DVSAI: Diverse View-Shared Anchors Based Incomplete Multi-View Clustering

Shengju Yu[1], **Siwei Wang**[2*], **Pei Zhang**[1], **Miao Wang**[2*], **Ziming Wang**[3],
Zhe Liu[4], **Liming Fang**[5], **En Zhu**[1], **Xinwang Liu**[1*]
[1]School of Computer, National University of Defense Technology, Changsha, 410073, China
[2]Intelligent Game and Decision Lab, Beijing, 100071, China
[3]China Academy of Aerospace Science and Innovation, Beijing, 100176, China
[4]Zhejiang Lab, Hangzhou, 311500, China
[5]Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China
yu-shengju@foxmail.com, {wangsiwei13, mercury.miao, xinwangliu}@nudt.edu.cn

Abstract

In numerous real-world applications, it is quite common that sample information is partially available for some views due to machine breakdown or sensor failure, causing the problem of incomplete multi-view clustering (IMVC). While several IMVC approaches using view-shared anchors have successfully achieved pleasing performance improvement, (1) they generally construct anchors with only one dimension, which could deteriorate the multi-view diversity, bringing about serious information loss; (2) the constructed anchors are typically with a single size, which could not sufficiently characterize the distribution of the whole samples, leading to limited clustering performance. For generating view-shared anchors with multi-dimension and multi-size for IMVC, we design a novel framework called Diverse View-Shared Anchors based Incomplete multi-view clustering (DVSAI). Concretely, we associate each partial view with several potential spaces. In each space, we enable anchors to communicate among views and generate the view-shared anchors with space-specific dimension and size. Consequently, spaces with various scales make the generated view-shared anchors enjoy diverse dimensions and sizes. Subsequently, we devise an integration scheme with linear computational and memory expenditures to integrate the outputted multi-scale unified anchor graphs such that running spectral algorithm generates the spectral embedding. Afterwards, we theoretically demonstrate that DVSAI owns linear time and space costs, thus well-suited for tackling large-size datasets. Finally, comprehensive experiments confirm the effectiveness and advantages of DVSAI.

Introduction

As a technology for effectively grouping multi-view data without requiring any priori labels, multi-view clustering has drawn widespread concerns recently (Li et al. 2022b; Wang et al. 2022a; Yu et al. 2023a). However, their performance is heavily dependent on the completeness of sample information. In many situations, the sample information could be missing in some views, bringing about partial multi-view data (Yang et al. 2021a; Yu et al. 2023b; Xia et al. 2022a). For example, in medical data, patients typically choose to do some types of examinations rather than all types of them; In web-page data, some webs may include audio, video and hyperlink simultaneously, and yet others may have one or two types (Zhang et al. 2019; Liang et al. 2023; Fu et al. 2023). The unpaired sample absence in respective view causes it to be quite difficult to capture the consensus representation, resulting in serious performance degradation. This induces the problem of incomplete multi-view clustering (IMVC).

For effectively tackling IMVC issues, numerous algorithms have been devised, like (Li, Jiang, and Zhou 2014; Fang et al. 2020; Wang et al. 2019a; Yang et al. 2021b; Wen et al. 2021). Li, Jiang, and Zhou (2014) adopt NMF scheme to handle partial views and seek to build the latent subspace such that similar samples reside tightly. Wang et al. (2019a) build up the connection between IMVC and perturbation theory, and learn the consensus characteristics by minimizing the perturbation risk. These methods receive the desirable clustering results, however, they usually take cubic time and/or square memory overheads. This causes them unable to tackle large-size datasets, greatly impeding the model's scalability. Recently, anchor strategy (Kang et al. 2020; Qiang et al. 2021; Xia et al. 2022b), a promising technology for decreasing the overheads, has captured increasing interest. Instead of constructing the relationship between all samples, it first chooses a small number of landmarks as anchors and then generates the relationship between anchors and all samples. Accordingly, the graph with size $n \times n$ is reduced to one with $m \times n$ where $m \ll n$.

IMVC algorithms based on anchors have made extraordinary progress in increasing both the efficiency and the clustering results (Wang et al. 2022b; Li et al. 2022a; Liu et al. 2022b), nevertheless, the view-shared anchors constructed by existing methods are generally with only one dimension, which could deteriorate the diversity between views, leading to severe information loss. This is mainly because every view normally has exclusive features and one common dimension may not be competent to characterize all views. In addition to this, the constructed view-shared anchors are typically with a single size, which could be unfavorable for deeply exploiting heterogeneous representations since only one size could not adequately reflect the actual distribution of the whole samples, resulting in less discriminative anchors. The clustering performance is also largely dependent on the choice of dimension and size of anchors, which to some extent limits the flexibility of the model.

[*] Corresponding authors

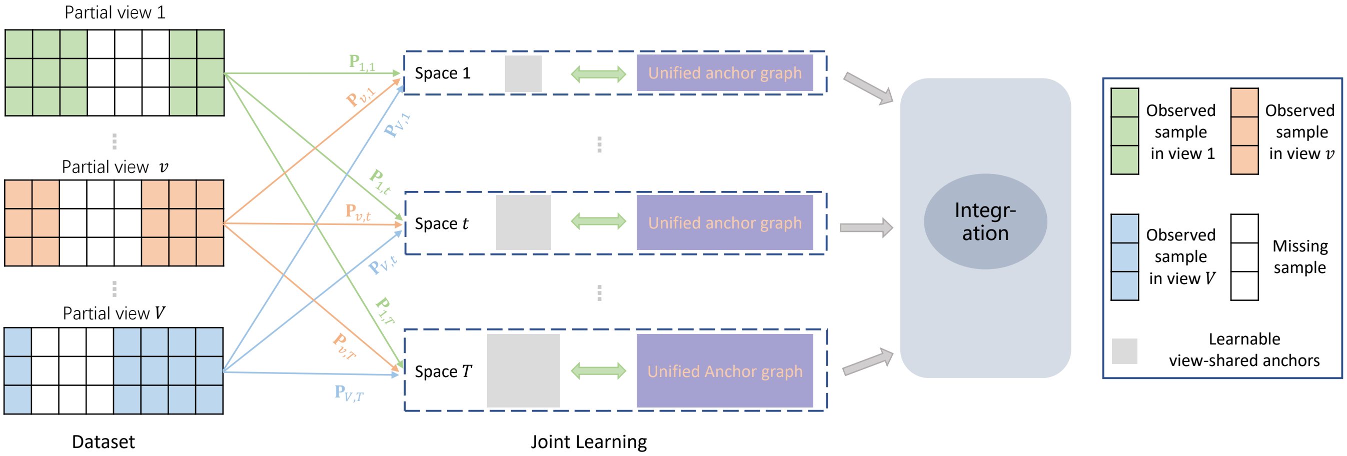


Figure 1: Illustration of the devised DVSAI. We utilize matrices $\{\mathbf{P}_{v,t}\}_{t=1}^{T}$ to build up T potential spaces for every partial view v. In each space, we enable anchors to communicate among views, and jointly learn across views to produce unified anchor graph and view-shared anchors with space-specific dimension and size. In such ways, spaces with diverse scales make the learned view-shared anchors enjoy multi-dimension and multi-size. Subsequently, these unified anchor graphs with various scales generated in all potential spaces are integrated together within linear computation and memory expenditures.

For overcoming these limitations, we develop an innovative framework named DVSAI, as described in Fig. 1. Specifically, we integrate anchor generation and anchor graph construction into a joint learning IMVC framework, and associate each partial view with T potential spaces. In each space, we enable anchors to communicate among views and learn across views to generate the view-shared anchors with space-specific dimension and size. Accordingly, spaces enjoying various scales guarantee the generated view-shared anchors to be with both multi-dimension and multi-size. Subsequently, it is essential to integrate the outputted unified anchor graphs with diverse scales together so as to generate the spectral embedding. Nevertheless, how to integrate graphs with different scales has not been well researched. For alleviating this problem, we develop an integration scheme enjoying linear computational and memory expenditures by avoiding building up the complete affinity of each anchor graph. Afterwards, for optimizing the resultant objective function, we devise a four variable alternative approach. In conjunction with it, our DVSAI is proven to be with linear time and space complexities, thus winning the ability to handle large-size datasets. By conducting comprehensive experiments on six common multi-view datasets, DVSAI's effectiveness and advantages are both verified. Our main contributions are summarized as follows:

- We generate view-shared anchors with multi-dimension and multi-size for IMVC so as to aggregate heterogeneous features more sufficiently.
- We offer an integration scheme with linear computational and storage expenditures to combine together the anchor graphs with diverse scales.
- We design an iterative method involving four variables, which owns linear time and space complexities, to optimize the resultant IMVC objective function. Reasonable experiments are implemented to validate our advantages.

Related Work

View-shared Anchor Clustering

Anchor strategy selects a few number of landmarks to approximately represent the overall samples, and then builds up the relation with original samples to generate graphs with small size (Kang et al. 2020; Li et al. 2023; Chen et al. 2022; Li et al. 2022c). The basic framework is defined as:

$$\min_{\mathbf{G}_v \geq 0, \mathbf{G}_v^\top \mathbf{1} = \mathbf{1}} \sum_{v=1}^{V} \left\| \mathbf{X}_v - \mathbf{A}_v \mathbf{G}_v \right\|_F^2 + \beta \left\| \mathbf{G}_v \right\|_F^2 , \quad (1)$$

where $\mathbf{X}_v \in \mathbb{R}^{d_v \times n}$, $\mathbf{A}_v \in \mathbb{R}^{d_v \times m}$, $\mathbf{G}_v \in \mathbb{R}^{m \times n}$ represent the sample matrix, anchors and anchor graph of the v-th view. Usually, the number of anchors, m, is much smaller than the number of samples, n. On this foundation, the framework using view-shared anchors can be formulated as:

$$\min_{\mathbf{G} \geq 0, \mathbf{G}^\top \mathbf{1} = \mathbf{1}} \sum_{v=1}^{V} \left\| \mathbf{W}_v \mathbf{X}_v - \mathbf{A}\mathbf{G} \right\|_F^2 + \beta \left\| \mathbf{G} \right\|_F^2 , \quad (2)$$

where $\mathbf{W}_v \in \mathbb{R}^{l \times d_v}$, $\mathbf{A} \in \mathbb{R}^{l \times m}$, $\mathbf{G} \in \mathbb{R}^{m \times n}$ represent the compression matrix for view v, view-shared anchors and unified anchor graph respectively. l and m denote the dimension and size of view-shared anchors. Following this line, Sun et al. (2021) utilize learning strategy instead of random sampling to produce anchors. Wang et al. (2022c) offer a framework without additional parameters, and generate the unified anchor graph with low-rank property. Liu et al. (2022c) directly output the clustering labels by constructing the unified anchor graph with k-connected components. Wang et al. (2022b) provide a flexible fusion framework to tackle any view incompleteness. Despite impressive results, the view-shared anchors generated by current methods are generally with only one dimension and a single size, which could deteriorate the data diversity and not well characterize the whole samples, limiting the clustering performance.

Formulation

Framework

We first measure the incompleteness of multi-view data via the given indicator vectors $\{w_v \in \mathbb{R}^{n_v}\}_{v=1}^{V}$, where n_v represents the number of the observed samples in view v. Specifically, for each partial view, we associate an index matrix $\mathbf{S}_v$ consisting of element $[\mathbf{S}_v]_{i,j}$,

$$[\mathbf{S}_v]_{i,j} = \begin{cases} 1 & \text{if } [w_v]_j = i \;\; \forall j = 1, 2, \cdots, n_v. \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Then, $\mathbf{X}_v\mathbf{S}_v \in \mathbb{R}^{d_v \times n_v}$ represents the samples available in view v. Afterwards, we make view-shared anchors learnable and jointly optimize view-shared anchors and unified anchor graph in one framework:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{G}} \sum_{v=1}^{V} \left\|\mathbf{X}_v\mathbf{S}_v - \mathbf{P}_v\mathbf{A}\mathbf{G}\mathbf{S}_v\right\|_F^2 + \beta \left\|\mathbf{G}\right\|_F^2 \\ \text{s.t. } \mathbf{A}^\top\mathbf{A} = \mathbf{I}_m, \mathbf{G} \geq 0, \mathbf{G}^\top\mathbf{1} = \mathbf{1}. \end{aligned} \tag{4}$$

Further, to generate the view-shared anchors with multi-dimension and multi-size, we assign several potential spaces with diverse scales to each partial view, and learn across views in all spaces so as to make view-shared anchors enjoy space-specific dimension and size:

$$\begin{aligned} \min_{\mathbf{P}_{v,t},\mathbf{A}_t,\mathbf{G}_t} \sum_{v=1}^{V}\sum_{t=1}^{T} \left\|\mathbf{X}_v\mathbf{S}_v - \mathbf{P}_{v,t}\mathbf{A}_t\mathbf{G}_t\mathbf{S}_v\right\|_F^2 + \beta \left\|\mathbf{G}_t\right\|_F^2 \\ \text{s.t. } \mathbf{P}_{v,t}^\top\mathbf{P}_{v,t} = \mathbf{I}_{l_t}, \mathbf{A}_t^\top\mathbf{A}_t = \mathbf{I}_{m_t}, \mathbf{G}_t \geq 0, \mathbf{G}_t^\top\mathbf{1} = \mathbf{1}, \end{aligned} \tag{5}$$

where $\{\mathbf{P}_{v,t} \in \mathbb{R}^{d_v \times l_t}\}_{t=1}^{T}$ represent T potential spaces introduced for view v, and $\mathbf{A}_t \in \mathbb{R}^{l_t \times m_t}$ represents the produced view-shared anchors in space t. l_t and m_t represent the dimension and size of view-shared anchors in space t respectively. The orthogonal constraints guarantee the learned view-shared anchors and potential spaces to be more discriminative. $\mathbf{P}_{v,t}$, $\mathbf{A}_t$ and $\mathbf{G}_t$ are iteratively updated into one framework, which makes them boost each other.

Additionally, considering that $\mathbf{A}_t$ could have different contributions in spaces $\{\mathbf{P}_{v,t}\}_{v=1}^{V}$, we offer a group of weight variables for each space to adaptively balance the contributions of view-shared anchors. Finally, we design the DVSAI framework as

$$\begin{aligned} \min_{\mathbf{P}_{v,t},\mathbf{A}_t,\mathbf{G}_t,\boldsymbol{\alpha}} \sum_{v=1}^{V}\sum_{t=1}^{T} \alpha_{v,t}^2 \Big(\left\|\mathbf{X}_v\mathbf{S}_v - \mathbf{P}_{v,t}\mathbf{A}_t\mathbf{G}_t\mathbf{S}_v\right\|_F^2 + \\ \beta \left\|\mathbf{G}_t\right\|_F^2 \Big) + \gamma \left\|\boldsymbol{\alpha}\right\|_F^2 \\ \text{s.t. } \mathbf{P}_{v,t}^\top\mathbf{P}_{v,t} = \mathbf{I}_{l_t}, \mathbf{A}_t^\top\mathbf{A}_t = \mathbf{I}_{m_t}, \\ \mathbf{G}_t \geq 0, \mathbf{G}_t^\top\mathbf{1} = \mathbf{1}, \boldsymbol{\alpha} \geq 0, \boldsymbol{\alpha}^\top\mathbf{1} = \mathbf{1}. \end{aligned} \tag{6}$$

In this manner, besides successfully generating view-shared anchors with diverse dimensions and sizes for IMVC, it is also able to automatically adjust the importance of view-shared anchors so as to better characterize partial multi-view data, increasing the model's expression. Moreover, the generated multi-scale unified anchor graphs $\{\mathbf{G}_t \in \mathbb{R}^{m_t \times n]}\}_{t=1}^{T}$ have more strong ability to capture the similarity relationship with samples, and thereby bring inspiring clustering performance improvement.

Solver

We solve Eq. (6) by the following four steps alternatively.

Step-1: Optimize $\{\mathbf{P}_{v,t}\}_{v=1,t=1}^{V\;\;,T}$. Fixing $\{\mathbf{A}_t\}$, $\{\mathbf{G}_t\}$ and $\boldsymbol{\alpha}$, the optimization w.r.t. each $\mathbf{P}_{v,t}$ in Eq. (6) is

$$\min_{\mathbf{P}_{v,t}} \left\|\mathbf{X}_v\mathbf{S}_v - \mathbf{P}_{v,t}\mathbf{A}_t\mathbf{G}_t\mathbf{S}_v\right\|_F^2 \text{ s.t. } \mathbf{P}_{v,t}^\top\mathbf{P}_{v,t} = \mathbf{I}_{l_t}. \tag{7}$$

After expanding F-norm and deleting the irrelevant terms, Eq. (7) equivalently becomes

$$\max_{\mathbf{P}_{v,t}} \operatorname{Tr}\left(\mathbf{P}_{v,t}^\top\mathbf{X}_v\mathbf{S}_v\mathbf{S}_v^\top\mathbf{G}_t^\top\mathbf{A}_t^\top\right) \text{ s.t. } \mathbf{P}_{v,t}^\top\mathbf{P}_{v,t} = \mathbf{I}_{l_t}. \tag{8}$$

Denote the SVD of $\mathbf{X}_v\mathbf{S}_v\mathbf{S}_v^\top\mathbf{G}_t^\top\mathbf{A}_t^\top$ as $\mathbf{U}\Sigma\mathbf{V}^\top$. According to the theorem in (Wang et al. 2019b), we have that the optimal $\mathbf{P}_{v,t}$ is $\mathbf{U}\mathbf{V}^\top$.

Considering that the multiplication of $\mathbf{X}_v$ and $\mathbf{S}_v\mathbf{S}_v^\top$ costs $\mathcal{O}(d_v n^2)$ complexity, directly performing SVD is relatively time-consuming. In particular, when the number of samples n is larger, this will bring about expensive time overhead, causing the inability to handling large-size tasks.

For improving the efficiency, we notice that

$$\mathbf{S}_v\mathbf{S}_v^\top = \operatorname{diag}(f_{v,1}, f_{v,2}, \cdots, f_{v,n}), \tag{9}$$

where $f_{v,i} = \sum_{j=1}^{n_v}[\mathbf{S}_v]_{i,j}$, $i = 1, \cdots, n$. That is, the diagonal element is the row sum of $\mathbf{S}_v$. Hence, we have

$$\mathbf{X_v}\mathbf{S}_v\mathbf{S}_v^\top = \Big[f_{v,1}[\mathbf{X_v}]_{:,1}, f_{v,2}[\mathbf{X_v}]_{:,2}, \cdots, f_{v,n}[\mathbf{X_v}]_{:,n}\Big]. \tag{10}$$

Further, Eq. (10) can be reformulated as

$$\mathbf{X_v}\mathbf{S}_v\mathbf{S}_v^\top = \mathbf{X_v} \odot \mathbf{F}_v, \tag{11}$$

where $\mathbf{F}_v = \mathbf{1}_{d_v \times 1} \cdot [f_{v,1}, f_{v,2}, \cdots, f_{v,n}]_{1 \times n}$. Hadamard product $\mathbf{X_v} \odot \mathbf{F}_v$ costs $\mathcal{O}(d_v n)$ complexity. Therefore, we run SVD on $(\mathbf{X_v} \odot \mathbf{F}_v)\mathbf{G}_t^\top\mathbf{A}_t^\top$ to acquire the optimal $\mathbf{P}_{v,t}$.

Step-2: Optimize $\{\mathbf{A}_t\}_{t=1}^{T}$. Fixing $\{\mathbf{P}_{v,t}\}$, $\{\mathbf{G}_t\}$ and $\boldsymbol{\alpha}$, the optimization w.r.t. each $\mathbf{A}_t$ in Eq. (6) is

$$\min_{\mathbf{A}_t^\top\mathbf{A}_t = \mathbf{I}_{m_t}} \sum_{v=1}^{V} \alpha_{v,t}^2 \left\|\mathbf{X}_v\mathbf{S}_v - \mathbf{P}_{v,t}\mathbf{A}_t\mathbf{G}_t\mathbf{S}_v\right\|_F^2. \tag{12}$$

Deleting irrelevant items, Eq. (12) is transformed as

$$\max_{\mathbf{A}_t^\top\mathbf{A}_t = \mathbf{I}_{m_t}} \operatorname{Tr}\left(\mathbf{A}_t^\top\left(\sum_{v=1}^{V}\alpha_{v,t}^2\mathbf{P}_{v,t}^\top\left(\mathbf{X}_v \odot \mathbf{F}_v\right)\mathbf{G}_t^\top\right)\right) \tag{13}$$

The optimal $\mathbf{A}_t$ is $\mathbf{U}\mathbf{V}^\top$ where $\mathbf{U}$ and $\mathbf{V}$ are the SVD matrices of the term $\sum_{v=1}^{V}\alpha_{v,t}^2\mathbf{P}_{v,t}^\top\left(\mathbf{X}_v \odot \mathbf{F}_v\right)\mathbf{G}_t^\top$ respectively.

Step-3: Optimize $\{\mathbf{G}_t\}_{t=1}^{T}$. Fixing $\{\mathbf{P}_{v,t}\}$, $\{\mathbf{A}_t\}$ and $\boldsymbol{\alpha}$, the optimization w.r.t. each $\mathbf{G}_t$ in Eq. (6) is

$$\begin{aligned} \min_{\mathbf{G}_t} \sum_{v=1}^{V} \alpha_{v,t}^2 \left(\left\|\mathbf{X}_v\mathbf{S}_v - \mathbf{P}_{v,t}\mathbf{A}_t\mathbf{G}_t\mathbf{S}_v\right\|_F^2 + \beta \left\|\mathbf{G}_t\right\|_F^2 \right) \\ \text{s.t. } \mathbf{G}_t \geq 0, \mathbf{G}_t^\top\mathbf{1} = \mathbf{1}. \end{aligned} \tag{14}$$

Given that the feasible region is aimed at the column of $\mathbf{G}_t$, hence, we can update $\mathbf{G}_t$ by column:

$$\begin{aligned} \min_{[\mathbf{G}_t]_{:,j}} & [\mathbf{G}_t^\top]_{j,:}[\mathbf{G}_t]_{:,j}\left(\sum_{v=1}^{V}\alpha_{v,t}^2\left(f_{v,j}+\beta\right)\right) \\ & -2[\mathbf{G}_t^\top]_{j,:}\left(\sum_{v=1}^{V}\alpha_{v,t}^2\mathbf{A}_t^\top\mathbf{P}_{v,t}^\top\mathbf{X}_v\odot\mathbf{F}_v\right)_{:,j} \\ & \text{s.t. } [\mathbf{G}_t]_{:,j}\geq 0, [\mathbf{G}_t]_{:,j}^\top\mathbf{1}=1. \end{aligned} \tag{15}$$

Note that the term $\sum_{v=1}^{V}\alpha_{v,t}^2\left(f_{v,j}+\beta\right)$ is a scalar, and hence we can equivalently rewrite Eq. (15) as

$$\min_{[\mathbf{G}_t]_{:,j}} \|[\mathbf{G}_t]_{:,j}-q_j\|_F^2 \text{ s.t. } [\mathbf{G}_t]_{:,j}\geq 0, [\mathbf{G}_t]_{:,j}^\top\mathbf{1}=1, \tag{16}$$

where $q_j = \frac{\left(\sum_{v=1}^{V}\alpha_{v,t}^2\mathbf{A}_t^\top\mathbf{P}_{v,t}^\top\mathbf{X}_v\odot\mathbf{F}_v\right)_{:,j}}{\sum_{v=1}^{V}\alpha_{v,t}^2(f_{v,j}+\beta)}$. For solving Eq. (16), we formulate the Lagrangian function as

$$\begin{aligned} \mathcal{L}\left([\mathbf{G}_t]_{:,j},\Phi,\varphi\right)= & \frac{1}{2}\|[\mathbf{G}_t]_{:,j}-q_j\|_F^2-\Phi^\top[\mathbf{G}_t]_{:,j} \\ & -\varphi\left([\mathbf{G}_t]_{:,j}^\top\mathbf{1}-1\right). \end{aligned} \tag{17}$$

Using KKT conditions, we can get

$$[\mathbf{G}_t]_{:,j}-q_j-\Phi-\varphi\mathbf{1}=0,\ \Phi^\top[\mathbf{G}_t]_{:,j}=0. \tag{18}$$

Further, in conjunction with $[\mathbf{G}_t]_{:,j}^\top\mathbf{1}=1$, we can get

$$[\mathbf{G}_t]_{:,j}=\max\left(q_j+\varphi\mathbf{1},0\right)\ ,\varphi=(1-q_j^\top\mathbf{1})/m_t. \tag{19}$$

Step-4: Optimize $\boldsymbol{\alpha}$. Fixing $\{\mathbf{P}_{v,t}\}$, $\{\mathbf{A}_t\}$ and $\{\mathbf{G}_t\}$, the optimization w.r.t. $\boldsymbol{\alpha}$ in Eq. (6) is

$$\min_{\boldsymbol{\alpha}}\sum_{v=1}^{V}\sum_{t=1}^{T}\alpha_{v,t}^2 h_{v,t} \text{ s.t. } \boldsymbol{\alpha}\geq 0, \boldsymbol{\alpha}^\top\mathbf{1}=\mathbf{1}, \tag{20}$$

where $h_{v,t}=\|\mathbf{X}_v\mathbf{S}_v-\mathbf{P}_{v,t}\mathbf{A}_t\mathbf{G}_t\mathbf{S}_v\|_F^2+\beta\|\mathbf{G}_t\|_F^2+\gamma$. For any given v and t, $h_{v,t}$ is a constant. Thus, we have

$$\alpha_{v,t}=\frac{\frac{1}{h_{v,t}}}{\sum_{v=1}^{V}\frac{1}{h_{v,t}}}. \tag{21}$$

Integration

We need to integrate the generated multi-scale unified anchor graphs $\{\mathbf{G}_t\in\mathbb{R}^{m_t\times n}\}_{t=1}^{T}$ together such that running spectral algorithm generates the spectral embedding.

According to Ref. (Chen and Cai 2011), we have that the complete affinity $\mathbf{O}_t$ of $\mathbf{G}_t$ can be recovered by setting $\mathbf{O}_t=\widehat{\mathbf{G}}_t^\top\widehat{\mathbf{G}}_t$, where $\widehat{\mathbf{G}}_t=\mathbf{M}_t^{-1/2}\mathbf{G}_t$. The matrix $\mathbf{M}_t\in\mathbb{R}^{m_t\times m_t}$ is diagonal, and the element $[\mathbf{M}_t]_{i,i}$ is set as $\sum_{j=1}^{n}[\mathbf{G}_t]_{i,j}$, $i=1,2,\cdots,m_t$. Therefore, the affinity between all views can be calculated as

$$\mathbf{O}=\sum_{v=1}^{V}\sum_{t=1}^{T}\alpha_{v,t}^2\mathbf{O}_t. \tag{22}$$

Subsequently, the spectral embedding can be acquired by performing spectral algorithm on $\mathbf{O}$.

Algorithm 1: DVSAI

Input:Partial views $\{\mathbf{X}_v\}_{v=1}^{V}$, index matrices $\{\mathbf{S}_v\}_{v=1}^{V}$, parameters β, γ;
Initialize: $\{\mathbf{P}_{v,t}\}_{v=1,t=1}^{V\ \ ,T}$, $\{\mathbf{A}_t\}_{t=1}^{T}$, $\{\mathbf{G}_t\}_{t=1}^{T}$, $\boldsymbol{\alpha}$;

```
repeat
    Solving every $\mathbf{P}_{v,t}$ by Eq. (8);
    Solving every $\mathbf{A}_t$ by Eq. (13);
    Solving every $\mathbf{G}_t$ by Eq. (15);
    Solving $\boldsymbol{\alpha}$ by Eq. (21);
until convergent
Integrating $\{\mathbf{G}_t\}_{t=1}^{T}$ by Eq. (24);
Generating $\mathbf{U}$ by running SVD on $\mathbf{L}$;
```

Output: Clustering indicators by running k-means on $\mathbf{U}$;

Considering that, however, $\mathbf{O}$ is with size $n\times n$, it will cost $\mathcal{O}(n^2)$ memory expenditure and $\mathcal{O}(n^3)$ computational expenditure when producing the spectral embedding.

To decrease the complexity, we observe that

$$\mathbf{O}=\mathbf{L}\mathbf{L}^\top, \tag{23}$$

where

$$\mathbf{L}=\left[\alpha_{1,1}\widehat{\mathbf{G}}_1^\top,\cdots,\alpha_{1,T}\widehat{\mathbf{G}}_T^\top,\cdots,\alpha_{V,1}\widehat{\mathbf{G}}_1^\top,\cdots,\alpha_{V,T}\widehat{\mathbf{G}}_T^\top\right]. \tag{24}$$

Denote the SVD of matrix $\mathbf{L}\in\mathbb{R}^{n\times(\sum_{t=1}^{T}m_t)V}$ as $\mathbf{U\Sigma V}^\top$. We have

$$\mathbf{O}=\mathbf{U\Sigma^2U}^\top, \tag{25}$$

which illustrates that the matrix consisting of the eigenvectors of $\mathbf{O}$ equals to $\mathbf{U}$. Therefore, $\mathbf{U}$ can be regarded as the spectral embedding, and we can run k-means on it to produce the discrete clustering indicators. Due to $V\sum_{t=1}^{T}m_t \ll n$, integrating $\{\mathbf{G}_t\}_{t=1}^{T}$ as $\mathbf{L}$ takes $\mathcal{O}(n)$ space cost, and performing SVD on $\mathbf{L}$ takes $\mathcal{O}(n)$ time cost. As a result, the space and computational overheads of our integration scheme are both linear w.r.t. n. The pipeline of the devised DVSAI algorithm is presented in Algorithm 1.

Dataset	Samples	Feature Dimension
Caltech101-7	1474	48, 40, 254, 1984, 512, 928
Caltech101-20	2386	48, 40, 254, 1984, 512, 928
CCV	6773	20, 20, 20
SUNRGBD	10335	4096, 4096
NUSWIDEOBJ	30000	65, 226, 145, 74, 129
YoutubeFace_Sel	101499	64, 512, 64, 647, 838

Table 1: Details of six benchmark datasets.

Complexity

Computational Complexity The time overhead is mainly composed of updating the variables $\{\mathbf{P}_{v,t}\}$, $\{\mathbf{A}_t\}$, $\{\mathbf{G}_t\}$ and $\boldsymbol{\alpha}$. When updating each $\mathbf{P}_{v,t}$, calculating $(\mathbf{X_v}\odot\mathbf{F}_v)\mathbf{G}_t^\top\mathbf{A}_t^\top$ and running SVD on it take $\mathcal{O}(d_vn+d_vnm_t+d_vm_tl_t)$ and $\mathcal{O}(d_vl_t^2)$ respectively. Hence, updating $\{\mathbf{P}_{v,t}\}$ takes $\mathcal{O}(dn\sum_{t=1}^{T}(m_t))$. When updating each $\mathbf{A}_t$, calculating $\sum_{v=1}^{V}\alpha_{v,t}^2\mathbf{P}_{v,t}^\top\left(\mathbf{X}_v\odot\mathbf{F}_v\right)\mathbf{G}_t^\top$ and running SVD on it take

Data	Methods	0.1			0.3			0.5			0.7		
		ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
Caltech101-7	BSV	54.90	30.14	72.57	54.57	20.05	68.45	54.66	10.26	61.98	54.57	14.08	64.85
	MIC	44.02	31.80	78.09	40.70	28.88	74.24	34.50	25.10	70.72	34.95	21.12	68.85
	MKKM-IK	34.95	30.11	77.71	34.64	29.41	77.84	41.02	35.45	77.17	39.36	33.12	77.00
	UEAF	41.87	39.15	79.52	38.46	38.30	79.63	36.58	32.27	76.43	34.02	31.97	77.75
	IK-MKC	33.11	30.36	77.49	43.72	28.52	77.14	49.04	28.07	77.05	40.64	29.17	76.17
	FLSD	55.89	41.76	81.13	55.97	39.70	80.32	55.37	37.71	79.32	54.75	35.42	78.07
	IMVC-CBG	48.57	43.60	80.35	46.67	43.62	81.59	46.60	40.59	81.05	43.76	42.33	79.86
	LSIMVC	41.35	29.74	77.17	38.94	28.90	74.32	44.37	32.45	77.73	45.69	32.10	77.81
	BGIMVSC	62.65	51.14	82.77	53.26	48.58	83.24	**61.39**	44.66	80.75	59.81	51.56	81.61
	PIMVC	56.44	56.08	85.96	55.67	54.52	84.48	49.87	50.03	83.65	54.99	53.73	85.55
	HCLS	62.22	59.78	85.31	**60.75**	50.43	85.35	60.02	48.42	84.94	61.30	47.81	84.46
	Ours	**62.73**	**62.02**	**87.90**	59.48	**60.69**	**87.34**	60.49	**58.05**	**86.70**	**61.34**	**58.49**	**87.25**
Caltech101-20	BSV	42.41	39.30	55.90	34.87	30.26	50.43	41.91	25.10	45.56	40.65	18.64	45.30
	MIC	31.62	37.45	60.18	28.07	33.88	57.28	26.90	30.36	54.29	23.37	25.08	50.58
	MKKM-IK	33.28	46.23	66.91	34.02	44.84	66.62	28.21	32.74	59.18	31.30	34.85	56.90
	UEAF	41.68	54.53	72.54	39.12	49.50	69.54	35.35	45.52	66.76	33.71	44.02	66.53
	IK-MKC	35.38	47.17	67.47	34.85	41.86	63.89	30.40	33.63	56.83	30.17	33.42	57.23
	FLSD	41.85	54.55	72.90	41.50	52.33	71.83	40.47	49.40	69.93	38.77	47.58	68.41
	IMVC-CBG	47.86	58.92	75.98	48.78	56.80	74.75	44.57	53.99	72.90	43.82	53.12	72.27
	LSIMVC	32.42	40.01	61.94	33.03	39.79	61.76	32.61	39.42	61.84	33.04	39.42	62.23
	BGIMVSC	50.78	52.43	66.18	50.62	47.80	64.82	43.20	46.82	60.15	48.32	47.08	63.94
	PIMVC	49.19	60.52	76.35	47.71	60.29	76.28	46.02	58.39	74.88	46.17	57.62	74.40
	HCLS	52.44	58.03	73.57	57.44	59.92	74.29	53.10	56.47	72.62	53.79	56.78	74.15
	Ours	**71.14**	**68.63**	**79.91**	**69.45**	**64.93**	**77.58**	**57.28**	**61.34**	**76.51**	**65.02**	**61.13**	**75.98**
CCV	BSV	11.75	5.94	13.67	12.08	5.25	12.88	11.82	4.60	12.70	12.00	4.39	12.57
	MIC	15.08	10.56	19.10	13.90	9.10	17.53	12.76	7.57	16.32	12.00	6.66	15.57
	MKKM-IK	15.42	11.55	19.53	15.20	10.83	19.02	13.74	9.47	18.16	14.57	10.15	18.40
	UEAF	19.06	17.05	22.32	19.02	16.04	21.31	16.90	14.03	19.29	15.97	12.94	18.63
	IK-MKC	18.03	14.11	22.19	16.65	12.78	20.39	15.30	11.50	18.83	14.75	11.18	18.40
	FLSD	14.31	12.33	18.46	13.81	10.70	17.98	14.23	10.48	18.05	13.70	10.20	17.56
	IMVC-CBG	21.73	16.82	25.11	20.22	15.70	23.32	18.55	14.41	21.98	17.15	12.91	20.88
	LSIMVC	19.26	18.17	24.39	18.94	16.82	23.76	17.45	14.83	21.55	16.19	12.79	20.13
	BGIMVSC	19.11	16.78	20.19	15.33	12.96	17.44	17.08	14.25	19.79	16.32	13.10	19.32
	PIMVC	18.51	17.51	23.21	18.30	16.38	23.48	16.66	14.27	20.93	16.01	13.35	19.67
	HCLS	17.67	13.49	21.82	18.51	13.39	22.68	17.99	13.13	21.33	17.90	13.15	21.71
	Ours	**23.67**	**18.16**	**26.16**	**21.97**	**17.06**	**24.81**	**20.42**	**15.76**	**23.07**	**20.00**	**14.70**	**22.88**

Table 2: Clustering results of different methods on Caltech101-7, Caltech101-20 and CCV.

$\mathcal{O}(dn+l_tdn+l_tnm_t)$ and $\mathcal{O}(l_tm_t^2)$. Hence, updating $\{\mathbf{A}_t\}$ takes $\mathcal{O}(n\sum_{t=1}^{T}(l_td+l_tm_t))$ totally. When updating each $\mathbf{G}_t$, constructing q and computing analytical solutions take $\mathcal{O}(m_tl_td+m_tdn+dn)$ and $\mathcal{O}(m_tn)$. Hence, updating $\{\mathbf{G}_t\}$ takes $\mathcal{O}(d\sum_{t=1}^{T}(m_tl_t)+dn\sum_{t=1}^{T}(m_t))$ totally. Constructing $h_{v,t}$ takes $\mathcal{O}(d_vn+d_vl_tm_t+d_vm_tn+m_tn)$. Hence, updating $\boldsymbol{\alpha}$ takes $\mathcal{O}(d\sum_{t=1}^{T}(l_tm_t)+nd\sum_{t=1}^{T}(m_t))$. Due to $m_t \leq l_t$ and $m_t \ll d$, hence, updating $\{\mathbf{P}_{v,t}\}$, $\{\mathbf{A}_t\}$, $\{\mathbf{G}_t\}$ and $\boldsymbol{\alpha}$ totally takes $\mathcal{O}(nd\sum_{t=1}^{T}(l_t))$. Evidently, it is linear w.r.t. n. In integration stage, it also takes $\mathcal{O}(n)$. Hence, our DVSAI enjoys linear time overhead w.r.t. n.

Space Complexity The main storage overheads of DVSAI are from the optimization variables $\{\mathbf{P}_{v,t} \in \mathbb{R}^{d_v \times l_t}\}$, $\{\mathbf{A}_t \in \mathbb{R}^{l_t \times m_t}\}$, $\{\mathbf{G}_t \in \mathbb{R}^{m_t \times n}\}$ and $\boldsymbol{\alpha} \in \mathbb{R}^{V \times T}$. Therefore, the memory cost is $d\sum_{t=1}^{T} l_t + n\sum_{t=1}^{T} m_t$. In general, the anchor dimension l_t is a constant and much smaller than n. Consequently, our space complexity is $\mathcal{O}(n)$.

Experiments

Datasets, Baselines and Setup

Six datasets are utilized in the experiments. **Caltech101-7** and **Caltech101-20** are two image datasets with small size. **CCV** and **SUNRGBD** are the video and the 3D datasets with medium size. **NUSWIDEOBJ** and **YoutubeFace_Sel** are large-size web-page and video datasets respectively. Their detailed descriptions are presented in Table 1. To demonstrate the superiority of DVSAI, we compare it with the following eleven strong baselines: **BSV**(Jordan and Weiss 2002), **MIC** (Shao, He, and Yu 2015), **MKKM-IK** (Liu et al. 2017), **UEAF** (Wen et al. 2019), **IK-MKC** (Liu et al. 2020), **FLSD** (Wen et al. 2021), **IMVC-CBG** (Wang et al. 2022b), **LSIMVC** (Liu et al. 2022a), **BGIMVSC** (Sun et al. 2023), **PIMVC** (Deng et al. 2023), **HCLS** (Wen et al. 2023).

We tune the hyper-parameter β in $[2^{-4}, 2^{-3}, \cdots, 2^3, 2^4]$ and γ in $[10^2, 10^3, 10^4, 10^5]$. We set the dimension and size of anchors l_t and m_t in space t to be the same, both as tk, and the number of spaces T as 5. Three metrics, ACC, NMI, Purity, are employed to assess the clustering performance.

Data	Methods	0.1			0.3			0.5			0.7		
		ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
SUNRGBD	BSV	4.76	3.16	13.33	5.34	3.14	13.09	6.03	3.15	12.88	6.98	3.33	13.15
	MIC	16.83	25.12	37.21	15.22	23.05	34.39	14.02	20.93	31.77	13.91	19.35	29.92
	MKKM-IK	11.25	15.04	26.61	11.41	14.86	26.61	11.13	15.05	26.88	11.34	15.24	26.99
	UEAF	17.81	25.21	37.00	16.18	23.04	35.08	15.35	21.59	33.58	14.10	**20.29**	31.64
	IK-MKC	19.85	24.83	37.67	17.96	21.89	34.47	16.68	19.85	32.33	15.35	18.67	30.83
	FLSD	15.09	22.57	34.28	14.67	21.46	32.99	14.28	20.45	31.76	14.33	20.12	31.91
	IMVC-CBG	17.95	24.40	37.94	16.60	21.99	35.34	15.73	19.94	32.79	15.95	18.68	31.32
	LSIMVC	11.75	18.35	29.24	12.00	18.28	29.58	12.79	18.24	29.59	12.95	18.64	30.56
	BGIMVSC	15.21	8.73	16.03	11.42	3.85	12.11	11.01	3.28	11.65	10.97	3.07	11.64
	PIMVC	16.09	24.89	36.45	15.55	**23.97**	35.50	14.99	**22.41**	33.96	14.60	20.07	32.88
	HCLS	20.74	24.73	37.04	19.34	23.08	35.24	17.43	22.21	34.25	17.25	20.05	32.89
	Ours	**23.98**	**25.34**	**37.31**	**21.83**	23.22	**35.54**	**21.08**	21.87	**34.32**	**19.83**	20.15	**33.59**
NUSWIDEOBJ	BSV	-	-	-	-	-	-	-	-	-	-	-	-
	MIC	-	-	-	-	-	-	-	-	-	-	-	-
	MKKM-IK	-	-	-	-	-	-	-	-	-	-	-	-
	UEAF	-	-	-	-	-	-	-	-	-	-	-	-
	IK-MKC	-	-	-	-	-	-	-	-	-	-	-	-
	FLSD	-	-	-	-	-	-	-	-	-	-	-	-
	IMVC-CBG	12.35	10.89	22.10	12.69	10.44	21.90	11.88	9.92	21.45	11.71	9.49	**21.13**
	LSIMVC	-	-	-	-	-	-	-	-	-	-	-	-
	BGIMVSC	-	-	-	-	-	-	-	-	-	-	-	-
	PIMVC	-	-	-	-	-	-	14.40	**12.38**	22.13	13.37	**11.27**	21.08
	HCLS	-	-	-	-	-	-	-	-	-	-	-	-
	Ours	**18.85**	**13.89**	**23.98**	**17.57**	**12.58**	**23.09**	**17.29**	11.82	**22.55**	**15.98**	10.69	21.10
YoutubeFace_Sel	BSV	-	-	-	-	-	-	-	-	-	-	-	-
	MIC	-	-	-	-	-	-	-	-	-	-	-	-
	MKKM-IK	-	-	-	-	-	-	-	-	-	-	-	-
	UEAF	-	-	-	-	-	-	-	-	-	-	-	-
	IK-MKC	-	-	-	-	-	-	-	-	-	-	-	-
	FLSD	-	-	-	-	-	-	-	-	-	-	-	-
	IMVC-CBG	18.50	15.67	31.24	16.89	13.61	29.85	15.63	12.18	28.71	14.06	10.97	28.16
	LSIMVC	-	-	-	-	-	-	-	-	-	-	-	-
	BGIMVSC	-	-	-	-	-	-	-	-	-	-	-	-
	PIMVC	-	-	-	-	-	-	-	-	-	-	-	-
	HCLS	-	-	-	-	-	-	-	-	-	-	-	-
	Ours	**27.13**	**25.82**	**37.14**	**26.67**	**24.25**	**35.94**	**25.84**	**22.95**	**35.38**	**23.58**	**20.46**	**33.70**

Table 3: Clustering results of different methods on SUNRGBD, NUSWIDEOBJ and YoutubeFace_Sel.

Dataset	0.1		0.3		0.5		0.7	
	COMP	Ours	COMP	Ours	COMP	Ours	COMP	Ours
Caltech101-7	0.37	0.08	0.37	0.08	0.38	0.07	0.36	0.07
Caltech101-20	0.84	0.36	0.84	0.35	0.82	0.34	0.83	0.35
CCV	4.31	0.26	4.27	0.25	4.29	0.26	4.28	0.25
SUNRGBD	11.03	0.51	11.21	0.51	11.11	0.52	10.95	0.50
NUSWIDEOBJ	151.21	4.32	154.46	4.29	150.52	4.29	150.93	4.16
YoutubeFace_Sel	-	13.67	-	13.83	-	13.70	-	13.73

Table 4: Execution time (in second) comparison between the COMP scheme and our graph integration scheme.

Experimental Results

Table 2 and 3 present the clustering results on six datasets under the missing rate $p = 0.1, 0.3, 0.5, 0.7$, where '-' denotes the memory overflow failure. We can have that:

(1) Our DVSAI displays obvious advantages over these advanced IMVC competitors. Especially under the missing rate 0.1, it exceeds all compared approaches in ACC, NMI and PUR. Moreover, on datasets Caltech101-20 and CCV, it makes the best results. On other datasets and missing rates, it also can generate comparable results. These give well evidence of the effectiveness of our DVSAI.

(2) For YoutubeFace_Sel, which has more than 100000 instances, most of approaches report the memory overflow failure. This indicates that these approaches can not effectively tackle large-size IMVC problem. By contrast, DVSAI can not only execute normally but also produce the best results in three metrics. These suggest that DVSAI is well able to work with large-size partial multi-view data.

(3) In contrast with MIC, MKKM-IK, IK-MKC, UEAF, FLSD, etc, which handle incomplete multi-view clustering

Dataset	Ablation	0.1			0.3			0.5			0.7		
		ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
Caltech101-7	SDSS	51.22	54.17	83.99	48.24	48.72	80.33	48.41	46.60	79.20	48.85	45.77	79.11
	SDMS	53.12	55.70	85.96	50.63	51.15	83.71	50.36	48.51	82.21	50.98	48.26	82.09
	MDSS	53.91	55.24	85.45	49.18	49.08	81.14	48.54	46.91	79.77	49.69	47.47	79.72
	Ours	**62.73**	**62.02**	**87.90**	**59.48**	**60.69**	**87.34**	**60.49**	**58.05**	**86.70**	**61.34**	**58.49**	**87.25**
Caltech101-20	SDSS	69.45	66.52	77.87	56.62	59.85	74.60	52.76	57.76	73.30	62.99	58.33	72.93
	SDMS	69.75	66.66	77.96	58.78	61.16	75.05	54.03	59.99	74.68	64.29	59.28	74.50
	MDSS	69.78	66.70	79.04	58.50	61.99	75.68	56.84	59.75	73.91	63.45	59.19	73.93
	Ours	**71.14**	**68.63**	**79.91**	**69.45**	**64.93**	**77.58**	**57.28**	**61.34**	**76.51**	**65.02**	**61.13**	**75.98**
CCV	SDSS	22.10	16.98	24.91	21.08	16.16	23.95	19.25	14.27	22.22	18.75	13.35	21.35
	SDMS	22.68	17.67	25.32	21.13	16.38	24.25	19.61	15.14	22.66	19.09	14.37	22.30
	MDSS	22.20	17.28	24.98	21.88	16.76	24.35	19.40	14.87	22.92	19.75	14.35	22.35
	Ours	**23.67**	**18.16**	**26.16**	**21.97**	**17.06**	**24.81**	**20.42**	**15.76**	**23.07**	**20.00**	**14.70**	**22.88**
SUNRGBD	SDSS	21.42	23.75	34.52	19.16	21.37	33.33	19.03	20.54	32.04	16.03	18.05	31.35
	SDMS	22.96	24.79	35.77	20.95	22.67	35.31	19.93	20.91	32.28	16.77	19.04	31.71
	MDSS	22.15	24.55	35.21	19.61	22.37	34.97	19.68	20.86	32.22	16.56	18.95	31.92
	Ours	**23.98**	**25.34**	**37.31**	**21.83**	**23.22**	**35.54**	**21.08**	**21.87**	**34.32**	**19.83**	**20.15**	**33.59**
NUSWIDEOBJ	SDSS	18.58	12.88	22.95	16.79	12.07	21.45	16.16	11.05	21.13	15.42	10.01	20.02
	SDMS	18.63	13.26	23.62	17.23	12.27	22.86	17.06	11.42	22.06	15.61	10.38	21.02
	MDSS	18.74	13.20	23.05	17.27	12.15	22.61	16.93	11.46	22.26	15.64	10.29	20.65
	Ours	**18.85**	**13.89**	**23.98**	**17.57**	**12.58**	**23.09**	**17.29**	**11.82**	**22.55**	**15.98**	**10.69**	**21.10**
YoutubeFace_Sel	SDSS	25.00	23.34	32.38	24.09	23.28	32.62	24.03	21.45	32.16	21.36	18.56	31.15
	SDMS	26.43	24.73	35.74	25.28	23.55	35.01	24.23	21.72	34.14	21.94	18.95	31.52
	MDSS	26.30	25.14	36.70	24.96	23.73	34.53	24.23	22.70	34.90	23.39	20.08	32.42
	Ours	**27.13**	**25.82**	**37.14**	**26.67**	**24.25**	**35.94**	**25.84**	**22.95**	**35.38**	**23.58**	**20.46**	**33.70**

Table 5: Ablation on the multi-dimension multi-size strategy.

tasks by subspace graph or kernel, DVSAI can produce the best results in most cases. These demonstrate that our strategy constructing anchor graph to solve IMVC is effective.

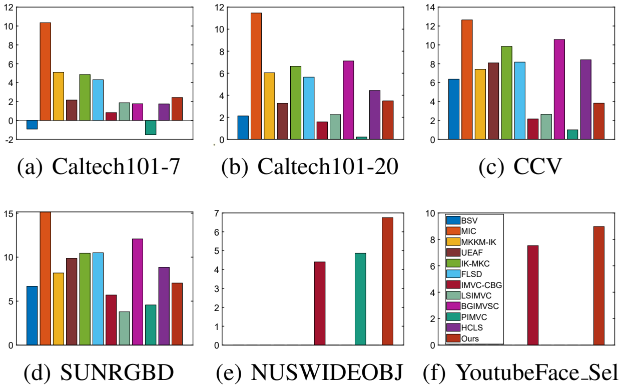
(4) In contrast with IMVC-CBG which generates consensus anchors for IMVC, our DVSAI outperforms it in terms of three metrics in most situations on all datasets. These illustrate that our scheme producing diverse view-shared anchors is more suitable for coping with IMVC problems.

(5) BGIMVSC, PIMVC and HCLS are slightly superior to DVSAI in certain aspects, possibly because BGIMVSC learns the consensus probability information by relaxing the spectral clustering framework, PIMVC exploits the structure representation among data via a group of projective regularizers, and HCLS tries to construct the high-level consensus graph based on a confidence graph. Nevertheless, they are all incapable of solving the large-size clustering problem.

Besides, to illustrate the efficiency of the devised integration scheme, we count its execution time. According to Table 4, we have that compared to the generally adopted COMP (Chen and Cai 2011), our integration scheme spends less time. Especially, when running on YoutubeFace_Sel, COMP induces memory overflow while ours still executes normally.

Efficiency

As proven previously, DVSAI enjoys $\mathcal{O}(n)$ time expenditure. For validating its efficiency, we record the running time. Fig. 2 shows that DVSAI has shorter running time. IMVC-CBG is a little faster than us, which is mainly because it generates a single anchor matrix. Despite time-saving, it is incapable of exploiting features at diverse scales.



(a) Caltech101-7 (b) Caltech101-20 (c) CCV

(d) SUNRGBD (e) NUSWIDEOBJ (f) YoutubeFace_Sel

Figure 2: Running time ($\log_2$) comparison on six datasets.

Convergence and Hyper-parameter Sensitivity

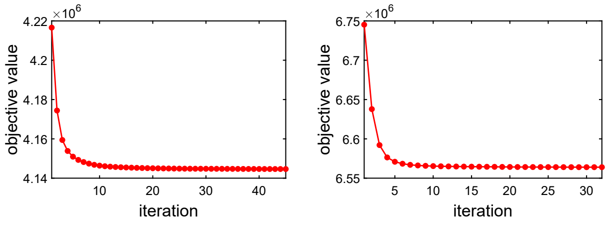
Fig. 3 show that after each iteration, the objective values reduce monotonically, and gradually reach a steady status, which well reveals DVSAI's convergence. Fig. 4 shows that the performance is not largely affected by β and γ, indicating that DVSAI is robust to hyperparameters to some extent.

Ablation Experiments

Unlike existing techniques generating the view-shared anchors with single-dimension single-size (SDSS), we successfully generate the ones with multi-dimension multi-size (MDMS). For validating the MDMS's effectiveness, we conduct four groups of ablations: SDSS, single-dimension multi-size (SDMS), multi-dimension single-size (MDSS),

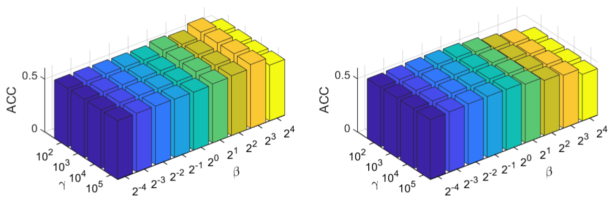
Dataset	T	0.1			0.3			0.5			0.7		
		ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
Caltech101-7	1	51.22	54.17	83.99	48.24	48.72	80.33	48.41	46.60	79.20	48.85	45.77	79.11
	2	52.31	57.40	85.09	51.70	54.50	84.19	50.27	51.50	81.80	51.61	50.13	82.75
	3	61.46	60.95	87.71	53.91	57.08	85.94	52.91	53.88	83.66	57.35	53.34	83.47
	4	62.15	61.74	**87.94**	53.31	56.00	83.74	58.00	55.41	84.69	58.88	54.85	83.92
	5	**62.73**	**62.02**	87.90	**59.48**	**60.69**	**87.34**	**60.49**	**58.05**	**86.70**	**61.34**	**58.49**	**87.25**
Caltech101-20	1	69.45	66.52	77.87	56.62	59.85	74.60	52.76	57.76	73.30	62.99	58.33	72.93
	2	69.52	67.02	78.45	55.75	59.09	74.35	53.15	58.84	74.20	61.41	58.72	73.16
	3	70.92	67.53	79.63	66.15	63.44	76.81	56.27	60.36	75.75	64.85	60.69	75.32
	4	**71.36**	67.85	**80.00**	66.63	62.94	76.05	56.20	59.37	74.82	59.67	60.16	75.62
	5	71.14	**68.63**	79.91	**69.45**	**64.93**	**77.58**	**57.28**	**61.34**	**76.51**	**65.02**	**61.13**	**75.98**
CCV	1	22.10	16.98	24.91	21.08	16.16	23.95	19.25	14.27	22.22	18.75	13.35	21.35
	2	22.27	17.31	25.14	21.53	16.72	24.41	20.18	15.22	**23.39**	19.45	14.27	22.39
	3	22.48	17.33	25.69	21.01	16.69	23.73	19.71	15.15	22.74	19.30	14.42	22.06
	4	22.90	17.43	25.78	20.98	16.87	23.95	20.01	15.49	22.75	19.31	14.40	22.18
	5	**23.67**	**18.16**	**26.16**	**21.97**	**17.06**	**24.81**	**20.42**	**15.76**	23.07	**20.00**	**14.70**	**22.88**
SUNRGBD	1	21.42	23.75	34.52	19.16	21.37	33.33	19.03	20.54	32.04	16.03	18.05	31.35
	2	24.03	24.83	35.78	20.53	22.74	35.44	20.45	20.66	32.28	18.43	19.62	32.62
	3	23.69	24.99	36.00	21.15	**23.23**	**35.82**	19.88	20.58	32.25	18.97	19.75	32.90
	4	23.72	25.33	36.53	21.31	23.17	35.71	20.71	20.54	32.39	19.03	19.96	33.26
	5	**23.98**	**25.34**	**37.31**	**21.83**	23.22	35.54	**21.08**	**21.87**	**34.32**	**19.83**	**20.15**	**33.59**
NUSWIDEOBJ	1	18.58	12.88	22.95	16.79	12.07	21.45	16.16	11.05	21.13	15.42	10.01	20.02
	2	18.42	13.70	23.82	17.39	12.37	22.95	16.57	11.71	22.32	15.55	10.66	**21.41**
	3	18.16	13.58	23.85	16.82	12.37	22.74	16.57	11.68	22.23	**16.49**	10.65	21.27
	4	18.64	**13.94**	**24.02**	17.17	12.53	22.87	**17.31**	11.70	22.36	15.79	10.67	21.20
	5	**18.85**	13.89	23.98	**17.57**	**12.58**	**23.09**	17.29	**11.82**	**22.55**	15.98	**10.69**	21.10
YoutubeFace_Sel	1	25.00	23.34	32.38	24.09	23.28	32.62	24.03	21.45	32.16	21.36	18.56	31.15
	2	**27.74**	25.68	**37.39**	24.12	23.42	33.93	24.37	22.39	34.53	23.55	20.42	33.39
	3	27.18	25.27	37.02	24.77	23.75	34.68	24.59	22.46	34.62	23.19	20.38	33.25
	4	26.95	25.37	36.91	24.68	23.88	35.00	24.48	22.30	34.55	23.14	20.37	33.39
	5	27.13	**25.82**	37.14	**26.67**	**24.25**	**35.94**	**25.84**	**22.95**	**35.38**	**23.58**	**20.46**	**33.70**

Table 6: Ablation on the number of potential spaces.



(a) Caltech101-7 (b) Caltech101-20

Figure 3: Convergence curves.



(a) Caltech101-7 (b) Caltech101-20

Figure 4: Sensitivity of hyper-parameters β and γ.

and our MDMS. Table 5 clearly illustrate that our MDMS strategy indeed brings the performance raise.

To explore the influence of the number of potential spaces T on clustering results, we respectively count the results when T takes 1, 2, 3, 4, 5 in Table 6. As seen, when T is 5, most of results are the best. Some sub-optimal results could be due to the redundancy of potential spaces. How to determine the optimal number of potential spaces automatically is expected to be further studied in the future.

Conclusion

In the article, we design a DVSAI framework to solve the IMVC problem. It integrates anchor generation and anchor graph construction, and generates view-shared anchors with multi-dimension and multi-size. Besides being able to alleviate the diversity deterioration, these diverse view-shared anchors can characterize the actual distribution of the whole samples more sufficiently. For efficiently combining the produced multi-scale graphs together, we devise an integration scheme with linear overheads. For optimizing the resultant IMVC objective, we propose an iterative approach involving four variables, which owns linear time and space expenditures. Plenty of experiments have displayed its advantages, even under high missing rate or/and large-size datasets.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (No. 2021YFB3100700 and 2022ZD0209103); in part by the National Natural Science Foundation of China (No. 62325604 and 62276271); in part by the Hunan Provincial Natural Science Foundation of China (No. 2021JJ30779).

References

Chen, M.-S.; Wang, C.-D.; Huang, D.; Lai, J.-H.; and Yu, P. S. 2022. Efficient Orthogonal Multi-view Subspace Clustering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 127–135.

Chen, X.; and Cai, D. 2011. Large scale spectral clustering with landmark-based representation. In *Twenty-fifth AAAI conference on artificial intelligence*.

Deng, S.; Wen, J.; Liu, C.; Yan, K.; Xu, G.; and Xu, Y. 2023. Projective incomplete multi-view clustering. *IEEE Transactions on Neural Networks and Learning Systems*.

Fang, X.; Hu, Y.; Zhou, P.; and Wu, D. O. 2020. V^3H: View variation and view heredity for incomplete multiview clustering. *IEEE Transactions on Artificial Intelligence*, 1(3): 233–247.

Fu, S.; Cao, Q.; Lei, Y.; Zhong, Y.; Zhan, Y.; and You, X. 2023. Few-Shot Learning With Dynamic Graph Structure Preserving. *IEEE Transactions on Industrial Informatics*.

Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14: 849–856.

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

Li, L.; Zhang, J.; Wang, S.; Liu, X.; Li, K.; and Li, K. 2023. Multi-view bipartite graph clustering with coupled noisy feature filter. *IEEE Transactions on Knowledge and Data Engineering*.

Li, M.; Wang, S.; Liu, X.; and Liu, S. 2022a. Parameter-free and scalable incomplete multiview clustering with prototype graph. *IEEE Transactions on Neural Networks and Learning Systems*.

Li, S.-Y.; Jiang, Y.; and Zhou, Z.-H. 2014. Partial multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2022b. Multiview Clustering: A Scalable and Parameter-Free Bipartite Graph Fusion Method. *IEEE transactions on pattern analysis and machine intelligence*, 44(1): 330–344.

Li, Z.; Tang, C.; Zheng, X.; Liu, X.; Zhang, W.; and Zhu, E. 2022c. High-order correlation preserved incomplete multi-view subspace clustering. *IEEE Transactions on Image Processing*, 31: 2067–2080.

Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; and Liu, X. 2023. Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1559–1568.

Liu, C.; Wu, Z.; Wen, J.; Xu, Y.; and Huang, C. 2022a. Localized sparse incomplete multi-view clustering. *IEEE Transactions on Multimedia*.

Liu, S.; Liu, X.; Wang, S.; Niu, X.; and Zhu, E. 2022b. Fast Incomplete Multi-View Clustering With View-Independent Anchors. *IEEE Transactions on Neural Networks and Learning Systems*.

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

Liu, X.; Li, M.; Wang, L.; Dou, Y.; Yin, J.; and Zhu, E. 2017. Multiple Kernel k-Means with Incomplete Kernels. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Zhu, E.; Liu, T.; Kloft, M.; Shen, D.; Yin, J.; and Gao, W. 2020. Multiple Kernel k-Means with Incomplete Kernels. *IEEE transactions on pattern analysis and machine intelligence*.

Qiang, Q.; Zhang, B.; Wang, F.; and Nie, F. 2021. Fast multi-view discrete clustering with anchor graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9360–9367.

Shao, W.; He, L.; and Yu, P. S. 2015. Multiple incomplete views clustering via weighted nonnegative matrix factorization with $\ell_{\{}2,1\}$ regularization. In *Joint European conference on machine learning and knowledge discovery in databases*, 318–334. Springer.

Sun, L.; Wen, J.; Liu, C.; Fei, L.; and Li, L. 2023. Balance guided incomplete multi-view spectral clustering. *Neural Networks*.

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

Wang, H.; Zong, L.; Liu, B.; Yang, Y.; and Zhou, W. 2019a. Spectral perturbation meets incomplete multi-view data. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3677–3683.

Wang, Q.; Tao, Z.; Gao, Q.; and Jiao, L. 2022a. Multi-view subspace clustering via structured multi-pathway network. *IEEE Transactions on Neural Networks and Learning Systems*.

Wang, S.; Liu, X.; Liu, L.; Tu, W.; Zhu, X.; Liu, J.; Zhou, S.; and Zhu, E. 2022b. Highly-efficient incomplete large-scale multi-view clustering with consensus bipartite graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9776–9785.

Wang, S.; Liu, X.; Zhu, E.; Tang, C.; Liu, J.; Hu, J.; Xia, J.; and Yin, J. 2019b. Multi-view clustering via late fusion alignment maximization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3778–3784.

Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; and Zhu, E. 2022c. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31: 556–568.

Wen, J.; Liu, C.; Xu, G.; Wu, Z.; Huang, C.; Fei, L.; and Xu, Y. 2023. Highly Confident Local Structure Based Consensus Graph Learning for Incomplete Multi-View Clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15712–15721.

Wen, J.; Zhang, Z.; Xu, Y.; Zhang, B.; Fei, L.; and Liu, H. 2019. Unified embedding alignment with missing views inferring for incomplete multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 5393–5400.

Wen, J.; Zhang, Z.; Zhang, Z.; Fei, L.; and Wang, M. 2021. Generalized Incomplete Multiview Clustering With Flexible Locality Structure Diffusion. *IEEE transactions on cybernetics*, 51(1): 101–114.

Xia, W.; Gao, Q.; Wang, Q.; and Gao, X. 2022a. Tensor completion-based incomplete multiview clustering. *IEEE Transactions on Cybernetics*, 52(12): 13635–13644.

Xia, W.; Gao, Q.; Wang, Q.; Gao, X.; Ding, C.; and Tao, D. 2022b. Tensorized bipartite graph learning for multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yang, M.; Li, Y.; Huang, Z.; Liu, Z.; Hu, P.; and Peng, X. 2021a. 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.

Yang, Y.; Zhan, D.-C.; Wu, Y.-F.; Liu, Z.-B.; Xiong, H.; and Jiang, Y. 2021b. Semi-Supervised Multi-Modal Clustering and Classification with Incomplete Modalities. *IEEE Transactions on Knowledge and Data Engineering*, 33(2): 682–695.

Yu, S.; Liu, S.; Wang, S.; Tang, C.; Luo, Z.; Liu, X.; and Zhu, E. 2023a. Sparse Low-Rank Multi-View Subspace Clustering With Consensus Anchors and Unified Bipartite Graph. *IEEE Transactions on Neural Networks and Learning Systems*.

Yu, S.; Wang, S.; Wen, Y.; Wang, Z.; Luo, Z.; Zhu, E.; and Liu, X. 2023b. How to Construct Corresponding Anchors for Incomplete Multiview Clustering. *IEEE Transactions on Circuits and Systems for Video Technology*.

Zhang, C.; Han, Z.; Fu, H.; Zhou, J. T.; Hu, Q.; et al. 2019. CPM-Nets: Cross partial multi-view networks. *Advances in Neural Information Processing Systems*, 32.