Active Deep Multi-view Clustering

Helin Zhao[1] , **Wei Chen**[1] , **Peng Zhou**[1*]

[1]Anhui Provincial International Joint Research Center for Advanced Technology in Medical Imaging, School of Computer Science and Technology, Anhui University

{e21201088, e23201102}@stu.ahu.edu.cn, zhoupeng@ahu.edu.cn

Abstract

Deep multi-view clustering has been widely studied. However, since it is an unsupervised task, where no labels are used to guide the training, it is still unreliable especially when handling complicated data. Although deep semi-supervised multi-view clustering can alleviate this problem by using some supervised information, the supervised information is often pregiven or randomly selected. Unfortunately, as we know, the clustering performance highly depends on the quality of the supervised information and most of the semi-supervised methods ignore the supervised information selection. To tackle this problem, in this paper, we propose a novel active deep multi-view clustering method, which can actively select important data for querying human annotations. In this method, we carefully design a fusion module, an active selection module, a supervised module, and an unsupervised module, and integrate them into a unified framework seamlessly. In this framework, we can obtain a more reliable clustering result with as few annotations as possible. The extensive experiments on benchmark data sets show that our method can outperform state-of-the-art unsupervised and semi-supervised methods, demonstrating the effectiveness and superiority of the proposed method. The code is available at https://github.com/wodedazhuozi/ADMC.

1 Introduction

Multi-view data are ubiquitous in real-world applications. For example, web pages on the Internet may contain multiple views such as images, texts, and videos. To handle these multi-view data, multi-view clustering is one of the challenging tasks and attracts much attention [Zhou *et al.*, 2019; Zhang *et al.*, 2019; Xie *et al.*, 2020; Wang *et al.*, 2020; Zhou and Du, 2023].

Since the deep neural network (DNN) has been widely used in many fields and achieves promising performance on many tasks, deep multi-view clustering attracts increasingly more attention [Xue *et al.*, 2021; Xu *et al.*, 2021; Lin *et al.*, 2021a; Xia *et al.*, 2021; Lin *et al.*, 2021b; Wang *et al.*, 2022; Xu *et al.*, 2022a; Du *et al.*, 2022; Yan *et al.*, 2023]. These methods apply the DNN to extract more semantic representations of data and do the clustering on the learned representations. For example, [Du *et al.*, 2021] used auto-encoder to extract features, and designed cross-entropy-based regularization and local regularization to ensure the consistency between any two views of a sample; [Xiao *et al.*, 2023] utilized view-specific graph convolution networks to learn the representation of each view and proposed a fusion method on the attribute-level and structure-level; [Pan and Kang, 2021] learned a new consensus graph by considering the relationship between nodes and the initial graph, and designed a novel graph-level contrastive loss.

However, since multi-view clustering is still an unsupervised task, where there is no supervised information in the learning, deep multi-view clustering may still obtain unreliable results, especially when handling complicated real-world data. To tackle this problem, some semi-supervised multi-view clustering methods are proposed [Chen *et al.*, 2022; Tang *et al.*, 2022; Qin *et al.*, 2021; Whang *et al.*, 2020; Zhu and Gao, 2022]. These methods apply some supervised information, such as the labels of data or the pairwise constraints, to guide the multi-view clustering. For example, [Tang *et al.*, 2022] applied the cannot-link and must-link constraints to guide the clustering and proposed constrained tensor representation learning model based on the unified constraint to learn the representations; [Qin *et al.*, 2021] utilized the labels of data to build the affinity matrix and applied the semi-supervised learning methods on the affinity matrix to obtain the clustering results. Although semi-supervised methods can alleviate the problem of lacking supervised information to some extent, all these methods ignore how to obtain the supervised information. The supervised information used in these methods is all pre-given or randomly generated. As we know, the performance of semi-supervised learning highly depends on the quality of the supervised information, and poor supervised information may deteriorate semi-supervised learning seriously. All these semi-supervised multi-view clustering methods ignore this supervised information selection problem.

To tackle the supervised information selection problem,

*Peng Zhou is the corresponding author.

in this paper, we propose a novel Active Deep Multi-view Clustering (ADMC), which introduces active learning into the multi-view clustering task. Active learning often involves multiple interactive batches. In each batch, the algorithm selects several data with a pre-given budget and queries the human annotations on these selected data [Hoi *et al.*, 2006; Chattopadhyay *et al.*, 2013; Wang *et al.*, 2015a; Yan and Huang, 2018; Wang *et al.*, 2015b; Wang *et al.*, 2019; Halder *et al.*, 2023; Sun *et al.*, 2022; Zhou *et al.*, 2023]. In our method, we design an active selection module to automatically select some important data for annotation by considering the *representative* and *diversity* properties of data. Besides the active selection module, we propose a fusion module to integrate the information in different views. Then, we propose a supervised module to apply the human annotations to train the whole model. Notice that in our active clustering setting, we wish to obtain a reliable result with as few annotations as possible. Therefore, we can only train the network with a very small number of data in the supervised module. To address this issue, we also carefully design an unsupervised module, which can make full use of the remaining unlabeled data to aid the network training. We seamlessly integrate these modules into a unified framework for multi-view clustering. The extensive experiments on benchmark data sets show that our method can outperform other state-of-the-art unsupervised and semi-supervised deep multi-view clustering methods, which well demonstrates the superiority of our method.

The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to tackle the active deep clustering problem on multi-view data.
- We propose a novel active deep multi-view clustering method that can automatically select important data for annotations. Due to the carefully designed active selection module and the unsupervised module, the proposed method can learn better representations with as few annotations as possible.
- Extensive experiments on benchmark data sets demonstrate the superiority of the proposed method.

2 Active Deep Multi-view Clustering

In this section, we introduce our proposed method ADMC. First, we introduce some notations in this paper for a better understanding of our method. Given a multi-view data set $\mathcal{X} = \{X^v \in \mathbb{R}^{N\times d_v}\}_{v=1}^V$ that can be divided into C clusters, where N denotes the number of samples, V represents the number of views, d_v denotes the sample dimension of the v-th view, and x_i^v denotes the v-th view of the i-th sample, we denote $\mathcal{L}$ and $\mathcal{U}$ as the sets of the labeled samples and unlabeled samples, respectively. Therefore, $\mathcal{L} \cup \mathcal{U} = \emptyset$ and $\mathcal{L} \cap \mathcal{U} = \mathcal{X}$. In the beginning, all samples are unlabeled, which means $\mathcal{L} = \emptyset$ and $\mathcal{U} = \mathcal{X}$. Since our ADMC is a batch-mode active learning method, we suppose that there are overall T batches, and the budget of each batch is K, which means in each batch, we can select at most K samples for querying the human annotations.

Figure 1 shows the architecture of our method. In our method, since there are V views, we adopt V same-structured auto-encoders [Hinton and Salakhutdinov, 2006] as the backbones to extract the latent features for each view. In the auto-encoder, we use the multi-layer perceptron (MLP) as the encoder which contains four hidden layers. The activation function used in each layer is nonlinear REctified Linear Unit (ReLU). Let E^v denote the used encoder for the v-th view, and D^v denote the decoder for the v-th view. The reconstructed $\tilde{x}_i^v$ can be represented as follows

$$\tilde{x}_i^v = D^v(E^v(x_i^v)) \tag{1}$$

Then, we use the following reconstruction loss l_{recon} for pre-training:

$$l_{recon} = \sum_{v=1}^{V}\sum_{i=1}^{N} \|x_i^v - D^v(E^v(x_i^v))\|_2^2 \tag{2}$$

Our method contains four modules: a fusion module, an active selection module, a supervised module, and an unsupervised module. After obtaining the features of each view, the fusion module ensembles the features with adaptive weights. Then, we select some important samples for annotation according to the fusion features in the active selection module. Afterward, the data set is divided into the labeled set and the unlabeled set, which are used in the supervised module and the unsupervised module respectively to train the backbone networks. The details of the four modules will be introduced in the following subsections.

2.1 Fusion Module

The fusion module utilizes multiple views to obtain comprehensive information for more accurate and robust results. The traditional feature fusion approaches, such as the summation or concatenation of each view, may be too simple to obtain the intrinsic representations for the complicated real-world multi-view data. In real-world applications, the views may contain various noises and the quality of each view may also differ. Therefore, the traditional methods, which directly and equally concatenate or sum them, may be inappropriate. Additionally, it is often hard to tell which view is better and which view is worse in advance.

To tackle this problem, we propose an adaptive feature fusion module, which can automatically learn the weight of each view, and then fuse all views according to the learned weights. Specifically, for a sample x_i, after obtaining the representations for each view $F = \{f_i^1, \cdots, f_i^V\}$ where $f_i^v = E^v(x_i^v)$, we feed f_i^v into an MLP to obtain the weight for the v-th view $w_i^{v\prime} = MLP(f_i^v) \in \mathbb{R}^{1\times 1}$. Then we use the Softmax function to get the final weight w_i^v for the v-th view of x_i as follows.

$$w_i^v = Softmax(w_i^{v\prime}) = \frac{e^{w_i^{v\prime}}}{\sum_{j=1}^{V} e^{w_i^{j\prime}}} \tag{3}$$

At last, the fused representation of x_i is:

$$\widetilde{f}_i = w_i^1 f_i^1 + \cdots + w_i^V f_i^V. \tag{4}$$

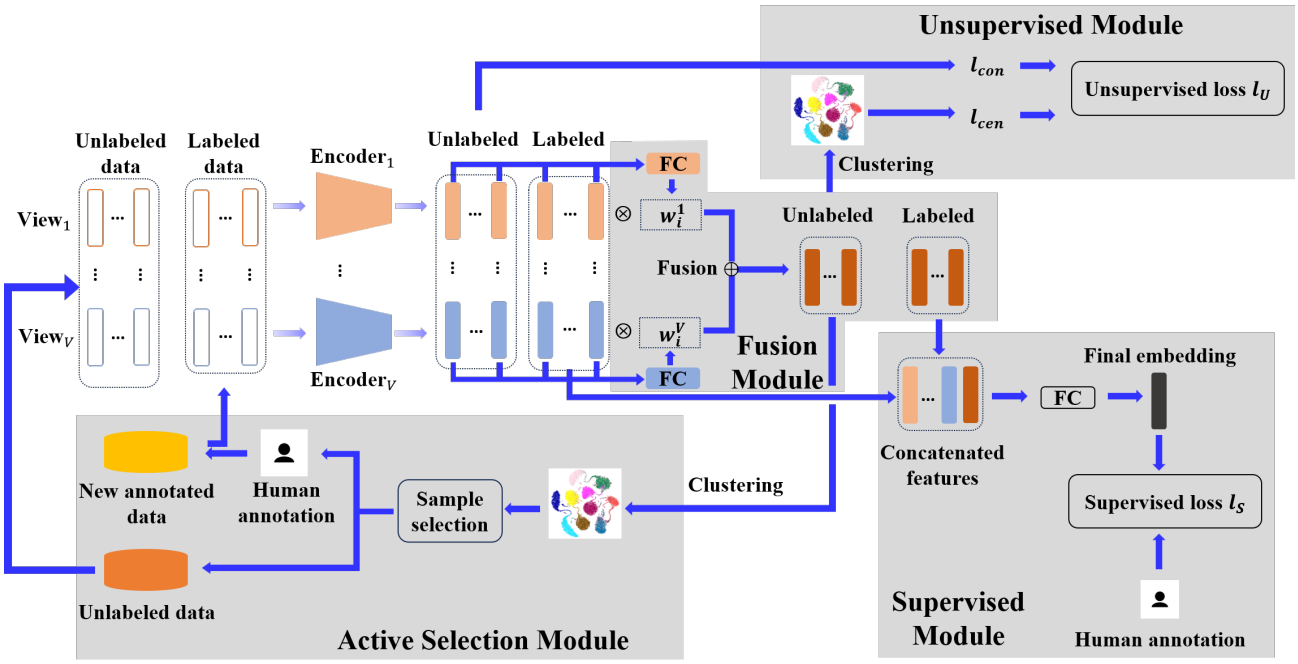


Figure 1: The architecture of ADMC. ADMC contains four modules. The fusion module integrates the representations in each view to obtain a consensus representation. The active selection module automatically selects some important samples for querying human annotations. The supervised module trains the backbone networks with the annotated data. The unsupervised module trains the backbone networks with the unlabeled data.

2.2 Active Selection Module

After obtaining the fused representation of each sample, we actively select several important samples for querying human annotations. We select the samples based on the following two properties:

- *Representativeness.* The representative data can capture the intrinsic clustering structure of data. If we obtain their annotations, we can easily characterize the clustering structure of all data.
- *Diversity.* If two samples are similar, we only need to annotate one of them. Therefore, to avoid the waste of the annotation budget, we should select diverse samples for querying.

To simultaneously achieve both two properties, we design a simple yet effective active selection module. Notice that the budget of each batch is K, which means in each batch, we can only query at most K samples for annotations. We first run k-means on the fused representations $\widetilde{f}_i$ (obtained by Eq.(4)) of all samples in the unlabeled set $\mathcal{U}$ to partition them into K clusters. In each cluster, the closer the sample is to the cluster center, the more representative this sample is. Therefore, for each cluster, we select the sample which is the closest to the cluster center for querying. Since in each cluster, we only select one sample which lies in the center of the cluster, and according to the property of k-means algorithm, all selected samples will be far away from each other and achieve the diversity property.

Denote $\mathcal{S}$ as the set of K selected samples. We query humans to annotate the samples in $\mathcal{S}$. Then, we move the samples in $\mathcal{S}$ from the unlabeled set $\mathcal{U}$ to the labeled set $\mathcal{L}$, which means we update $\mathcal{U} = \mathcal{U} - \mathcal{S}$ and $\mathcal{L} = \mathcal{L} \cup \mathcal{S}$.

2.3 Supervised Module

After obtaining the annotations of several samples, we can design a supervised module to apply the labeled data $\mathcal{L}$ to train the network. Given a labeled sample $x_i = \{x_i^1, \ldots, x_i^V\} \in \mathcal{L}$ whose label is y_i, its high-level representation from the auto-encoder is $\{f_i^1, \ldots, f_i^V\}$, and its fused representation is $\widetilde{f}_i$ (obtained from Eq.(4)). Notice that $\widetilde{f}_i$ contains the consensus information among all views, and $\{f_i^1, \ldots, f_i^V\}$ contains some individual information of each view. To obtain a more comprehensive representation, we first concatenate them as $[f_i^1 \| \ldots \| f_i^V \| \widetilde{f}_i]$, and then feed the concatenated representation into a classification layer, which is a Fully Connected (FC) layer to obtain the final embedding $\hat{y}_i \in \mathbb{R}^{1\times C}$, where C is the number of classes. Then, we feed the final embedding $\hat{y}_i$ into a Softmax function to obtain the probability vector of sample x_i as $\widetilde{y}_i \in \mathbb{R}^{1\times C}$.

To train the network, we use the following cross-entropy loss between the final representation $\widetilde{y}_i$ and the human annotation y_i:

$$l_{label} = \sum_{x_i \in \mathcal{L}} H(y_i, \widetilde{y}_i), \tag{5}$$

where $H(\cdot, \cdot)$ denotes the cross-entropy loss.

Besides this, the reconstruction loss l_{recon} is also used in the supervised module to prevent the model collapse. Therefore, the whole supervised loss l_S of labeled data is as follows

$$l_S = l_{label} + \gamma l_{recon} \tag{6}$$

where γ is a balanced parameter.

2.4 Unsupervised Module

In our active clustering setting, most of the samples are unlabeled. It is hard to directly learn a reliable representation because we do not have enough labeled data to guide the learning. To address this issue, we design an unsupervised module to enhance the representation learning.

In the unsupervised module, since we do not use the label information, we apply contrastive learning to guide the representation learning. Contrastive learning is to learn the common features between similar samples and distinguish the differences between dissimilar samples. In single-view data, contrastive learning often needs some data augmentation techniques to generate positive and negative pairs of data for learning. In our multi-view setting, we can construct positive and negative pairs of data more naturally.

Notice that in multi-view data, each data contains multiple views and thus we can make the different views of the same sample as the positive pairs and the views of different samples as the negative pairs. Specifically, we obtain the representations of each view of x_i and x_j as $f_i^1, \cdots, f_i^V$ and $f_j^1, \cdots, f_j^V$, respectively. We use (f_i^p, f_i^q) and (f_j^p, f_j^q) $(p, q \in \{1, 2, \cdots, V\})$ as the positive pairs and (f_i^p, f_j^q) as the negative pair. Then, we use the cosine similarity to measure the similarity between two representations as follows:

$$sim(f_i^p, f_j^q) = \frac{\langle f_i^p, f_j^q \rangle}{\|f_i^p\|_2 \|f_j^q\|_2}, \tag{7}$$

where $\langle \cdot, \cdot \rangle$ represents the inner production.

Based on this similarity, we can design the following contrastive loss:

$$l_{con} = -\frac{1}{|U|} \sum_{x_i \in \mathcal{U}} \sum_{p=1}^{V} \sum_{q=p+1}^{V} log \frac{e^{sim(f_i^p, f_i^q)/\tau}}{\sum_{j \neq i}^{N} \sum_{v=p,q} e^{sim(f_i^p, f_j^v)/\tau}} \tag{8}$$

where τ denotes the temperature parameter.

Eq.(8) is a sample-level loss that does not consider the cluster structure. To characterize the cluster structure, we also introduce a center loss. The basic idea of center loss is that a clear cluster structure often needs the samples to be close to their cluster centers. To achieve this, we utilize a k-means based center loss function to characterize the cluster structure. In more detail, after obtaining the fused representation $\widetilde{f}_1, \cdots, \widetilde{f}_N$ by Eq.(4), we run k-means on them to obtain C clusters. Then, we design the following center loss l_{cen}:

$$l_{cen} = \sum_{x_i \in \mathcal{U}} \|\widetilde{f}_i - c_i\|_2^2, \tag{9}$$

where c_i denotes the center of the cluster which x_i belongs to. Similar to the supervised module, we also add the reconstruction loss l_{recon} to the unsupervised module, which aims at preventing model collapse. As a result, the whole unsupervised loss l_U is as follows

$$l_U = l_{con} + \alpha l_{cen} + \beta l_{recon}, \tag{10}$$

where α and β are two balanced parameters.

Algorithm 1 Active Deep Multi-view Clustering

Input: Data set $\mathcal{X}$, cluster number C, the budget of each selection batch K, number of selection batches T.
Output: Clustering result

```
Initialize L = ∅, U = X.
Pretrain the auto-encoder by minimizing the reconstruction loss Eq.(2).
for i = 1, ··· , T do
    Obtain the representations of each view by the auto-encoder.
    Obtain the fusion representation F̃ by Eq.(3) and Eq.(4).
    Select samples to annotate with the active selection module, and then update L and U.
    Use the supervised loss Eq.(6) to train auto-encoder on L.
    Use the unsupervised loss Eq.(10) to train the auto-encoder on U.
end for
Obtain the final clustering result from the representation ỹ_1, ··· , ỹ_N.
```

Data set	#samples	#views	#classes
Caltech-2V	1400	40 / 254	7
Caltech-5V	1400	40 / 254 / 1984 / 512 / 928	7
CCV	6773	5000 / 5000 / 4000	20
Reuters	1200	2000 / 2000 / 2000 2000/ 2000	6
Cora	2708	2708 / 1433 / 2706 / 2706	7
BBCSport	544	3183 / 3203	5
Scene	4485	20 / 59 / 40	15
Noisy-Mnist	30000	784 / 784	10

Table 1: Description of the data sets.

2.5 Algorithm

At last, we feed all data into the backbones and the FC layer with Softmax trained in the supervised module to obtain the probability $\widetilde{y}_1, \cdots, \widetilde{y}_N$. Then, given x_i, we put it into the cluster with the largest probability in $\widetilde{y}_i$. The whole algorithm of our method is shown in Algorithm 1.

3 Experiment

In this section, we compare our method with the state-of-the-art unsupervised and semi-supervised methods on benchmark data sets.

3.1 Data sets

To validate the effectiveness of ADMC, we conduct experiments on eight benchmark data sets, including BBCSport [1], Caltech-2V [Fei-Fei *et al.*, 2004], Caltech-5V [Fei-Fei *et al.*, 2004], CCV [Jiang *et al.*, 2011], Reuters1200 [Amini *et al.*, 2009], Cora [Wen *et al.*, 2020], Scene [Fei-Fei and Perona, 2005], and Noisy-Mnist [2]. The details of these data sets are shown in Table 1.

[1] http://mlg.ucd.ie/datasets/

[2] https://github.com/nineleven/NoisyMNISTDetection

Metrics	Method	Caltech-2V	Caltech-5V	CCV	Reuters	Cora	BBCSport	Scene	Noisy-Mnist
ACC	CVCL	0.5648	0.6008	0.1044	0.4366	0.3039	0.5488	0.3822	0.9539
	DealMVC	0.1743	0.1750	0.0833	0.1817	0.2237	0.6176	0.0939	0.1090
	MFLVC	0.6207	0.7171	0.3201	0.4617	0.2493	0.3346	0.3681	0.9733
	DSMVC	0.6064	0.6143	0.1875	0.4108	0.3095	0.5147	0.4161	0.3781
	SDSNE	0.6214	0.7636	0.2596	0.2342	0.3774	0.7868	0.3969	OOM
	ADMC	**0.7643**	**0.9050**	**0.3325**	**0.6575**	**0.6499**	**0.9651**	**0.4591**	**0.9864**
NMI	CVCL	0.4164	0.4615	0.0000	0.2267	0.1270	0.4248	0.4124	0.8980
	DealMVC	0.0079	0.0095	0.0056	0.0046	0.0035	0.4759	0.0072	0.0007
	MFLVC	0.5374	0.6748	0.3001	0.2786	0.0297	0.0553	0.3845	0.9342
	DSMVC	0.5342	0.5056	0.1688	0.1490	0.0961	0.2197	0.4434	0.2999
	SDSNE	0.5240	0.7547	0.2633	0.1681	0.2379	0.6133	0.4272	OOM
	ADMC	**0.6187**	**0.8219**	**0.3068**	**0.4049**	**0.3639**	**0.8899**	**0.4057**	**0.9624**
ARI	CVCL	0.3300	0.3870	0.0000	0.1697	0.0819	0.3352	0.2292	0.9026
	DealMVC	0.0006	0.0016	0.0001	0.0005	0.0049	0.3450	0.0003	0.0001
	MFLVC	0.4329	0.5739	0.1528	0.2136	0.0122	0.0167	0.2254	0.9432
	DSMVC	0.4379	0.4139	0.0593	0.1217	0.0795	0.2457	0.2442	0.2078
	SDSNE	0.4005	0.6459	0.1018	0.0171	0.0485	0.5514	0.2307	OOM
	ADMC	**0.5767**	**0.8103**	**0.1599**	**0.3614**	**0.3708**	**0.9118**	**0.2664**	**0.9706**
PUR	CVCL	0.5781	0.6094	0.1044	0.4462	0.4085	0.6426	0.4234	0.9539
	DealMVC	0.1750	0.1771	0.1057	0.1817	0.3022	0.6507	0.1035	0.1150
	MFLVC	0.6207	0.7264	0.3329	0.4683	0.3135	0.3750	0.4172	0.9733
	DSMVC	0.6114	0.6200	0.2159	0.4133	0.3778	0.5202	0.4334	0.3916
	SDSNE	0.6393	0.7971	0.2788	0.2708	0.3818	0.7868	0.4457	OOM
	ADMC	**0.7643**	**0.8914**	**0.3357**	**0.6575**	**0.6499**	**0.9651**	**0.4604**	**0.9864**

Table 2: The clustering results compared with unsupervised methods. OOM means that the method does not run a result due to the out-of-memory error.

Method	# of annotations	Caltech-2V	Caltech-5V	CCV	Reuters	Cora	BBCSport	Scene	Noisy-Mnist
IMvGCN	10	0.4788	0.7914	0.0841	0.4300	0.4557	0.9157	0.2761	0.5305
	20	0.6200	0.8636	0.0890	0.5083	0.5487	0.9447	0.3621	0.6443
	30	0.6050	0.8493	0.1016	0.5367	0.5907	0.9494	0.3254	0.6756
	40	0.6591	0.8436	0.0992	0.5650	0.5077	0.9504	0.3978	0.6906
	50	0.6099	0.8457	0.1267	0.6308	0.6279	0.9574	0.4145	0.7132
DSRL	10	0.4657	0.7671	0.1181	0.3100	0.3065	0.8199	0.1353	OOM
	20	0.6264	0.8479	0.1397	0.4358	0.3161	0.7941	0.2163	OOM
	30	0.6064	0.8607	0.1314	0.4217	0.3275	0.8511	0.2593	OOM
	40	0.6457	0.8564	0.1400	0.5125	0.3346	0.8603	0.2640	OOM
	50	0.6786	0.8464	0.1537	0.5467	0.3497	0.9577	0.3115	OOM
MVAR	10	0.5057	0.5736	0.1107	0.2242	0.3253	0.5349	0.2002	0.2784
	20	0.5314	0.7036	0.1441	0.3800	0.3689	0.6029	0.2803	0.3108
	30	0.4857	0.7086	0.1727	0.4117	0.3685	0.7813	0.2584	0.2920
	40	0.5450	0.7214	0.1742	0.5075	0.4121	0.7169	0.2493	0.4723
	50	0.6179	0.8164	0.1978	0.5175	0.5037	0.8272	0.3142	0.4667
ADMC	10	0.6093	0.8086	0.1927	0.5242	0.5645	0.8199	0.3530	0.5981
	20	0.6679	0.8521	0.2486	0.5333	0.6089	0.8658	0.4312	0.9433
	30	0.7164	0.8843	0.2692	0.5842	0.6193	0.9375	0.4341	0.9688
	40	0.7521	0.8914	0.3191	0.6208	0.6448	0.9559	0.4406	**0.9866**
	50	**0.7643**	**0.9050**	**0.3325**	**0.6575**	**0.6499**	**0.9651**	**0.4591**	0.9864

Table 3: ACC results compared with semi-supervised methods. OOM means that the method does not run a result due to the out-of-memory error.

Method	# of annotations	Caltech-2V	Caltech-5V	CCV	Reuters	Cora	BBCSport	Scene	Noisy-Mnist
IMvGCN	10	0.4395	0.6796	0.1881	0.2164	0.2069	0.7672	0.3094	0.4460
	20	0.4947	0.7678	0.2004	0.3131	0.2780	0.8349	0.3419	0.5017
	30	0.5095	0.7852	0.1936	0.3164	0.3183	0.8609	0.3651	0.5528
	40	0.5092	0.7721	0.1880	0.3681	0.2458	0.8618	0.3548	0.5615
	50	0.5074	0.7730	0.1893	0.3570	0.3619	0.8750	0.3708	0.5918
DSRL	10	0.3137	0.6570	0.0638	0.1020	0.0104	0.7998	0.1175	OOM
	20	0.3962	0.7273	0.0828	0.1965	0.0316	0.6172	0.2058	OOM
	30	0.4000	0.7420	0.0628	0.2080	0.0538	0.7325	0.2258	OOM
	40	0.4328	0.7386	0.0724	0.2447	0.0697	0.7517	0.2335	OOM
	50	0.4671	0.7309	0.0908	0.2678	0.0932	0.8571	0.2656	OOM
MVAR	10	0.4555	0.4441	0.1155	0.0873	0.0354	0.2829	0.2667	0.2881
	20	0.3540	0.5231	0.1012	0.1582	0.1412	0.3300	0.2530	0.4179
	30	0.3074	0.5559	0.1187	0.1762	0.1057	0.5918	0.1665	0.3946
	40	0.3273	0.5376	0.1393	0.2172	0.1352	0.5630	0.1493	0.5012
	50	0.3814	0.6734	0.1249	0.2506	0.2562	0.6713	0.3162	0.5299
ADMC	10	0.5097	0.6883	0.2194	0.2993	0.3118	0.6568	0.3560	0.6280
	20	0.5455	0.7596	0.2604	0.3039	0.3323	0.7340	0.3926	0.8903
	30	0.5811	0.7924	0.2757	0.3408	0.3416	0.8163	0.3869	0.9321
	40	0.6107	0.7991	0.2997	0.3763	**0.3659**	0.8567	0.3942	**0.9631**
	50	**0.6187**	**0.8219**	**0.3068**	**0.4049**	0.3639	**0.8899**	**0.4057**	0.9624

Table 4: NMI results compared with semi-supervised methods. OOM means that the method does not run a result due to the out-of-memory error.

3.2 Experimental Setup

We compare ADMC with eight state-of-the-art multi-view clustering algorithms including five unsupervised algorithms and three semi-supervised algorithms. The five unsupervised multi-view algorithms are CVCL [Chen *et al.*, 2023], DealMVC [Yang *et al.*, 2023], MFLVC [Xu *et al.*, 2022b], DSMVC [Tang and Liu, 2022] and SDSNE [Liu *et al.*, 2022]. The three semi-supervised multi-view algorithms are IMvGCN [Wu *et al.*, 2023], DSRL [Wang *et al.*, 2021], and MVAR [Tao *et al.*, 2017].

For our proposed ADMC, the sizes of the four hidden layers in the auto-encoder are set to 500, 500, 2000, and 128, respectively. The FC in the fusion module is a four-layer MLP with RELU as an active function whose sizes are set to 128, 128, 256, and 1, respectively. The FC in the supervised module is a single layer whose size is the number of clusters. α is fixed as 10^{-5}, and β and γ are chosen from $[10^{-3}, 10^{3}]$. τ is fixed as 0.6. We use AdmW as the optimizer, The experiments contain 5 selection batches and the budget of each batch is 10 samples. For other compared methods, we use the public codes and pre-trained models provided by their authors to conduct experiments. All experiments are conducted on the PC with AMD Ryzen 7 7840H CPU, NVIDIA GeForce RTX 4060 GPU, and 16GB RAM.

We apply four wildly-used metrics to evaluate the final clustering performance, including clustering ACCuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and PURity (PUR).

3.3 Experimental Results

Table 2 shows the results of ADMC and state-of-the-art unsupervised methods. The best results are highlighted in bold. The results of ADMC in Table 2 are the results after 5 selection batches. Among all data sets on all metrics, ADMC achieves the best results. The results of ADMC are significantly better than other methods. It shows that the supervised information is helpful in the clustering task as claimed in the Introduction.

Tables 3 and 4 show the ACC and NMI results compared with semi-supervised methods on all data sets with different numbers of annotated data. The results w.r.t. ARI and PUR are shown in the Appendix. We can see that the performance of ADMC increases with the increase of human annotations on most data sets. When compared with other semi-supervised methods, our ADMC achieves better performance on all data sets at last. Moreover, we can also find that our ADMC achieves better performance with fewer annotations compared with other methods. For example, on Caltech-2V data set, we only need 30 annotations to outperform other methods with 50 annotations. It well demonstrates the effectiveness of our active clustering method.

3.4 Ablation Study

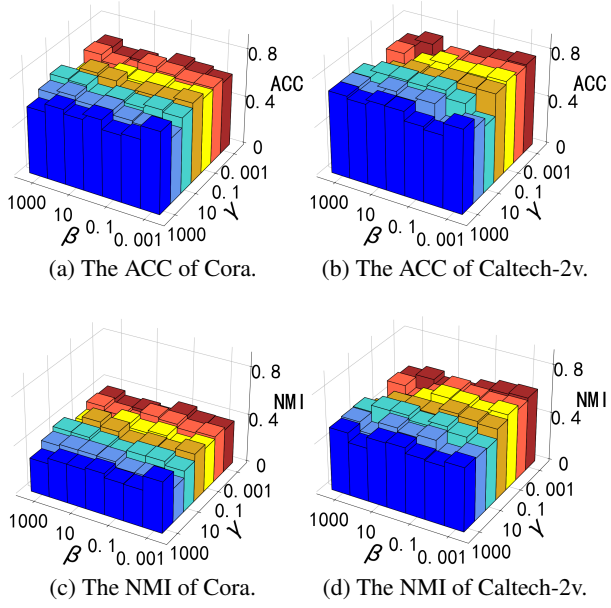
To further verify the effectiveness of our active selection module, we design an ablation study by comparing ADMC with its unsupervised and semi-supervised versions. In more detail, we denote **ADMC-U** as the unsupervised version without any annotations, which means it removes the active selection module and the supervised module. We denote **ADMC-R** as the semi-supervised version, which means in the active selection module, we randomly select K samples for querying annotations.

Table 5 shows the ACC and NMI results of our method and these two degenerated versions. The results w.r.t. ARI and PUR are shown in the Appendix. We can find that ADMC-U is much worse than ADMC-R and ADMC, even though we only have very few annotations in ADMC-R and ADMC. It shows that the supervised information is quite important for

Metrics	Method	# of annotations	Caltech-2V	Caltech-5V	CCV	Reuters	Cora	BBCSport	Scene	Noisy-Mnist
ACC	ADMC-U	0	0.2371	0.3886	0.2494	0.3858	0.3246	0.3548	0.2054	0.4758
	ADMC-R	10	0.4856	0.5000	0.1490	0.3800	0.4335	0.4926	0.2919	0.5430
		20	0.5814	0.6421	0.2479	0.4408	0.5295	0.7316	0.3530	0.7206
		30	0.6721	0.7500	0.2309	0.4633	0.4557	0.8309	0.4120	0.9469
		40	0.7093	0.8364	0.2631	0.5600	0.6226	0.8879	0.4192	0.9688
		50	0.7336	0.8529	0.2790	0.5683	0.6331	0.9062	0.4297	0.9748
	ADMC	10	0.6093	0.8086	0.1927	0.5242	0.5645	0.8199	0.3530	0.5981
		20	0.6679	0.8521	0.2486	0.5333	0.6089	0.8658	0.4312	0.9433
		30	0.7164	0.8843	0.2692	0.5842	0.6193	0.9375	0.4341	0.9688
		40	0.7521	0.8914	0.3191	0.6208	0.6448	0.9559	0.4406	**0.9866**
		50	**0.7643**	**0.9050**	**0.3325**	**0.6575**	**0.6499**	**0.9651**	**0.4591**	0.9864
NMI	ADMC-U	0	0.1107	0.3353	0.2494	0.1880	0.0956	0.0000	0.2725	0.4329
	ADMC-R	10	0.3753	0.4517	0.1147	0.1952	0.1736	0.2943	0.2934	0.5380
		20	0.4551	0.5981	0.2445	0.2159	0.2480	0.5839	0.3377	0.6826
		30	0.5341	0.6852	0.2267	0.2415	0.2593	0.6909	0.3668	0.8780
		40	0.5678	0.7578	0.2466	0.2990	0.3454	0.7455	0.3942	0.9211
		50	0.5960	0.7640	0.2441	0.2996	0.3860	0.7831	0.4004	0.9330
	ADMC	10	0.5097	0.6883	0.2194	0.2993	0.3118	0.6568	0.3560	0.6280
		20	0.5455	0.7596	0.2604	0.3039	0.3323	0.7340	0.3926	0.8903
		30	0.5811	0.7924	0.2757	0.3408	0.3416	0.8163	0.3869	0.9321
		40	0.6107	0.7991	0.2997	0.3763	**0.3659**	0.8567	0.3942	**0.9631**
		50	**0.6187**	**0.8219**	**0.3068**	**0.4049**	0.3639	**0.8899**	**0.4057**	0.9624

Table 5: The ACC and NMI results of ablation study.

the clustering. Moreover, compared with ADMC-R, ADMC often achieves better performance, even when there are only 10 annotations. It shows that our designed active selection module performs much better than the random selection no matter how many samples are annotated, demonstrating the effectiveness of the active selection module.



(a) The ACC of Cora. (b) The ACC of Caltech-2v.

(c) The NMI of Cora. (d) The NMI of Caltech-2v.

Figure 2: The ACC and NMI value of ADMC with different β and γ on Cora and Caltech-2v.

3.5 Hyper-parameter Study

To investigate the sensitivity of two hyper-parameters β and γ in ADMC, we show the ACC and NMI results with different values of β and γ in the range $[10^{-3}, 10^{3}]$. Figure 2 shows the results after 50 annotations on Cora and Caltech-2v data sets. The results on other data sets are similar. From Figure 2, we can see that the performance is stable in a relatively wide range. It means that we can easily select hyper-parameters. For example, we can select them in the range $[10^{-3}, 10^{-1}]$ to easily achieve a relatively good performance.

3.6 Conclusion

In this paper, we proposed a novel active deep multi-view clustering algorithm that can automatically select important samples for querying human annotations to guide the clustering. In this method, we provided an adaptive multi-view fusion module to ensemble multiple views by learning appropriate weights. Then, we designed an active selection module to identify representative and diverse samples for human annotations in a simple yet effective way. We used the annotations to train the network in a supervised module and we also carefully designed an unsupervised module with contrastive learning to enhance the multi-view representation learning. Experimental results on eight widely-used multi-view data sets by comparing it with state-of-the-art supervised and semi-supervised multi-view clustering methods show the effectiveness and superiority of our proposed ADMC. We also conducted the ablation study to compare it with its unsupervised and semi-supervised versions, which shows the effectiveness of our designed active selection module, and well demonstrates our motivation of active clustering.

In the future, we will further study other selection criteria, such as uncertainty and informative, and design a new active selection module for multi-view clustering.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China grants 62176001, and the Natural Science Project of Anhui Provincial Education Department under grant 2023AH030004.

References

[Amini *et al.*, 2009] Massih R Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views-an application to multilingual text categorization. *Advances in neural information processing systems*, 22, 2009.

[Chattopadhyay *et al.*, 2013] Rita Chattopadhyay, Zheng Wang, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Batch mode active sampling based on marginal probability distribution matching. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):13, 2013.

[Chen *et al.*, 2022] Rui Chen, Yongqiang Tang, Wensheng Zhang, and Wenlong Feng. Deep multi-view semi-supervised clustering with sample pairwise constraints. *ArXiv*, abs/2206.04949, 2022.

[Chen *et al.*, 2023] Jie Chen, Hua Mao, Wai Lok Woo, and Xi Peng. Deep multiview clustering by contrasting cluster assignments. *arXiv preprint arXiv:2304.10769*, 2023.

[Du *et al.*, 2021] Guowang Du, Lihua Zhou, Yudi Yang, Kevin Lü, and Lizhen Wang. Deep multiple auto-encoder-based multi-view clustering. *Data Science and Engineering*, 6(3):323–338, 2021.

[Du *et al.*, 2022] Guowang Du, Lihua Zhou, Kevin Lu, Hao Wu, and Zhimin Xu. Multiview subspace clustering with multilevel representations and adversarial regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 34:10279–10293, 2022.

[Fei-Fei and Perona, 2005] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531. IEEE, 2005.

[Fei-Fei *et al.*, 2004] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[Halder *et al.*, 2023] Bohnishikhan Halder, K. M. Azharul Hasan, Toshiyuki Amagasa, and Md. Manjur Ahmed. Autonomic active learning strategy using cluster-based ensemble classifier for concept drifts in imbalanced data stream. *Expert Syst. Appl.*, 231:120578, 2023.

[Hinton and Salakhutdinov, 2006] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[Hoi *et al.*, 2006] Steven CH Hoi, Rong Jin, Jianke Zhu, and Michael R Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424. ACM, 2006.

[Jiang *et al.*, 2011] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel Ellis, and Alexander C Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 1–8, 2011.

[Lin *et al.*, 2021a] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. Completer: Incomplete multi-view clustering via contrastive prediction. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11169–11178, 2021.

[Lin *et al.*, 2021b] Zhiping Lin, Zhao Kang, Lizong Zhang, and Ling Tian. Multi-view attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35:1872–1880, 2021.

[Liu *et al.*, 2022] Chenghua Liu, Zhuolin Liao, Yixuan Ma, and Kun Zhan. Stationary diffusion state neural estimation for multiview clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7542–7549, 2022.

[Pan and Kang, 2021] Erlin Pan and Zhao Kang. Multi-view contrastive graph clustering. In *Neural Information Processing Systems*, 2021.

[Qin *et al.*, 2021] Yalan Qin, Hanzhou Wu, Xinpeng Zhang, and Guorui Feng. Semi-supervised structured subspace learning for multi-view clustering. *IEEE Transactions on Image Processing*, 31:1–14, 2021.

[Sun *et al.*, 2022] Bicheng Sun, Peng Zhou, Liang Du, and Xuejun Li. Active deep image clustering. *Knowl. Based Syst.*, 252:109346, 2022.

[Tang and Liu, 2022] Huayi Tang and Yong Liu. Deep safe multi-view clustering: Reducing the risk of clustering performance degradation caused by view increase. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 202–211, 2022.

[Tang *et al.*, 2022] Yongqiang Tang, Yuan Xie, Chenyang Zhang, and Wensheng Zhang. Constrained tensor representation learning for multi-view semi-supervised subspace clustering. *IEEE Transactions on Multimedia*, 24:3920–3933, 2022.

[Tao *et al.*, 2017] Hong Tao, Chenping Hou, Feiping Nie, Jubo Zhu, and Dongyun Yi. Scalable multi-view semi-supervised classification via adaptive regression. *IEEE Transactions on Image Processing*, 26(9):4283–4296, 2017.

[Wang *et al.*, 2015a] Hanmo Wang, Liang Du, Lei Shi, Peng Zhou, Yuhua Qian, and Yi-Dong Shen. Experimental design with multiple kernels. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, pages 419–428. IEEE Computer Society, 2015.

[Wang *et al.*, 2015b] Hanmo Wang, Liang Du, Peng Zhou, Lei Shi, and Yi-Dong Shen. Convex batch mode active sampling via α-relative pearson divergence. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[Wang *et al.*, 2019] Hanmo Wang, Runwu Zhou, and Yi-Dong Shen. Bounding uncertainty for active batch selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5240–5247, 2019.

[Wang *et al.*, 2020] Qianqian Wang, Huanhuan Lian, Gan Sun, Quanxue Gao, and Licheng Jiao. icmsc: Incomplete cross-modal subspace clustering. *IEEE Transactions on Image Processing*, 30:305–317, 2020.

[Wang *et al.*, 2021] Shiping Wang, Zhaoliang Chen, Shide Du, and Zhouchen Lin. Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5042–5055, 2021.

[Wang *et al.*, 2022] Shiye Wang, Changsheng Li, Yanming Li, Ye Yuan, and Guoren Wang. Self-supervised information bottleneck for deep multi-view subspace clustering. *IEEE Transactions on Image Processing*, 32:1555–1567, 2022.

[Wen *et al.*, 2020] Jie Wen, Zheng Zhang, Zhao Zhang, Lunke Fei, and Meng Wang. Generalized incomplete multiview clustering with flexible locality structure diffusion. *IEEE transactions on cybernetics*, 51(1):101–114, 2020.

[Whang *et al.*, 2020] Joyce Jiyoung Whang, Rundong Du, Sangwon Jung, Geon Lee, Barry L. Drake, Qingqing Liu, Seonggoo Kang, and Haesun Park. Mega:multi-view semi-supervised clustering of hypergraphs. *Proceedings of the VLDB Endowment*, 13:698 – 711, 2020.

[Wu *et al.*, 2023] Zhihao Wu, Xincan Lin, Zhenghong Lin, Zhaoliang Chen, Yang Bai, and Shiping Wang. Interpretable graph convolutional network for multi-view semi-supervised learning. *IEEE Transactions on Multimedia*, 2023.

[Xia *et al.*, 2021] Wei Xia, Qianqian Wang, Quanxue Gao, Xiangdong Zhang, and Xinbo Gao. Self-supervised graph convolutional network for multi-view clustering. *IEEE Transactions on Multimedia*, 24:3182–3192, 2021.

[Xiao *et al.*, 2023] Shunxin Xiao, Shide Du, Zhaoliang Chen, Yunhe Zhang, and Shiping Wang. Dual fusion-propagation graph neural network for multi-view clustering. *IEEE Transactions on Multimedia*, 2023.

[Xie *et al.*, 2020] Yuan Xie, Bingqian Lin, Yanyun Qu, Cuihua Li, Wensheng Zhang, Lizhuang Ma, Yonggang Wen, and Dacheng Tao. Joint deep multi-view learning for image clustering. *IEEE Transactions on Knowledge and Data Engineering*, 33:3594–3606, 2020.

[Xu *et al.*, 2021] Jie Xu, Yazhou Ren, Huayi Tang, Xiaorong Pu, Xiaofeng Zhu, Ming Zeng, and Lifang He. Multi-vae: Learning disentangled view-common and view-peculiar visual representations for multi-view clustering. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9214–9223, 2021.

[Xu *et al.*, 2022a] Jie Xu, Chaozhuo Li, Yazhou Ren, Liang Peng, Yujie Mo, Xiaoshuang Shi, and Xiao lan Zhu. Deep incomplete multi-view clustering via mining cluster complementarity. In *AAAI Conference on Artificial Intelligence*, 2022.

[Xu *et al.*, 2022b] Jie Xu, Huayi Tang, Yazhou Ren, Liang Peng, Xiaofeng Zhu, and Lifang He. Multi-level feature learning for contrastive multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16051–16060, 2022.

[Xue *et al.*, 2021] Zhe Xue, Junping Du, Changwei Zheng, Jie Song, Wenqi Ren, and Meiyu Liang. Clustering-induced adaptive structure enhancing network for incomplete multi-view data. In *International Joint Conference on Artificial Intelligence*, 2021.

[Yan and Huang, 2018] Yifan Yan and Sheng-Jun Huang. Cost-effective active learning for hierarchical multi-label classification. In *IJCAI*, pages 2962–2968, 2018.

[Yan *et al.*, 2023] Weiqing Yan, Yuanyang Zhang, Chenlei Lv, Chang Tang, Guanghui Yue, Liang Liao, and Weisi Lin. Gcfagg: Global and cross-view feature aggregation for multi-view clustering. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19863–19872, 2023.

[Yang *et al.*, 2023] Xihong Yang, Jin Jiaqi, Siwei Wang, Ke Liang, Yue Liu, Yi Wen, Suyuan Liu, Sihang Zhou, Xinwang Liu, and En Zhu. Dealmvc: Dual contrastive calibration for multi-view clustering. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 337–346, 2023.

[Zhang *et al.*, 2019] Changqing Zhang, Zongbo Han, Yajie Cui, Huazhu Fu, Joey Tianyi Zhou, and Qinghua Hu. Cpm-nets: Cross partial multi-view networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 557–567, 2019.

[Zhou and Du, 2023] Peng Zhou and Liang Du. Learnable graph filter for multi-view clustering. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, pages 3089–3098. ACM, 2023.

[Zhou *et al.*, 2019] Peng Zhou, Yi-Dong Shen, Liang Du, Fan Ye, and Xuejun Li. Incremental multi-view spectral clustering. *Knowl. Based Syst.*, 174:73–86, 2019.

[Zhou *et al.*, 2023] Peng Zhou, Bicheng Sun, Xinwang Liu, Liang Du, and Xuejun Li. Active clustering ensemble with self-paced learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2023.

[Zhu and Gao, 2022] Zhaorui Zhu and Quanxue Gao. Semi-supervised clustering via cannot link relationship for multiview data. *IEEE Transactions on Circuits and Systems for Video Technology*, 32:8744–8755, 2022.