make the following observations:

1. BSV and Concat perform worse than the other incomplete multi-view clustering approaches in terms

Table 3 The clustering performance (mean% \pm std%) and CPU running time (sec.) (mean \pm std) of the proposed MKAGT_IMVC method are compared with the comparison methods on the 3Sources, BBCSport, and BBC datasets, where the three datasets are naturally incomplete multi-view datasets. Bold numbers denote the best results. '-' indicates that the method cannot be run on the corresponding dataset

Dataset	3Sources			BBCSport			BBC					
	Method \ Metric	ACC	NMI	Purity	CPU time	ACC	NMI	Purity	CPU time	ACC	NMI	Purity
BSV	24.76 \pm 0.00	2.36 \pm 0.00	28.12 \pm 0.00	10.70 \pm 0.00	36.91 \pm 0.00	9.49 \pm 0.00	40.84 \pm 0.00	22.80 \pm 0.00	38.97 \pm 0.00	19.07 \pm 0.00	42.88 \pm 0.00	199.86 \pm 0.00
Concat	82.69 \pm 0.00	66.89 \pm 0.00	82.69 \pm 0.00	12.13 \pm 0.00	89.82 \pm 0.00	73.97 \pm 0.00	89.82 \pm 0.00	23.45 \pm 0.00	90.79 \pm 0.00	75.48 \pm 0.00	90.79 \pm 0.00	640.27 \pm 0.00
AGC_IMC	56.25 \pm 0.00	49.47 \pm 0.00	58.89 \pm 0.00	9.23 \pm 0.00	72.73 \pm 0.00	72.03 \pm 0.00	75.98 \pm 0.00	15.71 \pm 0.00	90.61 \pm 0.00	75.57 \pm 0.00	90.61 \pm 0.00	416.01 \pm 0.00
MKMM-IK-MKC	72.84 \pm 0.00	61.89 \pm 0.00	72.84 \pm 0.00	0.64 \pm 0.00	87.52 \pm 0.00	69.49 \pm 0.00	87.52 \pm 0.00	1.57 \pm 0.00	90.56 \pm 0.00	75.03 \pm 0.00	90.56 \pm 0.00	16.26 \pm 0.00
DAIMC	73.32 \pm 0.00	59.63 \pm 0.00	73.32 \pm 0.00	386.53 \pm 0.00	85.48 \pm 0.00	65.62 \pm 0.00	85.48 \pm 0.00	209.68 \pm 0.00	72.31 \pm 0.00	53.32 \pm 0.00	72.31 \pm 0.00	1368.39 \pm 0.00
IMVTSC-MVI	81.97 \pm 0.00	70.33 \pm 0.00	81.97 \pm 0.00	28.71 \pm 0.00	84.26 \pm 0.00	73.01 \pm 0.00	84.26 \pm 0.00	43.24 \pm 0.00	94.52 \pm 0.00	83.24 \pm 0.00	94.52 \pm 0.00	78.56 \pm 0.00
GIMC_FLSD	84.86 \pm 0.00	69.51 \pm 0.00	84.86 \pm 0.00	27.35 \pm 0.00	92.27 \pm 0.00	78.82 \pm 0.00	92.27 \pm 0.00	32.22 \pm 0.00	81.80 \pm 0.00	64.44 \pm 0.00	81.80 \pm 0.00	166.34 \pm 0.00
AFMC	85.34 \pm 0.00	71.09 \pm 0.00	85.34 \pm 0.00	0.75 \pm 0.00	87.79 \pm 0.00	70.70 \pm 0.00	87.79 \pm 0.00	0.56 \pm 0.00	-	-	-	-
PIC	87.98 \pm 0.00	73.96 \pm 0.00	87.98 \pm 0.00	1.77 \pm 0.00	93.89 \pm 0.00	83.78 \pm 0.00	93.89 \pm 0.00	1.23 \pm 0.00	93.03 \pm 0.00	80.22 \pm 0.00	93.03 \pm 0.00	68.04 \pm 0.00
IMSC_AGL	90.87 \pm 0.00	78.35 \pm 0.00	90.87 \pm 0.00	18.24 \pm 0.00	94.03 \pm 0.00	83.19 \pm 0.00	94.03 \pm 0.00	45.10 \pm 0.00	93.84 \pm 0.00	81.79 \pm 0.00	93.84 \pm 0.00	1501.64 \pm 0.00
MKAGT_IMVC	89.42 \pm 0.00	75.23 \pm 0.00	89.42 \pm 0.00	10.71 \pm 0.00	99.73 \pm 0.00	99.01 \pm 0.00	99.73 \pm 0.00	34.33 \pm 0.00	99.01 \pm 0.00	96.32 \pm 0.00	99.01 \pm 0.00	193.59 \pm 0.00

of ACC, NMI, Purity in most incomplete situations of these datasets. BSV simply populates missing instances with the same mean values on corresponding views, which will cause the missing instances in the corresponding view to be split into the same cluster. Meanwhile, by directly concatenating the features of all views into one long dimension feature, Concat may generate too much redundant information, which is the main reason for poor clustering results. Thus, the experimental results demonstrate that it is noteworthy to explore the relationships between multiple views to improve the clustering performance.

2. We can see that the proposed MKAGT_IMVC method can handle arbitrary incomplete multi-view datasets, such as the three naturally incomplete multi-view datasets in Table 3, and the five randomly constructed incomplete multi-view datasets in Table 4, where each dataset has three sample incomplete rates 70%, 90%, and 100%, respectively. Obviously, on most above incomplete multi-view datasets, we can observe that MKAGT_IMVC significantly outperforms all the comparison methods in terms of ACC, NMI, Purity. For example, from Tables 3 and 4, MKAGT_IMVC obtains the three clustering performance metrics on the BBCSport, BBC, and Mfeat datasets, respectively, all of which are very close to 100%. From Table 4, we can observe that the experimental results of the proposed method and all the compared methods are relatively low in terms of ACC, NMI, and Purity on the LandUse-21 and Scene-15 datasets. The main reason is that LandUse-21 and Scene-15 are more complex than other datasets, especially their number of clusters is 21 and 15, respectively, which is much more than other datasets. This brings great challenges to the learning process of all methods in fitting the spatial distribution of the data. Nevertheless, the proposed MKAGT_IMVC method exceeds the second best method with 70%/90%/100% sample incomplete rates in terms of ACC, NMI, and Purity as about 14%/22%/17%, 18%/27%/21%, and 13%/23%/16% on the LandUse-21 dataset, 31%/32%/30%, 36%/38%/36%, and 25%/27%/24% on the Scene-15 dataset. On the Hdigit dataset, the proposed MKAGT_IMVC method exceeds the second best method with 90%/100% sample incomplete rates in terms of ACC, NMI, and Purity as about 15%/8%/13% and 50%/33%/48%. The main reason is that the proposed MKAGT_IMVC method can learn an optimal unified similarity matrix from multiple views by simultaneously considering the three key steps: (1) generate significantly representative anchor points; (2) construct multiple fused complete anchor graphs with various kernel functions; (3) develop an effective sub-space representation based graph learning model which

Table 4 The clustering performance (mean% ± std%) and CPU running time (sec.) (mean ± std) of the proposed MKAGT_IMVC method are compared with the comparison methods on the Caltech7, Mfeat, LandUse-21, Scene-15, and Hdigit datasets, where each dataset has three sample incomplete rates, 70%, 90%, and 100% respectively. Bold numbers denote the best results. '-' indicates that the method cannot be run on the corresponding dataset

Dataset	Method \ Metric	70%					90%					100%				
		ACC	NMI	Purity	CPU time	ACC	NMI	Purity	CPU time	ACC	NMI	Purity	CPU time			
Caltech7	BSV	37.58 ± 3.54	9.31 ± 4.07	61.66 ± 4.31	43.44 ± 4.23	37.06 ± 6.62	16.19 ± 9.81	63.92 ± 6.00	36.45 ± 9.44	41.15 ± 7.00	21.66 ± 8.77	63.47 ± 5.67	25.33 ± 12.20			
	Concat	32.39 ± 10.00	8.49 ± 5.05	60.62 ± 5.00	82.83 ± 16.79	33.81 ± 0.84	15.58 ± 0.78	67.83 ± 0.36	65.29 ± 0.21	32.62 ± 4.50	13.66 ± 0.95	66.45 ± 0.65	60.46 ± 0.39			
	IMSC-AGL	36.76 ± 1.01	44.46 ± 1.11	81.11 ± 1.00	607.38 ± 54.20	36.81 ± 2.36	43.68 ± 3.80	80.76 ± 2.39	515.17 ± 0.81	37.07 ± 3.22	42.48 ± 1.24	81.21 ± 0.59	478.35 ± 7.93			
	GIMC-FLSD	36.55 ± 0.93	33.84 ± 1.24	77.88 ± 1.96	52.77 ± 0.64	37.48 ± 1.42	32.70 ± 1.58	77.50 ± 2.52	41.86 ± 1.67	35.97 ± 1.75	27.63 ± 3.19	74.42 ± 1.28	38.44 ± 0.35			
	DAIMC	41.85 ± 3.36	47.62 ± 3.61	83.35 ± 1.12	100.62 ± 1.59	38.63 ± 4.03	45.87 ± 3.00	82.92 ± 0.74	104.33 ± 3.42	40.16 ± 6.17	44.13 ± 1.62	82.04 ± 0.91	106.78 ± 2.16			
	MKMM-IK-MKC	37.22 ± 1.12	34.61 ± 2.02	75.92 ± 0.91	48.97 ± 1.57	38.66 ± 1.12	33.99 ± 2.11	75.48 ± 0.76	54.08 ± 1.61	37.67 ± 1.53	34.29 ± 1.89	75.51 ± 0.82	52.34 ± 0.21			
	PIC	48.33 ± 5.07	49.13 ± 2.46	83.68 ± 0.92	45.65 ± 0.56	42.67 ± 7.46	47.63 ± 2.02	82.99 ± 0.85	48.49 ± 0.50	40.79 ± 6.54	45.92 ± 1.57	81.38 ± 0.84	46.22 ± 0.48			
	AGC-IMC	63.79 ± 3.98	56.97 ± 3.66	85.94 ± 1.34	407.11 ± 11.89	59.84 ± 2.96	53.80 ± 2.32	84.08 ± 0.79	356.49 ± 19.57	65.47 ± 1.68	54.81 ± 4.39	85.39 ± 1.41	351.62 ± 13.56			
	IMVTSC-MVI	63.55 ± 4.03	64.01 ± 3.31	89.01 ± 1.54	214.10 ± 0.71	61.95 ± 3.91	62.32 ± 2.08	88.52 ± 1.08	248.54 ± 3.87	61.87 ± 5.38	65.19 ± 3.60	90.23 ± 2.18	244.89 ± 2.83			
	MKAGT_IMVC	61.03 ± 2.65	71.80 ± 0.85	90.96 ± 0.19	119.77 ± 16.33	59.73 ± 3.32	69.81 ± 2.86	90.09 ± 1.35	111.43 ± 13.15	60.88 ± 1.97	70.81 ± 0.92	90.84 ± 0.05	96.71 ± 17.41			
Mfeat	BSV	25.84 ± 8.31	16.11 ± 9.91	27.07 ± 8.37	33.87 ± 14.73	29.52 ± 17.39	23.02 ± 20.78	31.29 ± 17.92	33.73 ± 13.40	29.59 ± 7.06	21.53 ± 7.35	30.59 ± 6.84	30.60 ± 14.30			
	Concat	16.51 ± 1.33	4.96 ± 2.11	17.42 ± 1.65	61.12 ± 7.31	18.97 ± 2.88	10.60 ± 6.14	19.54 ± 2.86	33.29 ± 15.66	18.24 ± 1.40	8.91 ± 3.23	18.81 ± 1.49	26.61 ± 12.11			
	GIMC-FLSD	25.70 ± 2.11	30.56 ± 1.33	27.22 ± 0.72	27.40 ± 2.24	26.44 ± 1.47	29.53 ± 4.14	28.43 ± 2.05	24.63 ± 0.53	46.49 ± 2.24	43.64 ± 1.48	48.85 ± 1.78	106.46 ± 3.68			
	MKMM-IK-MKC	74.01 ± 10.91	65.12 ± 8.94	74.10 ± 10.81	41.85 ± 1.41	67.24 ± 5.04	58.91 ± 3.38	67.50 ± 4.86	45.90 ± 1.32	65.10 ± 5.34	57.38 ± 2.18	66.30 ± 4.46	48.35 ± 2.89			
	DAIMC	84.25 ± 2.05	88.04 ± 1.21	86.91 ± 0.51	711.19 ± 83.96	83.55 ± 4.40	87.24 ± 2.37	84.78 ± 4.35	551.99 ± 33.27	83.29 ± 4.96	85.75 ± 1.86	84.01 ± 3.92	604.08 ± 14.71			
	AGC-IMC	84.37 ± 1.25	76.61 ± 1.00	84.37 ± 1.25	110.98 ± 0.56	85.59 ± 2.05	75.91 ± 2.21	85.59 ± 2.05	106.55 ± 6.08	85.33 ± 1.65	74.50 ± 1.16	85.33 ± 1.65	111.63 ± 0.50			
	PIC	87.46 ± 1.85	77.94 ± 1.53	87.46 ± 1.85	1429.19 ± 18.93	85.71 ± 1.72	75.62 ± 1.35	85.71 ± 1.72	1299.75 ± 22.08	83.01 ± 2.41	71.75 ± 2.40	83.01 ± 2.41	1252.94 ± 91.80			
	IMSC-AGL	98.58 ± 0.14	96.61 ± 0.30	98.58 ± 0.14	312.91 ± 0.73	99.23 ± 0.12	98.05 ± 0.24	99.23 ± 0.12	359.86 ± 1.67	99.61 ± 0.13	98.99 ± 0.34	99.61 ± 0.13	342.33 ± 1.65			
	IMVTSC-MVI	99.96 ± 0.02	99.89 ± 0.06	99.96 ± 0.02	117.04 ± 5.41	99.94 ± 0.07	99.84 ± 0.18	99.94 ± 0.07	114.08 ± 5.85	99.92 ± 0.03	99.78 ± 0.07	99.92 ± 0.03	112.39 ± 5.80			
	MKAGT_IMVC	13.81 ± 3.02	13.01 ± 2.91	14.85 ± 3.29	89.80 ± 19.81	15.87 ± 3.11	15.63 ± 4.43	17.26 ± 3.52	69.87 ± 2.00	17.68 ± 1.76	18.33 ± 2.73	18.82 ± 2.01	72.02 ± 5.98			
LandUse-21	BSV	9.29 ± 0.57	4.32 ± 0.42	9.73 ± 0.54	92.18 ± 8.63	9.06 ± 0.58	3.91 ± 0.58	9.44 ± 0.66	66.00 ± 1.68	9.19 ± 0.39	4.00 ± 0.51	9.69 ± 0.40	67.66 ± 3.08			
	Concat	16.41 ± 1.25	15.40 ± 1.76	18.15 ± 1.87	25.76 ± 0.17	13.62 ± 1.08	12.32 ± 2.08	15.01 ± 1.48	25.46 ± 0.10	13.26 ± 0.84	11.41 ± 1.38	14.82 ± 0.90	25.55 ± 0.07			
	DAIMC	18.30 ± 1.06	28.11 ± 1.38	22.30 ± 0.98	262.86 ± 2.35	16.50 ± 1.90	25.39 ± 1.98	19.96 ± 1.54	252.45 ± 5.22	16.76 ± 2.95	24.79 ± 2.71	19.70 ± 3.13	267.55 ± 5.17			
	GIMC-FLSD	18.53 ± 0.48	22.41 ± 0.89	22.38 ± 0.38	34.29 ± 0.69	18.00 ± 0.90	21.74 ± 0.62	21.70 ± 0.55	31.69 ± 0.23	18.12 ± 1.27	21.28 ± 0.92	21.44 ± 1.01	29.17 ± 0.10			
	APMC	20.41 ± 0.85	22.98 ± 0.59	23.48 ± 0.69	5.63 ± 0.17	18.43 ± 0.49	18.88 ± 0.52	20.31 ± 0.68	5.52 ± 0.50	16.42 ± 0.27	16.74 ± 0.92	18.25 ± 0.72	6.62 ± 0.54			
	MKMM-IK-MKC	19.28 ± 0.82	20.31 ± 1.12	21.50 ± 0.83	126.69 ± 0.81	18.48 ± 1.06	19.35 ± 0.95	21.03 ± 1.04	132.39 ± 0.33	18.24 ± 0.90	18.45 ± 0.69	20.94 ± 0.91	64.64 ± 0.28			
	PIC	21.63 ± 0.46	23.38 ± 0.59	24.18 ± 0.43	32.26 ± 0.17	18.53 ± 0.54	19.34 ± 0.63	20.90 ± 0.48	29.63 ± 0.28	17.30 ± 0.87	18.00 ± 1.04	19.26 ± 0.87	32.39 ± 0.26			
	IMSC-AGL	20.96 ± 0.37	21.56 ± 0.42	22.81 ± 0.52	819.70 ± 4.26	20.18 ± 0.80	20.02 ± 0.83	22.10 ± 0.83	765.25 ± 6.63	18.53 ± 0.59	18.04 ± 0.69	20.66 ± 0.73	765.88 ± 61.34			
	IMVTSC-MVI	24.22 ± 0.69	24.64 ± 0.38	25.81 ± 0.91	187.94 ± 0.53	20.68 ± 0.43	21.02 ± 0.53	22.28 ± 0.64	220.00 ± 6.40	19.68 ± 0.62	19.71 ± 0.98	21.59 ± 0.70	210.48 ± 6.90			
	MKAGT_IMVC	38.11 ± 4.10	46.82 ± 5.24	42.63 ± 4.66	160.73 ± 18.99	38.45 ± 1.40	48.13 ± 0.98	43.52 ± 1.18	144.62 ± 8.53	32.78 ± 4.77	42.14 ± 6.12	37.42 ± 4.41	151.68 ± 15.41			

Table 4 (continued)

Incomplete rate		70%					90%					100%					
Dataset	Method \ Metric	ACC	NMI	Purity	CPU time	ACC	NMI	Purity	CPU time	ACC	NMI	Purity	CPU time	ACC	NMI	Purity	CPU time
Scene-15	BSV	12.57 ± 0.71	6.23 ± 0.71	14.19 ± 0.94	153.19 ± 9.77	13.26 ± 1.22	6.10 ± 1.63	14.97 ± 1.67	140.79 ± 8.69	13.31 ± 1.53	7.56 ± 2.00	15.24 ± 1.50	131.72 ± 9.65	13.06 ± 2.05	8.17 ± 4.92	15.65 ± 3.10	71.31 ± 52.41
	Concat	10.47 ± 1.14	2.22 ± 0.85	11.63 ± 1.01	169.62 ± 43.26	12.70 ± 1.62	7.10 ± 4.27	14.64 ± 2.36	70.17 ± 49.04	19.22 ± 2.15	13.89 ± 2.46	21.73 ± 2.20	98.99 ± 0.74	20.55 ± 1.37	19.45 ± 0.42	23.53 ± 0.43	31.29 ± 1.60
	DAIMC	23.84 ± 1.85	18.94 ± 0.94	26.31 ± 1.78	100.39 ± 0.40	18.37 ± 1.71	13.25 ± 1.42	21.01 ± 1.94	98.31 ± 2.19	23.81 ± 1.17	22.89 ± 1.31	28.27 ± 1.50	261.59 ± 3.36	23.15 ± 0.71	22.40 ± 0.49	27.66 ± 0.51	274.75 ± 0.57
	APMC	30.47 ± 1.15	26.81 ± 0.70	32.71 ± 1.21	35.83 ± 0.35	22.75 ± 0.46	22.70 ± 0.67	26.30 ± 0.60	31.20 ± 2.53	24.63 ± 0.99	21.52 ± 0.88	27.10 ± 1.00	1155.38 ± 4.57	23.79 ± 1.46	21.41 ± 0.92	25.96 ± 1.00	1164.07 ± 4.53
	PIC	29.68 ± 1.47	26.38 ± 0.57	33.94 ± 0.65	272.80 ± 1.20	26.98 ± 0.74	23.46 ± 0.44	30.26 ± 0.52	882.48 ± 1.09	24.33 ± 0.40	21.18 ± 0.91	28.41 ± 0.98	434.00 ± 7.20	28.69 ± 0.41	27.17 ± 0.88	31.03 ± 0.80	189.23 ± 7.81
	IMVTSC-MVI	28.00 ± 1.74	23.32 ± 1.27	30.09 ± 1.04	1028.47 ± 3.71	30.55 ± 0.23	28.10 ± 0.38	32.73 ± 0.58	221.51 ± 1.57	27.70 ± 0.93	28.04 ± 0.93	30.85 ± 0.46	1590.61 ± 134.75	29.84 ± 1.16	26.61 ± 0.76	33.19 ± 1.26	6975.45 ± 82.60
	MKKM-1K-MKC	28.83 ± 1.14	25.49 ± 0.40	31.04 ± 1.11	283.11 ± 64.64	30.82 ± 1.05	30.28 ± 0.82	33.36 ± 0.75	1351.28 ± 7.40	68.61 ± 4.13	66.71 ± 3.47	71.28 ± 3.07	820.27 ± 101.86	54.75 ± 14.29	53.43 ± 15.63	57.60 ± 14.41	1121.39 ± 216.73
	GIMC_FLSD	33.17 ± 1.42	30.18 ± 0.52	35.29 ± 0.79	238.37 ± 1.29	31.14 ± 6.83	23.80 ± 9.57	31.87 ± 6.72	1514.36 ± 33.38	15.43 ± 1.30	5.89 ± 1.58	17.56 ± 1.85	1024.02 ± 35.71	15.89 ± 1.79	7.23 ± 3.13	18.08 ± 1.58	1033.48 ± 36.84
	AGC_IMC	29.38 ± 1.33	31.43 ± 0.87	34.10 ± 1.41	1491.54 ± 6.65	32.82 ± 1.59	27.42 ± 0.67	34.85 ± 0.88	78.33 ± 0.31	39.47 ± 0.61	35.47 ± 0.32	40.97 ± 0.54	5870.18 ± 163.87	37.95 ± 1.87	36.07 ± 1.54	40.52 ± 0.71	5817.07 ± 36.32
	IMSC_AGL	34.01 ± 1.18	30.95 ± 0.34	38.08 ± 0.87	8356.39 ± 162.12	34.75 ± 0.66	28.50 ± 0.51	37.33 ± 0.65	461.30 ± 56.28	40.04 ± 3.17	30.72 ± 2.70	44.31 ± 2.15	51106.35 ± 776.00	30.02 ± 3.15	21.33 ± 3.13	32.19 ± 3.21	45567.64 ± 197.34
Hdigit	MKAGT_IMVC	65.31 ± 12.55	62.92 ± 12.94	68.07 ± 12.32	1135.17 ± 242.71	38.10 ± 0.31	34.32 ± 0.53	38.31 ± 0.37	1452.47 ± 13.00	46.81 ± 1.62	39.30 ± 1.04	49.17 ± 0.63	1445.78 ± 9.69	32.55 ± 0.81	29.27 ± 1.10	35.78 ± 1.44	1250.12 ± 159.92
	BSV	15.85 ± 0.89	4.66 ± 0.78	17.41 ± 0.93	372.57 ± 17.93	39.47 ± 0.61	35.47 ± 0.32	40.97 ± 0.54	5870.18 ± 163.87	34.41 ± 0.29	29.65 ± 0.53	38.05 ± 0.25	264.87 ± 21.88	42.10 ± 0.43	50.68 ± 0.77	44.78 ± 0.55	6139.14 ± 94.22
	Concat	14.18 ± 0.52	2.86 ± 0.85	15.47 ± 0.62	1591.06 ± 7.97	32.82 ± 1.59	27.42 ± 0.67	34.85 ± 0.88	78.33 ± 0.31	33.31 ± 2.16	27.45 ± 1.22	35.82 ± 2.32	429.57 ± 24.46	33.31 ± 2.16	27.45 ± 1.22	35.82 ± 2.32	429.57 ± 24.46
	APMC	42.20 ± 0.68	33.29 ± 0.51	43.02 ± 0.38	143.03 ± 0.85	38.10 ± 0.31	34.32 ± 0.53	38.31 ± 0.37	1452.47 ± 13.00	36.39 ± 1.35	33.08 ± 1.37	37.37 ± 2.22	1487.67 ± 22.70	36.39 ± 1.35	33.08 ± 1.37	37.37 ± 2.22	1487.67 ± 22.70
	DAIMC	44.81 ± 2.40	34.94 ± 1.31	45.97 ± 2.08	459.16 ± 17.21	39.47 ± 0.61	35.47 ± 0.32	40.97 ± 0.54	5870.18 ± 163.87	37.95 ± 1.87	36.07 ± 1.54	40.52 ± 0.71	5817.07 ± 36.32	37.95 ± 1.87	36.07 ± 1.54	40.52 ± 0.71	5817.07 ± 36.32
	PIC	48.85 ± 1.21	42.82 ± 0.85	48.98 ± 1.28	1519.19 ± 15.49	40.04 ± 3.17	30.72 ± 2.70	44.31 ± 2.15	51106.35 ± 776.00	46.81 ± 1.62	39.30 ± 1.04	49.17 ± 0.63	1445.78 ± 9.69	32.55 ± 0.81	29.27 ± 1.10	35.78 ± 1.44	1250.12 ± 159.92
	IMVTSC-MVI	51.74 ± 0.36	46.37 ± 0.88	52.69 ± 0.49	5594.82 ± 43.73	46.81 ± 1.62	42.98 ± 1.69	57.57 ± 3.03	2255.64 ± 120.99	54.46 ± 2.65	42.98 ± 1.69	57.57 ± 3.03	2255.64 ± 120.99	34.41 ± 0.29	29.65 ± 0.53	38.05 ± 0.25	264.87 ± 21.88
	IMSC_AGL	56.12 ± 3.55	47.56 ± 1.37	57.74 ± 2.50	54162.77 ± 1004.92	40.04 ± 3.17	30.72 ± 2.70	44.31 ± 2.15	51106.35 ± 776.00	46.81 ± 1.62	39.30 ± 1.04	49.17 ± 0.63	1445.78 ± 9.69	32.55 ± 0.81	29.27 ± 1.10	35.78 ± 1.44	1250.12 ± 159.92
	GIMC_FLSD	63.35 ± 3.51	51.89 ± 1.21	63.88 ± 2.38	1671.84 ± 21.13	54.46 ± 2.65	42.98 ± 1.69	57.57 ± 3.03	2255.64 ± 120.99	79.80 ± 1.85	78.25 ± 1.14	81.44 ± 1.91	6267.05 ± 169.78	42.10 ± 0.43	50.68 ± 0.77	44.78 ± 0.55	6139.14 ± 94.22
	MKKM-1K-MKC	66.89 ± 5.20	53.13 ± 2.18	68.37 ± 3.67	1776.77 ± 150.40	79.80 ± 1.85	78.25 ± 1.14	81.44 ± 1.91	6267.05 ± 169.78	94.22 ± 0.24	86.53 ± 0.34	94.22 ± 0.24	6645.57 ± 426.34	92.46 ± 0.40	83.55 ± 0.69	92.46 ± 0.40	4795.45 ± 569.02
AGC_IMC	95.12 ± 0.18	88.55 ± 0.43	95.12 ± 0.18	6445.07 ± 267.38	94.22 ± 0.24	86.53 ± 0.34	94.22 ± 0.24	6645.57 ± 426.34	87.14 ± 8.74	78.17 ± 10.59	87.68 ± 7.89	7792.73 ± 3574.54	92.46 ± 0.40	83.55 ± 0.69	92.46 ± 0.40	4795.45 ± 569.02	
MKAGT_IMVC	87.14 ± 8.74	78.17 ± 10.59	87.68 ± 7.89	7792.73 ± 3574.54	94.22 ± 0.24	86.53 ± 0.34	94.22 ± 0.24	6645.57 ± 426.34									

integrates low-rank tensor constraint and consensus graph constraint.

3. The experimental results in terms of CPU running time are shown in Tables 3 and 4, we can observe that as the number of samples increases, the running time of all methods will increase significantly, especially on the Scene-15 and Hdigit datasets. Obviously, the CPU running time of APMC is much less than all comparison methods in most incomplete situations, especially on the large-scale datasets (i.e., Hdigit). The main reason is that APMC adopts the strategy of generating anchor points to construct the similarity graph, and its optimization process is a non-iterative learning algorithm. This proves that the anchor point strategy can effectively improve the operating efficiency of the incomplete multi-view clustering method. However, the clustering performance of APMC in terms of ACC, NMI, and Purity is not very good compared to other competitors, and it cannot handle datasets with sample incomplete rate of 100%, or datasets with more than 3 views. For the proposed MKAGT_IMVC method, under the condition of ensuring the best clustering performance in terms of ACC, NMI, and Purity, the CPU running time is much less than some state-of-the-art competitors (i.e., IMSC_AGL, AGC_IMC, and IMVTSC-MVI) in most scenarios, which validates the superior performance of the proposed MKAGT_IMVC method in terms of clustering ability and time-consuming.
4. From Table 4, we can observe that the standard deviation values got by the proposed MKAGT_IMVC method in terms of ACC, NMI, Purity are larger than other competitors on the Scene-15 dataset with sample incomplete rate of 70% and 100%, and the Hdigit dataset with sample incomplete rate of 70%. This is mainly because that the generated anchor points are not representative or the number is not enough in some random incomplete patterns with the same sample incomplete rate. In addition, it can be observed that as the sample incomplete rate increases, the clustering performance of all comparison methods in terms of ACC, NMI, Purity commonly decreases. However, the proposed MKAGT_IMVC method still maintains stable performance in terms of ACC, NMI, Purity in most cases, especially on the large-scale datasets (i.e., Hdigit). The main reason is that the proposed MKAGT_IMVC method selects enough and representative instances as the anchor points for view-wise to construct anchor graphs.

5.3 Effect of anchor selection

We explore the effect of anchor selection on the clustering performance of the proposed MKAGT_IMVC methods by

discussing the two core issues, that is, what differences will be made when MKAGT_IMVC adopts different anchor strategies and different number of anchors. Since the current anchor point selected by DAS and the next anchor point is not the same cluster with high probability, we assume that the anchor points selected within c times are evenly distributed in c clusters. To this end, the number of anchors can be obtained by the following expression:

$$l_{anchorNum} = \text{ceil}\left(\min\left(\text{floor}\left(\frac{n_Y}{c}\right)\right) \times \eta\right) \times c \quad (43)$$

where n_Y records the number of available instances in each view, c denotes the number of clusters, η denotes the proportion of anchors to the available instances, $\text{floor}(\cdot)$ rounds each element to the nearest integer less than or equal to that element, $\text{ceil}(\cdot)$ rounds each element to the nearest integer greater than or equal to that element.

The effect of different anchor strategies on the clustering performance of the proposed MKAGT_IMVC method is presented in Table 5. We adopt four anchor selection strategies, including the widely used k -means policy, the random policy, the manner of common instances appearing in all views, and the improved DAS for incomplete multi-view data, where the number of anchor points of DAS is the same as random policy. From the experimental results, we can observe that the DAS has relatively stable clustering performance in terms of ACC, NMI, and Purity compared to the other three strategies. In addition, DAS not only has a significant lead in clustering performance on the medium-size datasets (i.e., Scene-15) but also its CPU running time is much less than the random strategy with the same number of anchor points. The main reason is that DAS can efficiently select a sufficient number of the most representative anchor points from the incomplete multi-view datasets. Therefore, the improved DAS for incomplete multi-view data is the most suitable for the anchor point selection in the proposed MKAGT_IMVC method.

The effect of different number anchor points on the clustering performance of the proposed MKAGT_IMVC method is shown in Fig. 2. From the experimental results, we can observe that the start values of the clustering performance curves for ACC and NMI have relatively large values among all reported values. The overall trend of clustering performance curves keeps rising steadily with the increase of the number of anchor points on small-size datasets (i.e., BBCSport, Caltech7, BBC). Meanwhile, on the medium-size dataset (i.e., Scene-15), the clustering performance curves fluctuate up and down within a certain range as the number of anchor points increases. Whereas, it can be observed empirically that the proportion of anchors to the available instances η is within 0.5 and the optimal clustering performance can be obtained. Moreover, we can find that as the number of anchor points increases, the

Table 5 The clustering performance (mean% ± std%) and CPU running time (sec.) (mean ± std) of the proposed MKAGT.IMVC method equipped with different anchor selection strategies on the BBCSport,

BBC, Caltech7, and Scene-15 datasets, where the sample incomplete rates of Caltech7 and Scene-15 are 90%. Bold numbers denote the best results

Dataset	Anchor strategy \ Metric	ACC	NMI	Purity	CPU time
BBCSport	<i>K</i> -means	35.01 ± 0.00	12.42 ± 0.00	40.98 ± 0.00	7.16 ± 0.00
	Common	99.46 ± 0.00	97.91 ± 0.00	99.46 ± 0.00	35.39 ± 0.00
	Random	99.73 ± 0.00	98.90 ± 0.00	99.73 ± 0.00	33.42 ± 0.00
	DAS	99.73 ± 0.00	99.01 ± 0.00	99.73 ± 0.00	34.33 ± 0.00
BBC	<i>K</i> -means	27.51 ± 0.00	4.02 ± 0.00	30.70 ± 0.00	90.36 ± 0.00
	Common	94.56 ± 0.00	86.65 ± 0.00	94.56 ± 0.00	358.13 ± 0.00
	Random	99.15 ± 0.00	96.85 ± 0.00	99.15 ± 0.00	188.15 ± 0.00
	DAS	99.01 ± 0.00	96.32 ± 0.00	99.01 ± 0.00	193.59 ± 0.00
Caltech7	<i>K</i> -means	28.87 ± 2.25	5.28 ± 4.26	56.54 ± 2.68	115.08 ± 0.35
	Common	49.50 ± 2.71	62.91 ± 0.67	87.25 ± 0.32	62.94 ± 6.59
	Random	61.93 ± 4.59	72.48 ± 2.90	91.95 ± 1.65	100.73 ± 3.86
	DAS	59.73 ± 3.32	69.81 ± 2.86	90.09 ± 1.35	111.43 ± 13.15
Scene-15	<i>K</i> -means	13.24 ± 0.39	9.29 ± 0.92	15.84 ± 0.60	515.55 ± 27.81
	Common	59.26 ± 14.04	55.95 ± 13.41	62.21 ± 13.79	963.54 ± 263.58
	Random	47.76 ± 1.32	41.40 ± 1.29	49.77 ± 2.00	1433.83 ± 40.90
	DAS	68.61 ± 4.13	66.71 ± 3.47	71.28 ± 3.07	820.27 ± 101.86

overall trend of the CPU running time curve shows a rapid rise. Therefore, for small-size datasets, we should maximize the number of anchor points, find the best number of anchor points in the direction of $\eta \geq 0.5$, while for medium-size or large-size datasets, we can look for the optimal number of anchors in the direction of $\eta \leq 0.5$.

5.4 Effect of kernel selection

To investigate the effect of the combination of different predefined kernel functions on the clustering performance of the proposed MKAGT.IMVC method, as shown in Table 6, we adopt three predefined kernel functions,

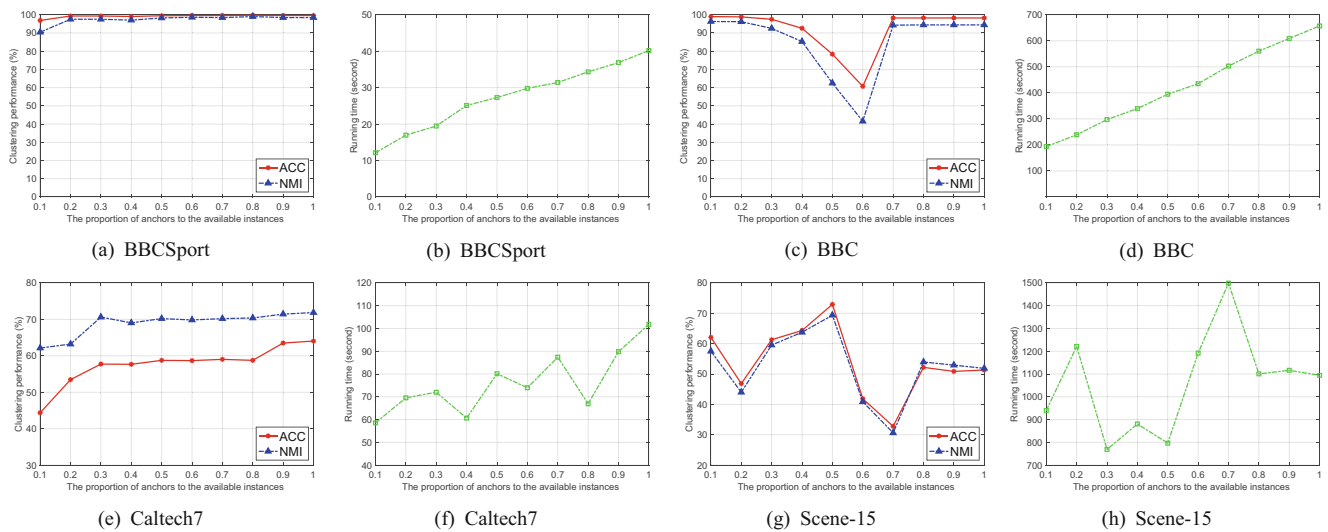


Fig. 2 The clustering performance and CPU running time of the proposed MKAGT.IMVC method with the varying number of anchor points on the BBCSport, BBC, Caltech7, and Scene-15 datasets, where the sample incomplete rates of Caltech7 and Scene-15 are 90%

Table 6 The clustering performance (mean% \pm std%) and CPU running time (sec.) (mean \pm std) of the proposed MKAGT.IMVC method equipped with different kernel selection strategies on the BBCSport,

BBC, Caltech7, and Scene-15 datasets, where the sample incomplete rates of Caltech7 and Scene-15 are 90%. Bold numbers denote the best results

Dataset	Kernel strategy \ Metric	ACC	NMI	Purity	CPU time
BBCSport	2 RBF Kernels	99.32 \pm 0.00	97.54 \pm 0.00	99.32 \pm 0.00	10.68 \pm 0.00
	2 RBF Chi-Square Kernels	31.89 \pm 0.00	1.19 \pm 0.00	35.96 \pm 0.00	61.31 \pm 0.00
	3 RBF Kernels	99.86 \pm 0.00	99.44 \pm 0.00	99.86 \pm 0.00	14.18 \pm 0.00
	3 RBF Chi-Square Kernels	91.99 \pm 0.00	80.82 \pm 0.00	91.99 \pm 0.00	82.35 \pm 0.00
	4 RBF Kernels	99.59 \pm 0.00	98.47 \pm 0.00	99.59 \pm 0.00	17.53 \pm 0.00
	4 RBF Chi-Square Kernels	93.89 \pm 0.00	85.13 \pm 0.00	93.89 \pm 0.00	107.69 \pm 0.00
	the proposed multiple Kernels	99.73 \pm 0.00	99.01 \pm 0.00	99.73 \pm 0.00	34.33 \pm 0.00
BBC	2 RBF Kernels	98.52 \pm 0.00	94.79 \pm 0.00	98.52 \pm 0.00	103.02 \pm 0.00
	2 RBF Chi-Square Kernels	36.31 \pm 0.00	23.70 \pm 0.00	37.08 \pm 0.00	261.86 \pm 0.00
	3 RBF Kernels	99.06 \pm 0.00	96.52 \pm 0.00	99.06 \pm 0.00	147.81 \pm 0.00
	3 RBF Chi-Square Kernels	43.28 \pm 0.00	33.29 \pm 0.00	47.91 \pm 0.00	358.98 \pm 0.00
	4 RBF Kernels	99.19 \pm 0.00	96.88 \pm 0.00	99.19 \pm 0.00	184.06 \pm 0.00
	4 RBF Chi-Square Kernels	64.58 \pm 0.00	64.31 \pm 0.00	74.02 \pm 0.00	475.15 \pm 0.00
	the proposed multiple Kernels	99.01 \pm 0.00	96.32 \pm 0.00	99.01 \pm 0.00	193.59 \pm 0.00
Caltech7	2 RBF Kernels	59.04 \pm 1.87	69.46 \pm 1.25	90.27 \pm 0.32	47.03 \pm 7.29
	2 RBF Chi-Square Kernels	56.11 \pm 4.99	66.55 \pm 5.28	89.27 \pm 2.28	92.76 \pm 4.03
	3 RBF Kernels	58.21 \pm 2.26	69.29 \pm 2.83	90.00 \pm 1.38	78.53 \pm 13.04
	3 RBF Chi-Square Kernels	61.21 \pm 4.60	70.00 \pm 2.60	90.14 \pm 1.31	131.51 \pm 25.84
	4 RBF Kernels	58.75 \pm 4.59	68.57 \pm 2.69	89.82 \pm 1.26	112.39 \pm 32.36
	4 RBF Chi-Square Kernels	56.02 \pm 2.18	66.15 \pm 1.75	88.13 \pm 1.10	219.52 \pm 32.32
	the proposed multiple Kernels	59.73 \pm 3.32	69.81 \pm 2.86	90.09 \pm 1.35	111.43 \pm 13.15
Scene-15	2 RBF Kernels	43.61 \pm 13.10	40.82 \pm 12.74	46.95 \pm 13.13	828.26 \pm 139.48
	2 RBF Chi-Square Kernels	55.87 \pm 15.27	55.41 \pm 16.01	58.60 \pm 15.24	759.33 \pm 248.76
	3 RBF Kernels	50.95 \pm 17.49	48.69 \pm 17.07	53.99 \pm 17.12	1091.97 \pm 197.65
	3 RBF Chi-Square Kernels	55.34 \pm 18.31	54.71 \pm 17.76	58.66 \pm 17.22	1075.95 \pm 181.83
	4 RBF Kernels	28.50 \pm 1.42	24.44 \pm 1.07	31.04 \pm 1.44	1772.48 \pm 61.97
	4 RBF Chi-Square Kernels	29.79 \pm 0.72	28.00 \pm 1.44	32.86 \pm 0.92	1706.98 \pm 29.51
	the proposed multiple Kernels	68.61 \pm 4.13	66.71 \pm 3.47	71.28 \pm 3.07	820.27 \pm 101.86

including RBF Kernel, Cosine Similarity Kernel, and RBF Chi-Square Kernel, to form seven combinations to conduct comparative experiments on four incomplete multi-view datasets. For the kernel parameter γ setting of a certain competitive combination, according to the number of kernel functions in the combination, the corresponding values are selected from the set $\{10, 1, 0.1, 0.01\}$ in turn. From the experimental results, we can observe that the proposed combination of multiple kernels performs robustly on the clustering performance in terms of ACC, NMI, and Purity than other comparative combinations, especially on the medium-size datasets. For example, in the Scene-15 dataset, the proposed combination not only obtains the optimal clustering performance but also its CPU running time is much less than most other comparison combinations. The main reason is that when we fuse multiple incomplete views, using multiple predefined kernel

functions of different properties to construct multiple fused complete anchor graphs, the proposed MKAGT.IMVC method not only sufficiently mines the intra-view and inter-view information but also well captures more nonlinear relations between samples. Therefore, these demonstrate that the proposed MKAGT.IMVC method adopts a variety of predefined kernel functions with different kernel spaces, which can improve the effectiveness, robustness, and efficiency of the clustering results.

5.5 Parameter sensitivity analysis

There are two tunable parameters λ_1 and λ_2 in our final objective function as formulated by (18). In the following, we conduct some experiments to explore the influence of different parameter combinations on clustering performance. As shown in Fig. 3, we present

Fig. 3 ACC (%) versus parameters λ_1 and λ_2 of the proposed MKAGT_IMVC method on the BBCSport, BBC, Caltech7, and Scene-15 datasets, where the sample incomplete rates of Caltech7 and Scene-15 are 90%

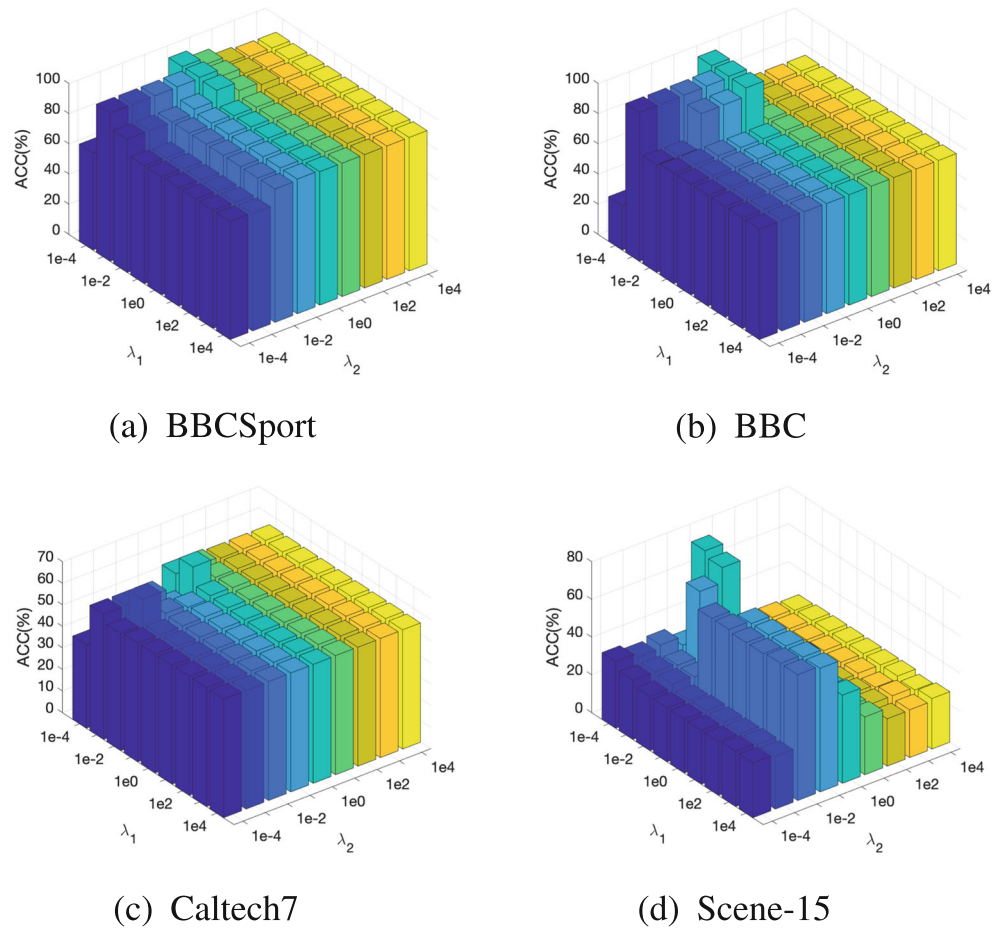
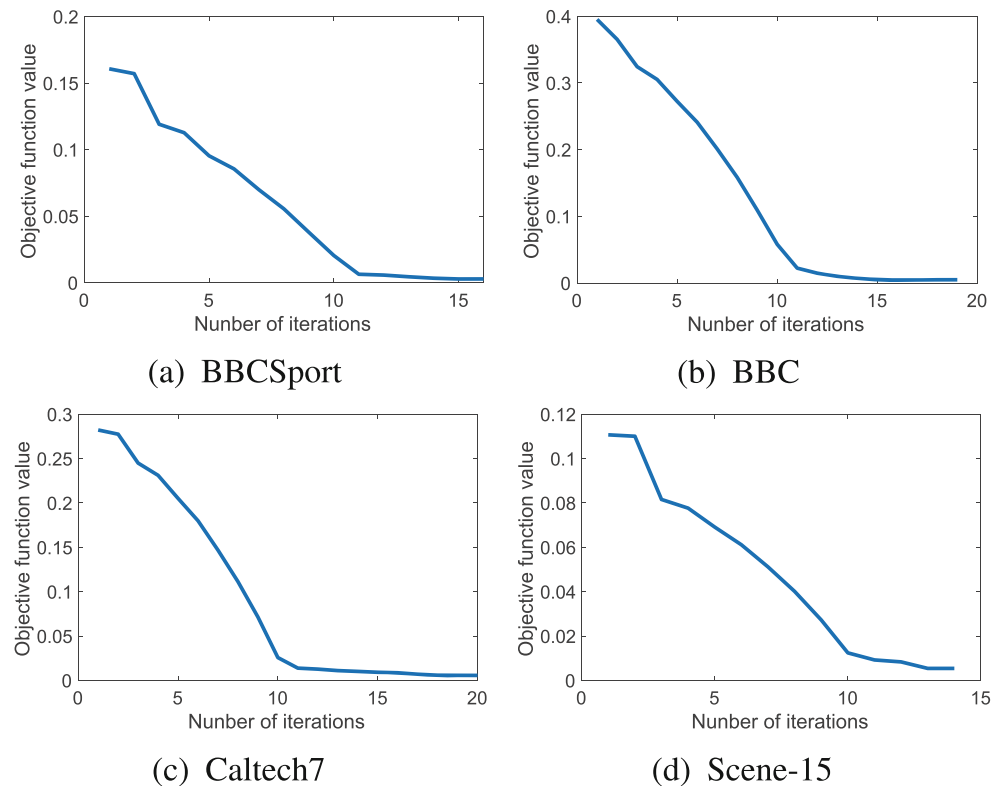


Fig. 4 The objective function value versus the number of iterations of the proposed MKAGT_IMVC method on the BBCSport, BBC, Caltech7, and Scene-15 datasets, where the sample incomplete rates of Caltech7 and Scene-15 are 90%



the clustering performance in terms of ACC in different combinations of the two trade-off parameters λ_1 and λ_2 on the BBCSport, BBC, Caltech7, and Scene-15 datasets, where λ_1 and λ_2 are both derived from $\{1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3, 1e4\}$. From the experimental results, we can observe that the proposed MKAGT_IMVC method is a little sensitive to parameters λ_1 and λ_2 , but in most cases, setting the two parameters to $\lambda_1 = 0.001$ and $\lambda_2 = 1$ can guarantee the best clustering performance. Obviously, the optimal value of parameter λ_2 is several orders of magnitude larger than the optimal value of parameter λ_1 . This is reasonable since λ_2 is used to adjust the proportion of the diversity term in (18), where the diversity term mainly refers to the complementary and consistent information of multiple views. These experiments further demonstrate that our method can effectively capture complementary and consistent information between views from multi-view datasets to improve clustering performance.

5.6 Empirical convergence analysis

In Section 4.5, we analyze the convergence property of the proposed MKAGT_IMVC method theoretically. In the following, we conduct some experiments to further analyze its convergence property. As shown in Fig. 4, we report the results of the convergence experiment on four incomplete multi-view datasets. The experiment results show that the objective function values decrease monotonically and go stable within 20 iterations in all cases. These experiments further demonstrate that the proposed MKAGT_IMVC method has excellent convergence.

6 Conclusion

While recently proposed incomplete multi-view clustering methods have achieved advanced performance, they still have several limitations to some extent, such as inflexibility and relatively high computational cost; neglecting valuable nonlinear relations among samples; failure to sufficiently explore complementary information among views; etc. This paper develops a novel and flexible unified graph learning framework called MKAGT_IMVC for addressing the above drawbacks. MKAGT_IMVC first extends an anchor selection approach that can simply and effectively generate a bundle of representative anchor points. Then, based on the selected anchor points and pre-defined kernel functions, multiple fused complete anchor graphs are constructed to capture more valuable nonlinear relations. Moreover, MKAGT_IMVC develops a graph learning model that integrates low-rank tensor constraint and consensus graph constraint, where all fused complete anchor graphs are

regarded as prior knowledge to initialize this model. These key steps make our method distinct yet superior to most methods. Extensive experiments clearly show that our method outperforms most state-of-the-art competitors in terms of clustering ability and time-consuming. In the future, we plan to further study how to adaptively select the number of anchors, rather than experimental assessments.

Acknowledgments This work was supported in part by the Joint Fund of the National Natural Science Foundation of China and Guangdong Province under Grant U1701266, in part by the Guangdong Provincial Key Laboratory of Intellectual Property and Big Data under Grant 2018B030322016, in part by the Natural Science Foundation of Guangdong Province under Grant 2018A030313751, in part by the Guangzhou key field research and development plan project under Grant 202206070003, in part by the Qingyuan Science and Technology Plan Project under Grant 2018D004 and 2018D005, and in part by the Special Projects for Key Fields in Higher Education of Guangdong Province under Grant 2020ZDZX3077 and 2021ZDZX1042.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bickel S, Scheffer T (2004) Multi-view clustering. In: Fourth IEEE international conference on data mining (ICDM'04), vol 4, Citeseer, pp 19–26
2. Greene D, Cunningham P (2006) Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proceedings of the 23rd international conference on machine learning, pp 377–384
3. Dai D, Van Gool L (2013) Ensemble projection for semi-supervised image classification. In: Proceedings of the IEEE international conference on computer vision, pp 2072–2079
4. Li Y, Yang M, Zhang ZM (2019) A survey of multi-view representation learning. *IEEE Trans Knowl Data Eng* 31(10):1863–1883
5. Tolić D, Antulov-Fantulin N, Kopriva I (2018) A nonlinear orthogonal non-negative matrix factorization approach to subspace clustering. *Pattern Recogn* 82:40–55
6. Huang S, Kang Z, Xu Z (2020) Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recogn* 97:107015
7. Zhang Z, Liu L, Shen F, Shen HT, Shao L (2019) Binary multi-view clustering. *IEEE Trans Pattern Anal Mach Intell* 41(7):1774–1782
8. 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

9. Zhang G-Y, Zhou Y-R, He X-Y, Wang C-D, Huang D (2020) One-step kernel multi-view subspace clustering. *Knowl-Based Syst* 189:105126
10. Zhan K, Nie F, Wang J, Yang Y (2019) Multiview consensus graph clustering. *IEEE Trans Image Process* 28(3):1261–1270
11. Wang H, Yang Y, Liu B (2020) Gmc: Graph-based multi-view clustering. *IEEE Trans Knowl Data Eng* 32(6):1116–1129
12. Li X, Zhang H, Wang R, Nie F (2020) Multi-view clustering: a scalable and parameter-free bipartite graph fusion method. *IEEE Trans Pattern Anal Mach Intell*, 1–1
13. Li Z, Wang Q, Tao Z, Gao Q, Yang Z (2019) Deep adversarial multi-view clustering network. In: *IJCAI'19 Proceedings of the Twenty-Eighth international joint conference on artificial intelligence*, pp 2952–2958
14. Xu J, Ren Y, Li G, Pan L, Zhu C, Xu Z (2021) Deep embedded multi-view clustering with collaborative training. *Inf Sci* 573:279–290
15. Wen J, Xu Y, Liu H (2020) Incomplete multiview spectral clustering with adaptive graph learning. *IEEE Trans Cybern* 50(4):1418–1429
16. Liu J, Teng S, Fei L, Zhang W, Fang X, Zhang Z, Wu N (2021) A novel consensus learning approach to incomplete multi-view clustering. *Pattern Recogn* 115:107890
17. Li S-Y, Jiang Y, Zhou Z-H (2014) Partial multi-view clustering. In: *Twenty-eighth AAAI conference on artificial intelligence*
18. Yang Y, Wang H (2018) Multi-view clustering: A survey. *Big Data Mining and Analytics* 1(2):83–107
19. Hu M, Chen S (2018) Doubly aligned incomplete multi-view clustering. In: *Proceedings of the Twenty-Seventh international joint conference on artificial intelligence*, pp 2262–2268
20. 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
21. Xu C, Guan Z, Zhao W, Wu H, Niu Y, Ling B (2019) Adversarial incomplete multi-view clustering. In: *IJCAI'19 Proceedings of the Twenty-Eighth international joint conference on artificial intelligence*, pp 3933–3939
22. Wang Q, Ding Z, Tao Z, Gao Q, Fu Y (2021) Generative partial multi-view clustering with adaptive fusion and cycle consistency. *IEEE Trans Image Process* 30:1771–1783
23. Zhang C, Cui Y, Han Z, Zhou JT, Fu H, Hu Q (2020) Deep partial multi-view learning. *IEEE Trans Pattern Anal Mach Intell*, 1–1
24. Lin Y, Gou Y, Liu Z, Li B, Lv J, Peng X (2021) Completer: Incomplete multi-view clustering via contrastive prediction. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp 11174–11183
25. 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*, pp 3753–3761
26. Wen J, Zhang Z, Xu Y, Zhang B, Fei L, Xie G.-S. (2020) Cdimc-net: Cognitive deep incomplete multi-view clustering network. In: *IJCAI'20 Proceedings of the Twenty-Ninth international joint conference on artificial intelligence*, pp 3230–3236
27. Liu X, Zhu X, Li M, Wang L, Zhu E, Liu T, Kloft M, Shen D, Yin J, Gao W (2020) Multiple kernel k -means with incomplete kernels. *IEEE Trans Pattern Anal Mach Intell* 42(5):1191–1204
28. Liu X, Zhu X, Li M, Wang L, Tang C, Yin J, Shen D, Wang H, Gao W (2019) Late fusion incomplete multi-view clustering. *IEEE Trans Pattern Anal Mach Intell* 41(10):2410–2423
29. Liu X, Li M, Tang C, Xia J, Xiong J, Liu L, Kloft M, Zhu E (2020) Efficient and effective regularized incomplete multi-view clustering. *IEEE Trans Pattern Anal Mach Intell*, 1–1
30. Wang H, Zong L, Liu B, Yang Y, Zhou W (2019) Spectral perturbation meets incomplete multi-view data. In: *Proceedings of the Twenty-Eighth international joint conference on artificial intelligence*, pp 3677–3683
31. Wen J, Yan K, Zhang Z, Xu Y, Wang J, Fei L, Zhang B (2020) Adaptive graph completion based incomplete multi-view clustering. *IEEE Transactions on Multimedia*, pp 1–1
32. Wen J, Zhang Z, Zhang Z, Zhu L, Fei L, Zhang B, Xu Y (2021) Unified tensor framework for incomplete multi-view clustering and missing-view inferring. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 35, pp 10273–10281
33. Guo J, Ye J (2019) Anchors bring ease: an embarrassingly simple approach to partial multi-view clustering. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 33, pp 118–125
34. Wang S, Cao J, Lei F, Dai Q, Liang S, Wing-Kuen Ling B (2021) Semi-supervised multi-view clustering with weighted anchor graph embedding. *Computational Intelligence and Neuroscience* 2021:1–22
35. Wen J, Zhang B, Xu Y, Yang J, Han N (2018) Adaptive weighted nonnegative low-rank representation. *Pattern Recogn* 81:326–340
36. Zhang P, Liu X, Xiong J, Zhou S, Zhao W, Zhu E, Cai Z (2020) Consensus one-step multi-view subspace clustering. *IEEE Trans Knowl Data Eng*, 1–1
37. Kilmer ME, Martin CD (2011) Factorization strategies for third-order tensors. *Linear Algebra Appl* 435(3):641–658
38. Wu J, Lin Z, Zha H (2019) Essential tensor learning for multi-view spectral clustering. *IEEE Trans Image Process* 28(12):5910–5922
39. Cai D, Chen X (2015) Large scale spectral clustering via landmark-based sparse representation. *IEEE Trans Cybern* 45(8):1669–1680
40. Ren Z, Sun Q, Wei D (2021) Multiple kernel clustering with kernel k -means coupled graph tensor learning. In: *Proceedings of the AAAI Conf Artif Intell*
41. Zhang X, Ren Z, Sun H, Bai K, Feng X, Liu Z (2021) Multiple kernel low-rank representation-based robust multi-view subspace clustering. *Inf Sci* 551:324–340
42. He F, Wang R, Jia W (2020) Fast semi-supervised learning with anchor graph for large hyperspectral images. *Pattern Recogn Lett* 130:319–326
43. Xie Y, Tao D, Zhang W, Liu Y, Zhang L, Qu Y (2018) On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *Int J Comput Vis* 126(11):1157–1179
44. Boyd SP, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3(1):1–122
45. Nie F, Wang X, Jordan M, Huang H (2016) The constrained laplacian rank algorithm for graph-based clustering. In: *Proceedings of the AAAI conference on artificial intelligence*, vol 30
46. Hu W, Tao D, Zhang W, Xie Y, Yang Y (2016) The twist tensor nuclear norm for video completion. *IEEE Trans Neural Netw Learn Syst* 28(12):2961–2973
47. Coates A, Ng A, Lee H An analysis of single-layer networks in unsupervised feature learning. In: *Proceedings of the Fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings*, pp 215–223 2011
48. Ren Z, Sun Q, Wu B, Zhang X, Yan W (2020) Learning latent low-rank and sparse embedding for robust image feature extraction. *IEEE Trans Image Process* 29:2094–2107
49. Fei-Fei L, Fergus R, Perona P (2004) Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: *2004 conference*

- on computer vision and pattern recognition workshop, IEEE, pp 178–178
50. Li Y, Nie F, Huang H, Huang J (2015) Large-scale multi-view spectral clustering via bipartite graph. In: Twenty-ninth AAAI conference on artificial intelligence
 51. Dua D, Graff C (2017) UCI Machine Learning Repository
 52. Yang Y, Newsam S (2010) Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, pp 270–279
 53. Lazebnik S, Schmid C, Ponce J Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE conference on computer vision & pattern recognition (CPRV'06), IEEE Computer Society, pp 2169–2178 2006
 54. Zhao H, Liu H, Fu Y (2016) Incomplete multi-modal visual data grouping. In: IJCAI'16 Proceedings of the Twenty-Fifth international joint conference on artificial intelligence, pp 2392–2398
 55. Ng A, Jordan M, Weiss Y (2001) On spectral clustering: Analysis and an algorithm. In: Advances in neural information processing systems, vol 14, pp 849–856
 56. Tao Z, Liu H, Fu H, Fu Y (2017) Image cosegmentation via saliency-guided constrained clustering with cosine similarity. In: Proceedings of the AAAI conference on artificial intelligence, vol 31
 57. Pele O, Werman M (2010) The quadratic-chi histogram distance family. In: European conference on computer vision, Springer, pp 749–762

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Senhong Wang received the M.S. degree in signal and information processing from Guangdong University of Technology, Guangzhou, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Information Engineering, Guangdong University of Technology, Guangzhou, China. His research interests include machine learning, computer vision and knowledge graph.



Jiangzhong Cao received his Ph.D. degree in communication and information system from School of Information Science and Technology, Sun Yat-sen University, China, in 2013. He is currently an Associate Professor with the School of Information Engineering, Guangdong University of Technology, Guangzhou, China. His research interests include computer vision, pattern recognition and deep learning.



Fangyuan Lei received his Ph.D. degree in information processing from the Northwestern Polytechnical University, China, in 2004. He is currently an Assistant Professor with the Guangdong Provincial Key Laboratory of Intellectual Property Big Data, Guangdong Polytechnic Normal University, China. He has published more than 30 articles in refereed journals and conference proceedings. His main research interests include artificial intelligence, big data technology and image processing.



Jianjian Jiang is a teacher with the Guangdong Provincial Key Laboratory of Intellectual Property Big Data, Guangdong Polytechnic Normal University. In 2021, he received his master's degree from Guangdong Polytechnic Normal University. His research areas include image processing and intellectual property data analysis.



Qingyun Dai received the Ph.D. degree in communication and information systems from South China University of Technology, Guangzhou, China, in 2001. From 2003 to 2006, she was a Post-Doctoral Researcher at Sun Yat-sen University, Guangzhou, China. She is currently a Professor in Guangdong Polytechnic Normal University and serves as the president of the university. She also leads Guangdong Provincial Key Laboratory of Intellectual


Property & Big Data. Her current research interests include image processing, knowledge graph, intellectual property big data analysis and integration of industry and education.



Bingo Wing-Kuen Ling received the B.Eng. (Hons) and M.Phil. degrees from the department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology, in 1997 and 2000, respectively, and the Ph.D. degree in the department of Electronic and Information Engineering from the Hong Kong Polytechnic University in 2003. In 2004, he joined the King's College London as a Lecturer. In 2010, he joined the University of

Lincoln as a Principal Lecturer and promoted to a Reader in 2011. In 2012, he joined the Guangdong University of Technology as a Full Professor. He is a Fellow of IET, a senior member of IEEE, a China National Young Thousand-People-Plan Distinguished Professor, a Guangdong Pearl Distinguished Scholar and a University Hundred-People-Plan Distinguished Professor. He serves in the nonlinear circuits and systems technical committee, the digital signal processing technical committee and the power electronics and systems technical committee of the IEEE Circuits and Systems Community. He was awarded the best reviewer prizes from the IEEE Instrumentation and Measurement Society in 2008 and 2012. He has also served as the guest editor-in-chief of several special issues of highly rated international journals, such as the *Circuits, Systems and Signal Processing*, and the *American Journal of Engineering and Applied Sciences*, and is currently an associate editor of the *IET Signal Processing*, the *Circuits, Systems and Signal Processing*, *Measurement*, *Measurement: Sensors*, *Journal of the Franklin Institute*, *Journal of Industrial Management and Optimization*, etc. He has published an undergraduate textbook, a research monograph, several book chapters, a book review published in an IEEE journal, more than 200 internationally leading journal papers and more than 150 highly rated international conference papers. His research interest include optimization theory, time frequency analysis, image processing, biomedical signal processing, symbolic dynamics as well as fuzzy and impulsive control theory.

Affiliations

Senhong Wang¹  · Jiangzhong Cao¹ · Fangyuan Lei² · Jianjian Jiang² · Qingyun Dai² · Bingo Wing-Kuen Ling¹

Senhong Wang
shwang777@qq.com

Fangyuan Lei
leify@gpnu.edu.cn

Jianjian Jiang
371290422@qq.com

Qingyun Dai
dqy@gpnu.edu.cn

Bingo Wing-Kuen Ling
yongquanling@gdut.edu.cn

¹ School of Information Engineering, Guangdong University of Technology, Guangzhou, 510006, China

² Guangdong Provincial Key Laboratory of Intellectual Property and Big Data, Guangdong Polytechnic Normal University, Guangzhou, 510006, China