Efficient Anchor Learning-based Multi-view Clustering – A Late Fusion Approach

Tiejian Zhang
National University of Defense Technology
Changsha, China
zhangtiejian@nudt.edu.cn

Xinwang Liu*
National University of Defense Technology
Changsha, China
xinwangliu@nudt.edu.cn

En Zhu*
National University of Defense Technology
Changsha, China
enzhu@nudt.edu.cn

Sihang Zhou
National University of Defense Technology
Changsha, China
sihangjoe@gmail.com

Zhibin Dong
National University of Defense Technology
Changsha, China
dzb20@nudt.edu.cn

ABSTRACT

Anchor enhanced multi-view late fusion clustering has attracted numerous researchers' attention for its high clustering accuracy and promising efficiency. However, in the existing methods, the anchor points are usually generated through sampling or linearly combining the samples within the datasets, which could result in enormous time consumption and limited representation capability. To solve the problem, in our method, we learn the view-specific anchor points by learning them directly. Specifically, in our method, we first reconstruct the partition matrix of each view through multiplying a view-specific anchor matrix by a consensus reconstruction matrix. Then, by maximizing the weighted alignment between the base partition matrix and its estimated version in each view, we learn the optimal anchor points for each view. In particular, unlike previous late fusion algorithms, which define anchor points as linear combinations of existing samples, we define anchor points as a series of orthogonal vectors that are directly learned through optimization, which expands the learning space of the anchor points. Moreover, based on the above design, the resultant algorithm has only linear complexity and no hyper-parameter. Experiments on 12 benchmark kernel datasets and 5 large-scale datasets illustrate that the proposed Efficient Anchor Learning-based Multi-view Clustering (AL-MVC) algorithm achieves the state-of-the-art performance in both clustering performance and efficiency.

CCS CONCEPTS

• **Computing methodologies** → **Cluster analysis**; • **Theory of computation** → **Unsupervised learning and clustering**.

*Corresponding authors of this research.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548124

KEYWORDS

Multi-view clustering, Base partition, Consensus partition, Anchor point, Late fusion

ACM Reference Format:
Tiejian Zhang, Xinwang Liu, En Zhu, Sihang Zhou, and Zhibin Dong. 2022. Efficient Anchor Learning-based Multi-view Clustering – A Late Fusion Approach. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548124

1 INTRODUCTION

Multi-view data is a common data form in the fields of data analysis and computer vision [4, 10, 23–25, 29, 36, 43]. A large amount of unlabeled data is an important research object that researchers need to face [8, 9, 14, 27, 33, 42]. Facing this problem, a great deal of multi-view clustering is proposed [1, 3, 15, 16, 20, 31, 35, 39, 41, 44, 45, 47]. In order to meet the needs of large-scale multi-view data clustering, sampling-based multi-view clustering method is one of the most popular research topics [2, 7, 12, 13, 28, 32, 40]. In [13], researchers first select k independent anchors on each view by k-means method and replace the original map with the corresponding anchor map. After the affinity matrix and degree matrix are constructed by anchor graph, multi-view spectral clustering is used to solve the problem. [7] extends the use of anchors to multi-view subspace learning. It uses the k-means method to get the anchor points and then uses a similarity graph matrix to combine the anchor points to reconstruct the original sample linearly. [17, 32] considers that the anchors obtained by k-means are of low quality, which will limit the effect of graph learning and do harm to clustering performance. It learns an optimal consensus graph by fusing the anchor learning and the original sample reconstruction into a unified objective form.

The information fusion of the above methods at the sample level is easily affected by noise, which limits the performance of clustering. In the meantime, because the samples in the real world usually have higher dimensions, the operation cost in the data learning process is huge, which affects the efficiency of the algorithm. For this reason, methods of multi-view clustering based on late fusion are proposed. [37] learns the optimal consensus partition by maximizing the alignment of the consensus partition and the base partitions of each view. In order to prevent the result from deviating too far

from the average partition, the basis partition of the average kernel is introduced as a regularization term. [46] uses a consensus sampling matrix to divide the base partitions into linear combinations and then gets the anchors. After that, the data reconstruction matrix and anchor points of each view are used to reconstruct the base partition. Finally, reconstruction matrices are used to learn an optimal consensus partition.

Nevertheless, although existing multi-view methods have improved clustering performance in different aspects, we observe the following shortcomings in them: 1) The use of a sampling matrix for a linear combination of samples increases the risk that good anchors can not be learned from the poor quality of samples. 2) High computational and storage complexity limits the application of the algorithm to large-scale data. 3) The introduction of too many hyper-parameters increases the risk of over-fitting, and in the meantime, it increases the operation burden hundreds of times.

In order to overcome these shortcomings, this paper proposes a new late fusion multi-view clustering method, namely Efficient Anchor Learning-based Multi-view Clustering (AL-MVC). Specifically, we define anchor points as a series of orthogonal vectors that are directly learned through optimization, which expands the learning space of the anchor points. Then we reconstruct the partition matrix of each view by multiplying a view-specific anchor matrix by a consensus reconstruction matrix. Finally, we learn the optimal consensus reconstruction partition and optimal anchor points for each view with orthogonal constraints on them. In summary, the contributions of this paper can be summarized in the following three aspects:

(1) We design a late fusion framework based on anchor learning to reconstruct the base partitions of each view by the view-specific anchor and consensus reconstruction matrix. In the previous method, anchor points are selected by a linear combination of samples, and the quality of original samples greatly restricts the quality of anchor points. We define the anchor points as a series of orthogonal vectors directly through optimization learning, expanding the anchor points' learning space.

(2) We maximize the weighted alignment between the base partition matrix and each view estimate and replace the regularization term of the consensus reconstruction matrix in the previous method with orthogonal constraints. In this way, we enhance the representation ability of the clustering reconstruction matrix and realize the algorithm contained no hyper-parameter. The effectiveness of the proposed method is verified by experiments on multi-kernel datasets and large-scale datasets.

(3) By fusing multi-view information at the partition level, this framework can efficiently solve all multi-view clustering problems with base partition as input. With a sample size of $60,000$, the algorithm takes about just 3 seconds to run. The linear computation and storage complexity as well as the extremely fast convergence speed greatly expand the algorithm in the large-scale multi-view data application scenario.

For the rest of this article, we will cover the related work in section 2. We then describe the proposed method and its optimization process in section 3. All the experimental results are shown and analyzed in section 4. At the end of this paper, we summarize and put forward a prospect in section 5.

Table 1: Basic notations for the proposed AL-MVC

Notations	Meaning
n	Number of samples
m	Number of anchors
k	Number of clusters
l	Characteristic dimension of base partition
$\mathbf{X}_i \in \mathbb{R}^{n\times d}$	Raw data matrix of i-th view
$\mathbf{K}_i \in \mathbb{R}^{n\times n}$	Kernel matrix of i-th view
$\mathbf{H}_i \in \mathbb{R}^{n\times k}$	Base partitions i-th view
$\mathbf{C} \in \mathbb{R}^{n\times m}$	Consensus sampling matrix
$\mathbf{S}_i \in \mathbb{R}^{n\times m}$	Reconstruction matrix of i-th view
$\mathbf{A}_i \in \mathbb{R}^{l\times m}$	Anchor matrix of i-th view
$\mathbf{Z} \in \mathbb{R}^{m\times n}$	Consensus reconstruction matrix and partition
$\mathbf{I}_m \in \mathbb{R}^{m\times m}$	Identity matrix
β_i	Weight of the i-th view

2 RELATED WORK

2.1 Notations

To give a more explicit description of the proposed approach, we record the symbols in Table 1. We use lowercase, bold lowercase, and bold uppercase to represent scalars, vectors, and matrices.

2.2 Multi-kernel K-means

Kernel k-means is to map the data points in the input space to the high-level feature space through a nonlinear mapping and cluster in the new space. The nonlinear mapping increases the probability of data points' linear separability, achieving more accurate clustering results for non-convex data distribution. Further, multi-kernel k-means can handle clustering on multi-view data. Specifically:

In the i-th view, the sample point $\mathbf{x}$ is mapped to the reproducing kernel Hilbert space using a feature map $\phi_i(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}_i$. $\boldsymbol{\gamma}^\top = [\gamma_1, \cdots, \gamma_p]$ are the weights of the p kernel functions $\{\kappa_i(\cdot,\cdot)\}_{i=1}^p$ respectively. Then the kernel function can be defined as follows:

$$\kappa_{\boldsymbol{\gamma}}(\mathbf{x}_a, \mathbf{x}_b) = \phi_{\boldsymbol{\gamma}}^\top(\mathbf{x}_a)\phi_{\boldsymbol{\gamma}}(\mathbf{x}_b) = \sum_{i=1}^p \gamma_i^2 \kappa_i(\mathbf{x}_a, \mathbf{x}_b), \tag{1}$$

Furthermore, the optimal objective formula of MKKM(multi-kernel k-means) can be given in the following form:

$$\begin{aligned} &\min_{\mathbf{H},\boldsymbol{\gamma}} \ \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) \\ &s.t. \ \ \mathbf{H} \in \mathbb{R}^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \ \boldsymbol{\gamma}^\top\mathbf{1}_p = 1, \ \gamma_i \geq 0, \end{aligned} \tag{2}$$

Where n is the number of samples, k is number of clusters and $\mathbf{I}_k \in \mathbb{R}^{n\times k}$ is an identity matrix. $\mathbf{H}$ represents the optimal partition, it can be solved by alternately optimizing $\mathbf{H}$ and $\boldsymbol{\gamma}$ [6]. The complexity of eigenvalue decomposition is $O(kn^2)$ when gamma is fixed to solve $\mathbf{H}$.

2.3 Late Fusion Multi-view Clustering

In [37], researchers believe that although each view describes the characteristics of the data from a different angle, the implied clustering representation should be consistent. The basic idea of the method is to fuse multi-view information at the partition level, that is, to learn an optimal consensus partition by using base partitions that are obtained from different views. Particularly speaking:

On each view, $\{\mathbf{E}_i\}_{i=1}^p$ can be computed using spectral clustering, kernel k-means, and so on. $\mathbf{E}_i$ is the base partition that contains the clustering information. With the underlying partition as input, a method of maximizing alignment is designed in the following form:

$$\begin{aligned} \max_{\mathbf{E},\{\mathbf{B}_i\}_{i=1}^p,\boldsymbol{\gamma}} \quad & \mathrm{Tr}\left(\mathbf{E}^\top \mathbf{X}\right) + \mu \mathrm{Tr}\left(\mathbf{E}^\top \mathbf{Q}\right) \\ s.t. \quad & \mathbf{E}^\top \mathbf{E} = \mathbf{I}_k, \ \mathbf{B}_i^\top \mathbf{B}_i = \mathbf{I}_k, \ \sum_{i=1}^p \gamma_i^2 = 1, \\ & \gamma_i \geq 0, \ \mathbf{X} = \sum_{i=1}^p \gamma_i \mathbf{E}_i \mathbf{B}_i, \end{aligned} \tag{3}$$

$\mathbf{E} \in \mathbb{R}^{n\times k}$ is the consensus partition that is ultimately needed. $\{\mathbf{B}_i\}_{i=1}^p \in \mathbb{R}^{k\times k}$ are rotation matrices used to align the features of different views. $\mathbf{Q} \in \mathbb{R}^{n\times k}$ represents the average partition and can be used to prevent the optimal partition from moving too far away from the mean partition. In order to fuse multi-view clustering information at the partition level, $\mathbf{E}$ is learned by a linear combination of base partition $\mathbf{X}$ and average partition $\mathbf{Q}$. μ is a trade-off hyper-parameter used to balance the two parts of the formula.

2.4 Anchor-based Late Fusion Multi-view Clustering

In order to obtain higher computational efficiency and better clustering performance, [46] improved the multi-view subspace methods by late fusion and anchor selection. Firstly, the base partition $\mathbf{H}_i \in \mathbb{R}^{l\times n}$ is obtained by KKM(kernel k-means) method, which can be regarded as a set of n samples with the characteristic number l. Then, a consensus sampling matrix $\mathbf{C} \in \mathbb{R}^{n\times m}$ is used to linearly combine n samples in $\mathbf{H}_i$ to obtain m anchors with high expressivity to reduce the number of samples. Finally, use the anchor points to rebuild the base partition $\mathbf{H}_i$ on each view. The goal of the algorithm is as follows:

$$\begin{aligned} \min_{\mathbf{C},\mathbf{S},\{\mathbf{S}_i\}_{i=1}^p} & -\sum_{i=1}^p \mathrm{Tr}(\mathbf{H}_i(\mathbf{H}_i\mathbf{C}\mathbf{S}_i{}^\top)^\top) + \alpha \sum_{i=1}^p \|\mathbf{S}-\mathbf{S}_i\|_{\mathrm{F}}^2, \\ s.t. \ & 0 \leq \mathbf{S}_i \leq 1, \ 0 \leq \mathbf{S} \leq 1, \ \mathbf{C}^\top \mathbf{C} = \mathbf{I}_m. \end{aligned} \tag{4}$$

In this framework, the consensus sampling matrix $\mathbf{C}$ merges the information of each view and guides the learning of the reconstruction matrix $\mathbf{S}_i \in \mathbb{R}^{n\times m}$. With the information of partition level in $\mathbf{S}_i$, the optimal consensus reconstruction matrix $\mathbf{S}$ can be learned. Moreover, the maximum alignment framework measures the reconstruction loss, which makes the optimization solution more convenient. In this way, the computational complexity of clustering is controlled within $O(n^2)$.

We think that this method still has some limitations: 1) Sampling $\mathbf{H}_i$ with the matrix $\mathbf{C}$ to generate anchor points, which will lead to the quality of anchor points being limited by $\mathbf{H}_i$'s quality, which is not conducive to the improvement of performance. 2) There is a quadratic term of $\mathbf{H}_i$ in the optimization, which leads to the increase of computational complexity with the quadratic of the sample size. 3) In the second term of the optimization formula, the consensus reconstruction matrix $\mathbf{S}$ is learned by fusing the base partition matrix $\mathbf{S}_i$ information of each view. The addition of the hyper-parameter of the second term results in a tenfold increase in computational complexity.

3 PROPOSED METHOD

In this section, we first introduce the motivation and formulation of the proposed AL-MVC method. Then we give the optimization process in detail. Finally, we theoretically analyze the convergence and complexity of the proposed method.

3.1 Motivation and Formulation

Previous studies have shown that using anchors to reconstruct data is an effective way to reduce computational complexity [13]. Moreover, fusing multi-view information at the partition level can significantly improve the algorithm's performance. In order to obtain the appropriate anchor points for the clustering task, we expect to merge the selection of anchor points and the process of clustering into a whole frame. However, unlike [46], we no longer use the sampling matrix $\mathbf{C}$ to extract the samples from $\mathbf{H}_i$ to get the anchor $\mathbf{H}_i\mathbf{C}$, instead of learning the anchor $\mathbf{A}_i \in \mathbb{R}^{l\times m}$ for each view directly. The advantages of this approach are: 1) It avoids the high-quality anchor that can not be sampled due to the limitation of $\mathbf{H}_i$. 2) The quadratic term of $\mathbf{H}_i$ in the optimization formula is eliminated, which makes it possible to reduce the computational complexity from quadratic to linear. In addition, we set each view to have a different weight β_i. Based on the above, the formula can be written as follows,

$$\begin{aligned} \min_{\mathbf{A}_i,\mathbf{Z},\{\mathbf{Z}_i\}_{i=1}^p,\boldsymbol{\beta}} & -\sum_{i=1}^p \mathrm{Tr}(\beta_i \mathbf{H}_i(\mathbf{A}_i\mathbf{Z}_i)^\top) + \lambda \sum_{i=1}^p \|\mathbf{Z}-\mathbf{Z}_i\|_{\mathrm{F}}^2, \\ s.t. \ & 0 \leq \mathbf{Z}_i \leq 1, \ 0 \leq \mathbf{Z} \leq 1, \ \mathbf{A}_i^\top \mathbf{A}_i = \mathbf{I}_m, \ \|\boldsymbol{\beta}\|_2 = 1. \end{aligned} \tag{5}$$

Moreover, we think that it is not necessary to learn the reconstruction matrix $\mathbf{Z}_i \in \mathbb{R}^{m\times n}$ on each view and then use them to learn the consensus reconstruction matrix $\mathbf{Z} \in \mathbb{R}^{m\times n}$. Because $\mathbf{Z}$ has fused multiple view information at the partition level when reconstructing the base partitions $\mathbf{H}_i$ for each view. Furthermore, $\mathbf{Z}$ can be regarded as n sample points with m-dimensional characteristics. Each sample has a unique coordinate representation in the space formed by the base of the m dimension. In order to increase the information richness of the base and further improve the clustering representation of consensus partition, we use orthogonal constraints to replace the previous constraints on $\mathbf{Z}$. We end up with a formula that has no regularization term:

$$\begin{aligned} \max_{\mathbf{A}_i,\mathbf{Z},\boldsymbol{\beta}} \quad & \sum_{i=1}^p \beta_i \mathrm{Tr}(\mathbf{H}_i(\mathbf{A}_i\mathbf{Z})^\top) \\ s.t. \ & \mathbf{Z}\mathbf{Z}^\top = \mathbf{I}_m, \ \mathbf{A}_i^\top \mathbf{A}_i = \mathbf{I}_m, \|\boldsymbol{\beta}\|_2 = 1, \end{aligned} \tag{6}$$

where $\mathbf{H}_i \in \mathbb{R}^{l\times n}$ is the base partition of i-th view and l is fixed as $l = 2k$ in the experiment. $\mathbf{A}_i \in \mathbb{R}^{l\times m}$ represent the anchor points set of i-th view, while m is the number of anchors. $\mathbf{Z} \in \mathbb{R}^{m\times n}$ is the consensus partition matrix which is also a consensus reconstruction matrix. The optimal $\mathbf{Z}$ is used to input k-means to get the clustering result. β_i is the weight of each view. Anchor $\mathbf{A}_i$ and $\mathbf{Z}$ rebuild $\mathbf{H}_i$ and maximum alignment $\mathbf{H}_i$ to minimize the reconstruction loss.

3.2 Optimization

The above optimization formula is a nonconvex optimization problem because the optimization constraints are nonconvex. We use alternative optimization techniques to solve the problem, in which

Algorithm 1 The Proposed AL-MVC

Input: Base partitions $\{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_p\}$, number of clusters k, number of anchors m and tolerance threshold ϵ_0.
Output: consensus partition $\mathbf{Z}$.
Initialize each $\{\mathbf{A}_i^{(1)}\}_{i=1}^{p} \in \mathbb{R}^{l\times m}$ to a matrix with diagonal elements 1 and other elements 0, $\mathbf{Z}^{(0)}$ as a zero matrix, $\boldsymbol{\beta}^{(1)} = \frac{\mathbf{1}}{\sqrt{p}}$ and $t = 1$.
repeat
  Update $\mathbf{Z}^{(t)}$ by solving Eq. (8) with fixed $\mathbf{A}_i^{(t)}$ and $\boldsymbol{\beta}^{(t)}$.
  Update $\mathbf{A}_i^{(t)}$ by solving Eq. (7) with fixed $\boldsymbol{\beta}^{(t)}$ and $\mathbf{Z}^{(t)}$.
  Update $\boldsymbol{\beta}^{(t)}$ by solving Eq. (9) with fixed $\mathbf{Z}^{(t)}$ and $\mathbf{A}_i^{(t)}$.
  $t = t + 1$.
until $\left(\text{obj}^{(t-1)} - \text{obj}^{(t)}\right)/\text{obj}^{(t)} \leq \epsilon_0$

we optimize one variable with other variables fixed. Details are as follows:

3.2.1 ***Optimization of*** $\mathbf{A}_i$. Keeping $\mathbf{Z}$ and $\boldsymbol{\beta}$ constant, and noting that the $\mathbf{A}_j$ on each view is independent of each other, we can write an optimal formula for $\mathbf{A}_i$ as follows,

$$\begin{aligned} \max_{\mathbf{A}_i} \ & \text{Tr}(\mathbf{A}_i{}^\top \mathbf{W}_1) \\ s.t. \ & \mathbf{A}_i^\top \mathbf{A}_i = \mathbf{I}_m, \end{aligned} \tag{7}$$

where $\mathbf{W}_1 = \mathbf{H}_i\mathbf{Z}^\top$. For the above problems, we use the method in [46] to solve them. Using singular value decomposition(SVD) on $\mathbf{W}_1$, we can get $\mathbf{W}_1 = \mathbf{U}_1\Sigma_1\mathbf{V}_1^\top$. The optimal solution closed-form is $\mathbf{A}_i = \mathbf{U}_1\mathbf{V}_1^\top$. Each $\mathbf{A}_i$ of p views can be solved in accordance with this method.

3.2.2 ***Optimization of*** $\mathbf{Z}$. Keeping $\mathbf{A}_i$ and $\boldsymbol{\beta}$ constant, we can write an optimal formula for $\mathbf{Z}$ as follows,

$$\begin{aligned} \max_{\mathbf{Z}} \ & \text{Tr}(\mathbf{Z}^\top \mathbf{W}_2) \\ s.t. \ & \mathbf{Z}\mathbf{Z}^\top = \mathbf{I}_m, \end{aligned} \tag{8}$$

where $\mathbf{W}_2 = \sum_{i=1}^{p} \beta_i \mathbf{A}_i^\top \mathbf{H}_i$. Using singular value decomposition(SVD) on $\mathbf{W}_1$, we can get $\mathbf{W}_2 = \mathbf{U}_2\Sigma_2\mathbf{V}_2^\top$. The optimal solution closed-form is $\mathbf{Z} = \mathbf{U}_2\mathbf{V}_2^\top$.

3.2.3 ***Optimization of*** $\boldsymbol{\beta}$. Keeping irrelevant variables constant, we can write an optimal formula for $\boldsymbol{\beta}$ as follows,

$$\begin{aligned} \max_{\boldsymbol{\beta}} \ & \textstyle\sum_{i=1}^{p} \beta_i c_i \\ s.t. \ & \|\boldsymbol{\beta}\|_2 = 1, \end{aligned} \tag{9}$$

where $c_i = \text{Tr}(\mathbf{H}_i(\mathbf{A}_i\mathbf{Z})^\top)$ and $\boldsymbol{\beta}^\top = [\beta_1, \beta_2, ..., \beta_p]$. According to the Cauchy–Schwarz inequality, we can get the closed-form optimal solution,

$$\beta_i = c_i / \sqrt{\textstyle\sum_{i=1}^{p} {c_i}^2}. \tag{10}$$

We summarize the above optimization process in Algorithm 1 in which $obj^{(t)}$ represents the objective value of the t-th iteration. After getting the output $\mathbf{Z}$, k-means method is performed to get the final clustering result.

3.3 Convergence and Complexity Analysis

3.3.1 ***Convergence.*** In order to illustrate the convergence of the proposed algorithm, we need to prove that the optimization Eq. (6) is monotonically increasing and has an upper bound.

Theorem 1. *Optimization formula Eq. (6) has an upper bound.*

Proof. $\forall$ real number a, b satisfy $ab \leq \frac{1}{2}(a^2 + b^2)$. It is easy to extend it to matrix form $\text{Tr}(\mathbf{AB}) \leq \frac{1}{2}(\text{Tr}(\mathbf{A}\mathbf{A}^\top) + \text{Tr}(\mathbf{B}\mathbf{B}^\top))$. Then we can get $\text{Tr}(\mathbf{H}_i(\mathbf{A}_i\mathbf{Z})^\top) \leq \frac{1}{2}(\text{Tr}(\mathbf{H}_i\mathbf{H}_i^\top) + \text{Tr}((\mathbf{A}_i\mathbf{Z})^\top(\mathbf{A}_i\mathbf{Z})))$. $\mathbf{H}_i$ is a constant matrix, so $\text{Tr}(\mathbf{H}_i\mathbf{H}_i^\top)$ is a positive constant. Due to the orthogonal constraint of $\mathbf{A}_i$ and $\mathbf{Z}$, $\text{Tr}(\mathbf{H}_i\mathbf{H}_i^\top) + \text{Tr}((\mathbf{A}_i\mathbf{Z})^\top(\mathbf{A}_i\mathbf{Z}))$ is also a positive constant and can be represented by r. In this way, $\sum_{i=1}^{p} \beta_i \text{Tr}(\mathbf{H}_i(\mathbf{A}_i\mathbf{Z})^\top) \leq p\text{Tr}(\mathbf{H}_i(\mathbf{A}_i\mathbf{Z})^\top) = pr$. Therefore, Eq. (6) has an upper bound. □

In the iterative optimization algorithm we designed, every variable in each iteration has an optimal closed-form solution. This guarantees that the target value of the algorithm is monotonically increasing. In conclusion, we can ensure that the algorithm is convergent in theory.

3.3.2 ***Computational Complexity.*** We analyze the computational complexity of each step of Algorithm 1. When $\mathbf{Z}$ is updated, the computation complexity of eigenvalue decomposition and matrix multiplication are $O(m^2n+mln)$. When updating $\mathbf{A}_i$, the cost of the calculation is $O((m^2n+mln)p)$. When updating $\boldsymbol{\beta}$, the cost of the calculation is $O((l^2n+mln)p)$. Therefore, after t iterations, the total computational complexity is $O(t((l^2n+mln)p+(m^2n+mln)(p+1)))$. The proposed method has linear computational complexity.

3.3.3 ***Storage Complexity.*** Matrices in proposed method are $\mathbf{Z}$, $\mathbf{H}_i$, $\mathbf{A}_i$, $\mathbf{W}_1$, $\mathbf{W}_2$, which occupy the memory space of $O(2mn + pln + 2plm)$. Vectors $\boldsymbol{\beta}$ and $\boldsymbol{c}$ cost $O(2p)$ memory. Total storage complexity is $O(2mn + pln + 2p(lm + 1))$ which is also linear with respect to sample size n.

4 EXPERIMENTS

In this section, we first present the benchmark kernel datasets, the settings of the experiment, and the comparison algorithms. Then We evaluate the proposed algorithm experimentally from both clustering performance and the running time. Moreover, we design two ablation experiments to verify the validity of anchor-based learning and orthogonal constraint on $\mathbf{Z}$. In order to illustrate the performance and speed advantages of the proposed algorithm, we compare the performance of two new large-scale algorithms with ours. Finally, we complete the convergence analysis of the algorithm and explore the effect of the number of anchors on our method.

4.1 Benchmark Datasets and Experimental Settings

4.1.1 ***Datasets Introduction.*** In the proposed method, the base partitions are generated by base kernels via KKM algorithm. The basic information of 12 benchmark kernel datasets involved is listed in Table 2.

Table 2: Information of benchmark kernel datasets

Dataset	Samples	Kernel	Cluster	Data Type
BBCSport-2view	554	2	5	text
Reuters	18758	5	6	text
CCV	6773	6	20	video
Caltech101-10	1020	48	102	image
Caltech101-15	1530	48	102	image
Caltech101-20	2040	48	102	image
Caltech101-25	2550	48	102	image
Caltech101-30	3060	48	102	image
Flower102	8189	4	102	image
Scene15_mtv	4485	3	15	image
Football-9view	248	9	20	graph
SensITVehicle	1500	2	3	graph

4.1.2 ***Experimental Setup.*** In our experimental setting, the number of clusters k is known. Without a hyper-parameter, the characteristic dimension of the base partition is fixed as $l = 2k$ and the number of anchors is constant $m = k$. To make the comparison fair, we download the code from the corresponding website or request it from the corresponding author. Because all of the methods end up using k-means to get the final clustering results, we run k-means 20 times and calculate the average result of the clustering in each experiment. We used the widely accepted accuracy (ACC), normalized mutual information (NMI), and purity metrics to measure the clustering performance. All experiments are completed on a desktop computer with a 3.70GHz Intel(R) Core(TM) $i9-10900X$ CPU and 64GB RAM and MATLAB 2022a (64bit).

4.2 Compared Algorithms

To illustrate the effectiveness of the proposed method, several representative models are compared, including the classical method **MKKM** [5] which fuses multi-kernel information by linear combination, three improved multi-kernel methods **MKKM-MiR** [18], **SwMC** [26] and **ONKC** [21], two late fusion kernel method **LF-MVC** [37] and **OPLF** [19], a novel kernel method based on minimization and maximization problem **SMKKM** [22], two local kernel methods **LKAM** [11] and **SPMKC** [30], and a late fusion anchor-based representative models, **CSA-MKC** [46].

4.3 Experimental Results

The clustering performance of the proposed AL-MVC method with 10 comparison algorithms on 12 datasets is detailed in Table 3. The best results are shown in red, suboptimal results in blue, and '-' to represent out-of-memory or running time of more than a month. Analyzing the results of the experiment, we can draw the following conclusions:

1) As can be seen from Table 3, our algorithm shows excellent performance on ACC, NMI, and purity. On the 12 benchmark kernel datasets, the proposed method has the best ACC results. Only two NMI metrics and one purity metric are suboptimal but are very close to optimal ones.

2) Compared with other multi-kernel clustering methods, the four late fusion methods LF-MVC [37], OPLF [19], CSA-MKC [46] and our AL-MVC, have obvious advantages both in clustering performance and the number of hyper-parameters. They learn consensus partitioning by fusing partition-level information from multiple views and can mine deeper data features.

3) Compared with the three late fusion methods, the proposed method is still superior in performance. LF-MVC is a baseline of the late-fusion methods and is comprehensively surpassed by the proposed algorithm in performance. OPLF is also a late fusion approach with no hyper-parameter, yet our approach outperforms it on all datasets. Moreover, CSA-MKC is also an effective method, but the computational complexity with the quadratic of the sample size and two hyper-parameters limits its application.

The above experimental results have fully demonstrated the effectiveness of our method. In summary, our approach has the following advantages: 1) Our method reconstructs base partition with view-specific anchor matrices and a consensus reconstruction matrix, which integrate anchor learning and clustering tasks into a unified framework. 2) We define anchor points as a series of orthogonal vectors that are directly learned through optimization rather than linear combinations of existing samples in precious [46], which increases the representativeness of the anchors. 3) Moreover, based on the algorithm framework of maximum alignment, our method achieves linear computation and storage complexity without any hyper-parameter.

4.4 Ablation Study

This section demonstrates the validity of anchor-based learning and the effectiveness of applying orthogonal constraints to the consensus reconstruction matrix through two ablation experiments.

4.4.1 ***Design of Ablation Formula.*** In order to illustrate the validity of learning view-specific anchors, we design the formula of ablation 1 Eq. (11) to obtain anchor points by a linear combination of existing samples based on the consensus sampling matrix:

$$\begin{aligned} &\max_{\mathbf{C},\mathbf{Z},\boldsymbol{\beta}} \ \sum_{i=1}^{p} \beta_i \mathrm{Tr}(\mathbf{H}_i(\mathbf{H}_i\mathbf{C}\mathbf{Z})^\top) \\ &s.t.\ \mathbf{Z}\mathbf{Z}^\top = \mathbf{I}_m,\ \ \mathbf{C}^\top\mathbf{C} = \mathbf{I}_m, \|\boldsymbol{\beta}\|_2 = 1 \end{aligned} \tag{11}$$

where $\mathbf{C} \in \mathbb{R}^{n\times m}$ is the consensus sampling matrix.

What's more, we apply the constraint $0 \le \mathbf{Z} \le 1$ to the consensus partition matrix $\mathbf{Z}$ and get the formula of ablation 2 Eq. (12), which can demonstrate the effectiveness of orthogonal constraints on $\mathbf{Z}$,

$$\begin{aligned} &\max_{\mathbf{A}_i,\mathbf{Z},\boldsymbol{\beta}} \ \sum_{i=1}^{p} \beta_i \mathrm{Tr}(\mathbf{H}_i(\mathbf{A}_i\mathbf{Z})^\top) \\ &s.t.\ 0 \le \mathbf{Z} \le 1,\ \ \mathbf{A}_i^\top\mathbf{A}_i = \mathbf{I}_m, \|\boldsymbol{\beta}\|_2 = 1 \end{aligned} \tag{12}$$

4.4.2 ***Results of Ablation Study.*** In Table 4, we record the ACC, NMI, purity, and running time of the ablation experiments, respectively. Compared with the two ablation experiments, our method has significant advantages both in clustering performance and computation efficiency, which fully proves the validity of learning anchors and the effectiveness of applying orthogonal constraints to consensus partition.

Table 3: The performance of our algorithm and ten comparison algorithms on twelve benchmark kernel datasets.

Dataset	MKKM (2011)	MKKM-MiR (2016)	LKAM (2016)	ONKC (2017)	SwMC (2017)	LF-MVC (2019)	SMKKM (2020)	SPMKC (2020)	OPLF (2021)	CSA-MKC (2021)	Ours
Parameter number	0	1	2	2	0	1	0	4	0	2	0
ACC(%)											
BBCSport_2view	39.38	39.51	35.05	39.71	36.76	60.06	39.41	80.88	60.85	79.43	91.77
Caltech101-10	22.12	32.70	30.76	32.43	20.69	34.29	33.45	30.00	26.67	35.53	36.03
Caltech101-15	19.75	31.85	28.86	30.44	16.67	33.07	31.79	26.80	28.24	34.47	35.02
Caltech101-20	18.30	30.40	27.66	29.34	13.73	32.24	31.51	26.91	29.71	34.28	34.72
Caltech101-25	16.97	29.17	26.08	28.91	15.45	31.47	29.86	24.82	26.35	33.41	34.40
Caltech101-30	16.63	28.51	24.78	28.25	8.86	31.58	30.63	24.35	26.67	32.81	34.27
CCV	17.99	21.24	20.38	22.39	10.84	25.13	22.24	9.67	22.74	27.86	27.93
Flower102	22.41	40.22	41.40	39.55	6.72	38.45	42.51	-	29.78	46.12	46.29
Football-9view	72.96	77.12	76.25	78.49	35.08	79.07	70.38	45.97	80.24	84.12	85.03
Scene15_mtv	41.18	38.41	41.42	39.93	11.33	45.82	43.60	11.82	43.26	45.89	47.66
SensITVehicle	53.36	54.13	66.40	54.21	34.67	66.28	54.27	54.20	54.87	68.73	72.33
Reuters	45.44	46.15	-	41.85	-	45.68	45.52	-	44.65	46.26	46.54
NMI(%)											
BBCSport_2view	15.69	15.77	5.87	16.10	2.63	40.38	15.73	62.00	41.46	64.76	78.13
Caltech101-10	55.47	62.26	60.91	61.89	34.60	63.17	62.72	60.06	55.46	63.87	64.17
Caltech101-15	49.41	57.95	56.11	57.15	24.37	59.08	58.10	53.76	54.50	59.87	60.23
Caltech101-20	45.38	54.77	52.86	54.07	19.25	56.15	55.53	50.90	52.22	57.60	57.71
Caltech101-25	42.26	52.21	50.09	51.88	21.01	53.91	52.83	47.90	49.04	55.68	56.08
Caltech101-30	40.11	50.23	47.62	49.94	11.32	52.48	51.78	44.92	47.59	53.52	54.49
CCV	15.04	18.03	17.58	18.52	1.07	20.09	18.22	1.60	18.72	23.10	23.30
Flower102	42.67	56.71	56.89	56.11	5.51	54.94	58.63	-	46.77	60.74	60.85
Football-9view	78.91	79.88	79.67	81.18	37.34	83.79	75.91	48.90	86.69	90.01	88.89
Scene15_mtv	38.62	37.25	42.14	37.73	2.61	42.71	40.60	2.89	41.88	45.05	44.59
SensITVehicle	10.25	11.32	23.63	11.31	1.55	23.53	11.24	20.31	12.28	27.56	31.34
Reuters	27.35	25.30	-	22.27	-	27.39	27.75	-	27.09	27.55	30.01
PUR(%)											
BBCSport_2view	48.86	48.91	41.68	49.13	37.50	68.78	48.86	80.88	68.57	81.64	91.77
Caltech101-10	23.42	34.80	32.74	34.22	24.22	36.20	35.38	31.76	27.55	38.34	38.45
Caltech101-15	21.21	33.64	30.68	32.35	20.33	35.11	33.63	28.56	29.08	37.03	37.13
Caltech101-20	19.94	32.37	29.69	31.17	16.57	34.17	33.43	28.53	30.88	36.78	37.02
Caltech101-25	18.63	31.25	28.36	31.04	17.80	33.86	32.12	27.22	27.29	35.89	37.06
Caltech101-30	18.01	30.60	26.79	30.06	10.75	33.68	32.69	25.95	28.27	35.07	36.68
CCV	22.18	23.74	23.32	24.64	11.35	28.16	25.29	11.78	26.52	31.04	31.40
Flower102	27.79	46.34	48.05	45.63	8.08	44.56	48.64	-	34.03	52.80	53.11
Football-9view	75.71	78.35	77.74	80.10	39.11	82.26	72.30	47.98	82.26	86.68	87.58
Scene15_mtv	44.29	42.40	46.01	43.60	11.62	49.36	48.38	13.00	47.65	50.89	52.28
SensITVehicle	53.36	54.13	66.40	54.21	35.13	66.28	54.27	54.20	54.87	68.73	72.33
Reuters	52.94	52.15	-	52.63	-	53.23	53.27	-	52.92	66.71	65.67

4.5 Running Time Comparison

In order to illustrate the advantages of the algorithm in computational efficiency, we use LF-MVC time as a benchmark to get the relative running time of all methods, then show the logarithm of the base 10 to the relative running time in Figure 1. As can be seen from the figure, our method has an obvious speed advantage on the whole. Although our algorithm is slightly slower than LF-MVC on some small datasets, our algorithm does not contain a hyper-parameter and does not have to pay the hundredfold cost of traversing all the cases. Moreover, the computational efficiency advantage of our method is more evident on the larger datasets while it has excellent clustering performance.

4.6 Comparison with Large-scale Algorithms

Five large-scale datasets, ranging in size from 6,773 to 60,000, are selected and their information recorded in Table 5. To better illustrate the effectiveness and efficiency of the algorithm on large-scale data, we compare it with two latest large-scale multi-view algorithms SMVSC [32] proposed in 2021, FPMSC [38] proposed in 2022 and record the results in Table 6. As can be seen from the table, our algorithm not only has advantages in clustering performance but

Table 4: Results of ablation study on twelve benchmark datasets

Dataset	ACC(%)			NMI(%)			Purity(%)			Time(s)		
	Ablation1	Ablation2	Ours	Ablation1	Ablation2	Ours	Ablation1	Ablation2	Ours	Ablation1	Ablation2	Ours
BBCSport_2view	83.64	48.40	91.77	65.63	23.92	78.13	83.64	53.06	91.77	0.53	69.24	0.03
Caltech101-10	35.48	30.27	36.03	63.84	58.48	64.17	37.81	32.41	38.45	40.54	235.75	13.83
Caltech101-15	33.82	31.04	35.02	59.47	56.24	60.23	36.09	33.05	37.13	76.26	310.19	16.79
Caltech101-20	33.27	31.48	34.72	56.76	54.62	57.71	35.41	33.60	37.02	104.01	716.46	20.07
Caltech101-25	31.83	30.87	34.40	54.40	52.74	56.08	34.62	33.33	37.06	80.75	755.13	22.39
Caltech101-30	31.48	31.71	34.27	52.61	51.90	54.49	34.02	33.89	36.68	174.77	1133.74	24.90
CCV	27.44	26.73	27.93	21.57	21.59	23.30	29.96	30.26	31.40	56.13	2408.20	0.60
Flower102	37.18	43.19	46.29	51.99	56.98	60.85	42.77	49.66	53.11	165.48	2667.56	5.60
Football-9view	85.95	73.35	85.03	90.13	79.57	88.89	88.75	76.59	87.58	0.69	30.52	0.12
Scene15_mtv	44.97	42.97	47.66	43.20	38.92	44.59	50.52	45.78	52.28	18.14	1204.55	0.33
SensITVehicle	50.40	40.93	72.33	8.93	5.77	31.34	50.40	42.07	72.33	1.48	229.40	0.02
Reuters	45.60	43.26	46.54	30.66	19.73	30.01	65.00	64.94	65.67	220.16	6533.68	1.00

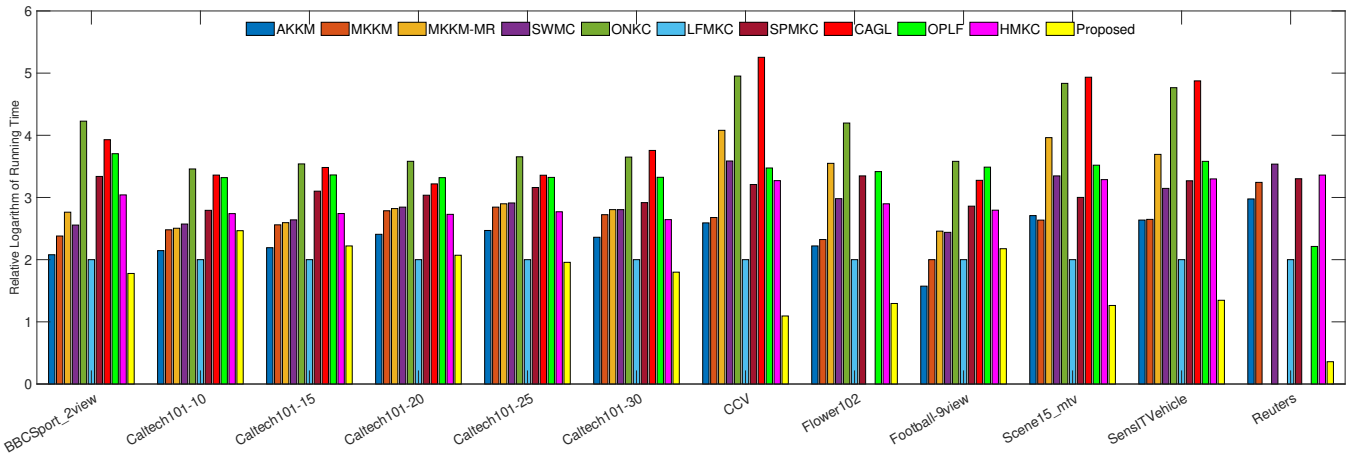
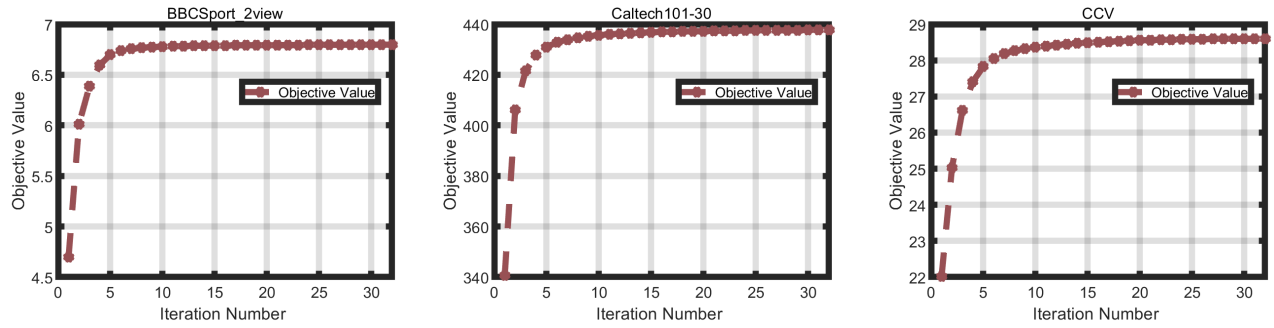


Figure 1: Time comparison on twelve benchmark kernel datasets.



(a) BBCSport_2view (b) Caltech101-30 (c) CCV

Figure 2: The target value of each iteration of the proposed method on three datasets.

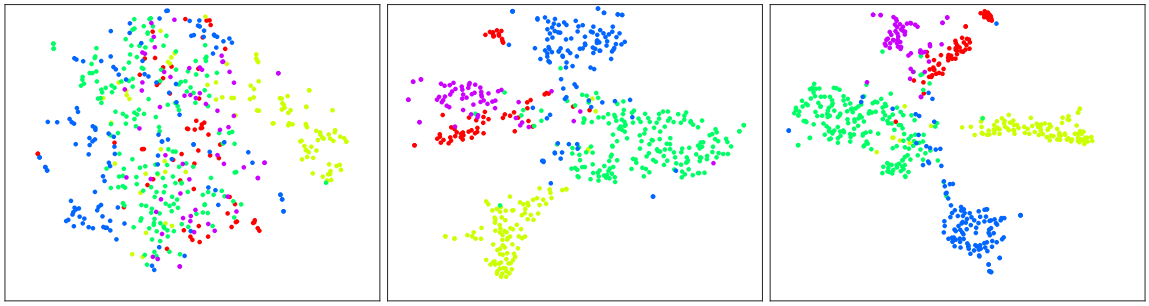
Table 5: Information of five large-scale datasets

Dataset	Sample	View(Kernel)	Cluster	Feature
CCV	6773	3	20	20,20,20
ALOI-100	10800	4	100	77,13,64,125
AWA	30475	6	50	2688,2000,252 2000,2000,2000
Cifar100	50000	3	10	512,2048,1024
MNIST	60000	3	10	342,1024,64

also outperforms the other two algorithms in computing speed. In particular, our method can achieve 99.08% ACC in less than 3 seconds by clustering MNIST dataset with 60000 samples. Because the dimensions of the original samples are much larger than those of the base partitions, our late fusion algorithm has more advantages in large-scale data than other large-scale methods.

4.7 Convergence Analysis

In Section 3, we have proved the convergence of the algorithm in theory. In this subsection, to further illustrate the convergence of



(a) BBCSport-2view(iter=1) (b) BBCSport-2view(iter=5) (c) BBCSport-2view(iter=15)

Figure 3: Consensus partition Z shown by t-SNE method when iterations are 1, 5, 15 on the BBCSport-2view dataset.

the algorithm, we first record the variation of the target value of iterations on 3 datasets in Figure 2. Then, the consensus partition matrix $\mathbf{Z}$ updated of each iteration on the BBCSport-2view is visualized using the t-SNE [34] method, and the results are shown in Figure 3. From Figure 2 we can see that the target value increases monotonically as the number of iterations increases. Observing from Figure 2 and Figure 3, it takes less than 15 iterations to converge in general.

4.8 Quantitative Analysis of Anchor Point

In our experimental setup, the number of anchors was fixed at $m = k$. We increase the search space for the number of anchors and set $m = [k, 2k, ..., 6k]$. The accuracy of the algorithm on Reuters dataset is shown in Figure 4. From the figure, we can find the optimal number of anchors is $m = 4k$, and ACC is over 50% in this case. Therefore, in future work, we will research the choice of the number of anchors.

5 CONCLUSION

In this paper, we propose a novel late fusion method – Efficient Anchor Learning-based Multi-view Clustering (AL-MVC) to enhance the performance and efficiency of multi-view clustering. We multiply the anchor matrix of each view by the consensus reconstruction matrix to reconstruct the base partition. Then maximum alignment partitions and their estimated versions are designed to learn the optimal anchor points and consensus reconstruction matrix. The orthogonal constraint is applied to the consensus reconstruction matrix to enhance its ability of feature representation and further improve its clustering performance. Based on the integration of anchor learning and clustering in a unified framework design, our algorithm not only shows excellent clustering performance but also has only linear computing and storage complexity without hyper-parameter. In the future, we will explore the optimal number of anchors and apply the algorithm to missing multi-view clustering based on the complementary information of multiple views.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61773392, 61872377, 61922088 and 61976196).

Table 6: Comparison with two latest large-scale algorithms

Dataset	Metric	FPMSC	SMC	Ours
CCV	ACC(%)	21.97	21.39	27.93
	NMI(%)	16.65	16.71	23.29
	Purity(%)	24.35	24.07	31.36
	Time(s)	87.44	13.90	0.60
ALOI-100	ACC(%)	33.97	35.37	67.98
	NMI(%)	65.14	58.15	79.07
	Purity(%)	34.92	36.30	69.82
	Time(s)	451.08	224.09	8.14
AWA	ACC(%)	9.06	9.33	9.95
	NMI(%)	10.66	9.97	11.27
	Purity(%)	9.46	10.47	12.01
	Time(s)	2060.00	2285.66	12.03
Cifar100	ACC(%)	67.32	72.33	91.92
	NMI(%)	90.67	88.18	97.93
	Purity(%)	67.86	73.30	93.98
	Time(s)	3457.74	2792.46	22.98
MNIST	ACC(%)	98.84	98.84	99.08
	NMI(%)	96.51	96.49	97.17
	Purity(%)	98.84	98.84	99.08
	Time(s)	1012.81	438.63	2.98

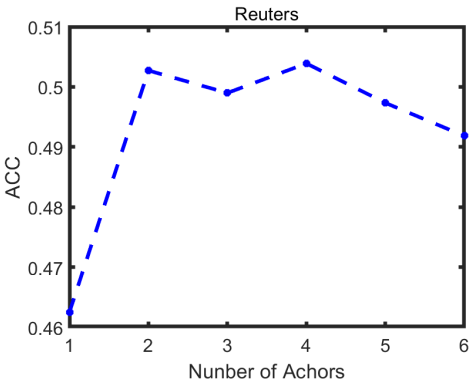


Figure 4: The effect of anchor number on proposed algorithm accuracy on Reuters dataset.

REFERENCES

[1] Man-Sheng Chen, Ling Huang, Chang-Dong Wang, Dong Huang, and Jian-Huang Lai. 2021. Relaxed multi-view clustering in latent embedding space. *Information Fusion* 68 (2021), 8 – 21.

[2] Xinlei Chen and Deng Cai. 2011. Large Scale Spectral Clustering with Landmark-Based Representation. In *Aaai Conference on Artificial Intelligence*.

[3] Yongyong Chen, Xiaolin Xiao, and Yicong Zhou. 2019. Jointly learning kernel representation tensor and affinity matrix for multi-view clustering. *IEEE Transactions on Multimedia* (2019).

[4] Bo Huang, Tingfa Xu, Shenwang Jiang, Yiwen Chen, and Yu Bai. 2020. Robust Visual Tracking via Constrained Multi-Kernel Correlation Filters. *IEEE Transactions on Multimedia* 22, 11 (2020), 2820–2832.

[5] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. 2011. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems* 20, 1 (2011), 120–134.

[6] S. Jegelka, A. Gretton, B Schölkopf, B. Sriperumbudur, U Von Luxburg, M. Hund, and Z. Aziz. 2009. Generalized clustering via kernel embeddings. *Springer-Verlag* (2009).

[7] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu. 2019. Large-scale Multi-view Subspace Clustering in Linear Time. *arXiv* (2019).

[8] Abhishek Kumar and Hal Daumé. 2011. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 393–400.

[9] Abhishek Kumar, Piyush Rai, and Hal Daume. 2011. Co-regularized multi-view spectral clustering. *Advances in neural information processing systems* 24 (2011), 1413–1421.

[10] Miaomiao Li, Xinwang Liu, Lei Wang, Yong Dou, Jianping Yin, and En Zhu. 2016. Multiple kernel clustering with local kernel alignment maximization. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 1704–1710.

[11] Miaomiao Li, Xinwang Liu, Lei Wang, Yong Dou, Jianping Yin, and En Zhu. 2016. Multiple Kernel Clustering with Local Kernel Alignment Maximization. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) *(IJCAI'16)*. AAAI Press, 1704–1710.

[12] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. 2020. Multiview Clustering: A Scalable and Parameter-Free Bipartite Graph Fusion Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 1 (2020), 330–344.

[13] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. 2015. Large-Scale Multi-View Spectral Clustering via Bipartite Graph. (2015).

[14] Zhenglai Li, Chang Tang, Xinwang Liu, Xiao Zheng, Guanghui Yue, Wei Zhang, and En Zhu. 2021. Consensus Graph Learning for Multi-view Clustering. *IEEE Transactions on Multimedia* (2021).

[15] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. 2021. COMPLETER: Incomplete multi-view clustering via contrastive prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11174–11183.

[16] Jiyuan Liu, Xinwang Liu, Jian Xiong, Qing Liao, Sihang Zhou, Siwei Wang, and Yuexiang Yang. 2020. Optimal Neighborhood Multiple Kernel Clustering with Adaptive Local Kernels. *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1. https://doi.org/10.1109/TKDE.2020.3014104

[17] Suyuan Liu, Siwei Wang, Pei Zhang, Kai Xu, Xinwang Liu, Changwang Zhang, and Feng Gao. 2022. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7576–7584.

[18] Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, and En Zhu. 2016. Multiple Kernel k-Means Clustering with Matrix-Induced Regularization. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 1888–1894.

[19] Xinwang Liu, Li Liu, Qing Liao, Siwei Wang, Yi Zhang, Wenxuan Tu, Chang Tang, Jiyuan Liu, and En Zhu. 2021. One pass late fusion multi-view clustering. In *International Conference on Machine Learning*. PMLR, 6850–6859.

[20] Xinwang Liu, Lei Wang, Jianping Yin, En Zhu, and Jian Zhang. 2013. An efficient approach to integrating radius information into multiple kernel learning. *IEEE transactions on cybernetics* 43, 2 (2013), 557–569.

[21] Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, and Han Li. 2017. Optimal Neighborhood Kernel Clustering with Multiple Kernels. In *AAAI2017*.

[22] X. Liu, E. Zhu, J. Liu, T. Hospedales, Y. Wang, and M. Wang. 2020. SimpleMKKM: Simple Multiple Kernel K-means. (2020).

[23] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. 2022. Deep Graph Clustering via Dual Correlation Reduction.

[24] Yue Liu, Xihong Yang, Sihang Zhou, and Xinwang Liu. 2022. Simple Contrastive Graph Clustering. *arXiv preprint arXiv:2205.07865* (2022).

[25] Yue Liu, Sihang Zhou, Xinwang Liu, Wenxuan Tu, and Xihong Yang. 2022. Improved Dual Correlation Reduction Network. *arXiv preprint arXiv:2202.12533* (2022).

[26] Feiping Nie, Jing Li, and Xuelong Li. 2017. Self-Weighted Multiview Clustering with Multiple Graphs. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press, 2564–2570.

[27] Feiping Nie, Lai Tian, and Xuelong Li. 2018. Multiview clustering via adaptively weighted procrustes. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2022–2030.

[28] Xi Peng, Jiashi Feng, Shijie Xiao, Wei-Yun Yau, Joey Tianyi Zhou, and Songfan Yang. 2018. Structured autoencoders for subspace clustering. *IEEE Transactions on Image Processing* 27, 10 (2018), 5076–5086.

[29] Leigang Qu, Meng Liu, Da Cao, Liqiang Nie, and Qi Tian. 2020. *Context-Aware Multi-View Summarization Network for Image-Text Matching*. Association for Computing Machinery, New York, NY, USA, 1047–1055. https://doi.org/10.1145/3394171.3413961

[30] Zhenwen Ren and Quansen Sun. 2020. Simultaneous global and local graph structure preserving for multiple kernel clustering. *IEEE transactions on neural networks and learning systems* 32, 5 (2020), 1839–1851.

[31] Mengjing Sun, Siwei Wang, Pei Zhang, Xinwang Liu, Sihang Zhou, Xifeng Guo, and En Zhu. 2021. Projective Multiple Kernel Subspace Clustering. *IEEE Transactions on Multimedia* (2021).

[32] Mengjing Sun, Pei Zhang, Siwei Wang, Sihang Zhou, Wenxuan Tu, Xinwang Liu, En Zhu, and Changjian Wang. 2021. Scalable multi-view subspace clustering with unified anchors. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3528–3536.

[33] Chang Tang, Xinzhong Zhu, Xinwang Liu, Miaomiao Li, Pichao Wang, Changqing Zhang, and Lizhe Wang. 2018. Learning a joint affinity graph for multiview subspace clustering. *IEEE Transactions on Multimedia* 21, 7 (2018), 1724–1736.

[34] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9 (2008).

[35] Boyue Wang, Yongli Hu, Junbin Gao, Yanfeng Sun, Fujiao Ju, and Baocai Yin. 2020. Learning Adaptive Neighborhood Graph on Grassmann Manifolds for Video/Image-Set Subspace Clustering. *IEEE Transactions on Multimedia* (2020).

[36] Huibing Wang, Yang Wang, Zhao Zhang, Xianping Fu, Li Zhuo, Mingliang Xu, and Meng Wang. 2020. Kernelized multiview subspace analysis by self-weighted learning. *IEEE Transactions on Multimedia* (2020).

[37] Siwei Wang, Xinwang Liu, En Zhu, Chang Tang, Jiyuan Liu, Jingtao Hu, Jingyuan Xia, and Jianping Yin. 2019. Multi-view Clustering via Late Fusion Alignment Maximization.. In *IJCAI*. 3778–3784.

[38] Siwei Wang, Xinwang Liu, Xinzhong Zhu, Pei Zhang, Yi Zhang, Feng Gao, and En Zhu. 2022. Fast Parameter-Free Multi-View Subspace Clustering With Consensus Anchor Guidance. *IEEE Trans. Image Process.* 31 (2022), 556–568. https://doi.org/10.1109/TIP.2021.3131941

[39] Mouxing Yang, Yunfan Li, Zhenyu Huang, Zitao Liu, Peng Hu, and Xi Peng. 2021. Partially view-aligned representation learning with noise-robust contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1134–1143.

[40] Xiwen Yao, Junwei Han, Dingwen Zhang, and Feiping Nie. 2017. Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering. *IEEE Transactions on Image Processing* 26, 7 (2017), 3196–3209.

[41] Junning Zhang, Qunxing Su, Bo Tang, Cheng Wang, and Yining Li. 2021. DPSNet: Multitask Learning Using Geometry Reasoning for Scene Depth and Semantics. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–12. https://doi.org/10.1109/TNNLS.2021.3107362

[42] Tiejian Zhang, Xinwang Liu, Lei Gong, Siwei Wang, Xin Niu, and Li Shen. 2021. Late Fusion Multiple Kernel Clustering with Local Kernel Alignment Maximization. *IEEE Transactions on Multimedia* (2021), 1–1. https://doi.org/10.1109/TMM.2021.3136094

[43] Yi Zhang, Xinwang Liu, Jiyuan Liu, Sisi Dai, Changwang Zhang, Kai Xu, and En Zhu. 2022. Fusion Multiple Kernel K-means. (2022).

[44] Yi Zhang, Xinwang Liu, Siwei Wang, Jiyuan Liu, Sisi Dai, and En Zhu. 2021. One-Stage Incomplete Multi-view Clustering via Late Fusion. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2717–2725.

[45] Sihang Zhou, Xinwang Liu, Miaomiao Li, En Zhu, Li Liu, Changwang Zhang, and Jianping Yin. 2020. Multiple Kernel Clustering With Neighbor-Kernel Subspace Segmentation. *IEEE Transactions on Neural Networks and Learning Systems* 31, 4 (2020), 1351–1362. https://doi.org/10.1109/TNNLS.2019.2919900

[46] Sihang Zhou, Qiyuan Ou, Xinwang Liu, Siqi Wang, Luyan Liu, Siwei Wang, En Zhu, Jianping Yin, and Xin Xu. 2021. Multiple Kernel Clustering With Compressed Subspace Alignment. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–12. https://doi.org/10.1109/TNNLS.2021.3093426

[47] Sihang Zhou, En Zhu, Xinwang Liu, Tianming Zheng, Qiang Liu, Jingyuan Xia, and Jianping Yin. 2020. Subspace segmentation-based robust multiple kernel clustering. *Information Fusion* 53 (2020), 145 – 154.