Applied Intelligence
https://doi.org/10.1007/s10489-021-03129-0

Efficient multi-view clustering networks

Guanzhou Ke[1] · Zhiyong Hong[1] · Wenhua Yu[1] · Xin Zhang[1] · Zeyi Liu[1]

Accepted: 19 December 2021

Abstract
In the last decade, deep learning has made remarkable progress on multi-view clustering (MvC), with existing literature adopting a broad target to guide the network learning process, such as minimizing the reconstruction loss. However, despite this strategy being effective, it lacks efficiency. Hence, in this paper, we proposed a novel framework, entitled Efficient Multi-view Clustering Networks (EMC-Nets), which guarantees the network's learning efficiency and produces a common discriminative representation from multiple sources. Specifically, we developed an alternating process, involving an approximation and an instruction process, which effectively stimulate the process of multi-view feature fusion to force network to learn a discriminative common representation. The approximation process employs a standard clustering algorithm, i.e., k-means, to generate pseudo labels corresponding to the current common representation, and then it leverages the pseudo labels to force the network to approximate a reasonable cluster distribution. Considering the instruction process, it aims to provide a correct learning direction for the approximation process and prevent the network from obtaining trivial solutions. Experiment results on four real-world datasets demonstrate that the proposed method outperforms state-of-the-art methods. Our source code will be available soon at https://github.com/Guanzhou-Ke/EMC-Nets.

Keywords Clustering · Multi-view learning · Self-supervised

Introduction Clustering is a fundamental task in machine learning, with most current works studying the single-view clustering case. However, utilizing a single view does not thoroughly represent the original object features. In fact, in real-world scenarios, data are frequently represented as multiple views or modalities. Given the diversity of feature extraction methods, these different views are usually governed by various statistical properties. For instance, an image may be characterized by various descriptors, such as SIFT [1], histograms of oriented gradients (HoG) [2], and local binary pattern (LBP) [3]. While in Online Video Analysis (OVA), the video can be split into visual and audio signals. Nevertheless, it is not easy to directly transfer a single view cluster to a multi-view setup, with the significant concern attributed to two aspects [4]: consistency and complementary information between the views. This is because simply concatenating each view's information instead of utilizing specific technology to fuse the views is likely to introduce additional noise and reduce the generalization capability. Hence, numerous multi-view clustering methods have been proposed exploring the common representation among multiple sources.

In the past two decades, most multi-view clustering algorithms focus on exploring consistency representation among views. A unified representation is of vital importance for downstream tasks, e.g., Anomaly Detection or OVA. Existing methods involve applying multi-view data utilizing a non-negative matrix factorization scheme [5] to extract the common representation factor among multiple views by exploiting joint matrix factorization and normalization procedures. Binary Multi-View Clustering (BMVC) [6] obtains the common binary code space of large-scale multi-view images by unifying a compact collaborative discrete representation and a binary clustering structure. BMVC can complete large-scale image clustering while ensuring efficiency and low computing resource requirements. However, its performance excessively relies on the initialization parameters employed. To solve partial parameter dependence, [7] proposes a novel learning paradigm, namely geometric consistency (GC) and cluster assignment consistency. GC aims to learn data connection, i.e., data points that belong to the

✉ Zhiyong Hong
hongmr@163.com

[1] Department of Intelligent Manufacturing, Wuyi University, 529020 Jiangmen, China

Published online: 10 January 2022

same cluster and minimize the discrepancy of pairwise connection among different views. Although [7] automatically determines the clustering size, it cannot effectively extract the non-linear and complex data information, and the data size limits its clustering capacity. Another representative way of extracting common representation is the Canonical Correlation Analysis (CCA) [8], which explores the statistical properties between two views, searching for two projections to map the two views onto a low-dimensional common representation where the linear correlation between these two views is maximized.

However, CCA is not able to explore the non-linear correlation of complex data. Kernel CCA (KCCA) [9] introduces kernel techniques to overcome the linear limitation. Nevertheless, different views require utilizing different kernels that require expert knowledge. With the recent flourish of deep learning, neural network architecture is widely used to extract non-linear features among different views automatically. Deep Canonical Correlation Analysis (DCCA) [10] is a deep network-based extension algorithm of CCA. It leverages neural networks to extract non-linear features from two views, then it applies a linear CCA layer to project these features onto a low-dimensional space. Furthermore, in [11], the researchers combine deep learning and matrix factorization to solve the MvC problem. This strategy learns the hierarchical semantics of multi-view data similarly to a deep neural network (DNN) while providing the class information in the last layer by capturing the geometric structure of each view. A limitation of this method is the high computation resource demand when meeting large-scale data. An alternative solution is proposed in [12], where the subspace is clustered, and DNN is exploited to develop an affinity fusion layer that explores shared affinity across all the views. A downside of this technique is imposing a huge parameter matrix to maintaining the affinity matrix.

A recent study [13] argued that only extracting correlation (consistence) and ignoring independence (complementarity) cannot accurately measure a real object, i.e., a good (comprehensive) common representation should contain as much information as possible about the consistency among views and also should contain unique information about different views. Comprehensive representation enhances the model's robustness because models with different tasks and preferences can flexibly learn the required high-level semantics from comprehensive representation. In contrast, it is difficult to explore the comprehensive semantics with previous methods, which only extract consistent information. The work of [14] integrates the Locally Linear Embedding (LLE) and Laplacian EigenMaps (LE) schemes to extract the potential comprehensive representation among multiple views and achieve the migration from the view-space to the task-space. Literature in [15, 16] has made remarkable progress in exploring comprehensive representation on multiple views by employing the reconstruction function as the main body of the objective function and adopt Generative Adversarial Networks (GANs) [17] as a constraint to force the model learning a discriminative common representation.

Although the existing methods can achieve extremely high performance with additional constraints, they are still based on a broad learning target, i.e., reconstruction, with the latter is a vital learning target for unsupervised learning. However, the lack of a clear learning direction, e.g., labels in supervised learning, provides a reconstruction function that is effective but inefficient as an unsupervised loss (the reconstruction weaknesses are presented in Section 3.3. To address this issue, we extend the pseudo label generation technique of [18, 19] and provide a clear target for the network to learn discriminative MvC features. Furthermore, we highlight that pseudo label generation with low-quality initialization degrades generalization performance, as it lacks providing the correct direction generation and imposes an error correction mechanism during label generation. This work introduces the reconstruction function as the instructor of learning direction and an error correction mechanism to avoid a trivial pseudo label generation solution. Therefore, we propose the Efficient Multi-view Clustering Networks (EMC-Nets), which aims to efficiently guarantee the efficiency of network learning and produce a discriminative common representation. Concretely, as shown in Fig. 1, the common representation can be extracted from data originating from different views utilizing a feature extractor. The latter comprises the view-specific encoder and the attention fusion layer, which consists of the standard transformer encoder [20]. Then, the clustering module and supervisor module leverage common representation to generate the pseudo labels and reconstruct the corresponding views, respectively.

The essential advantage of our model lies in the simple network architecture and alternating training process. Considering the training dynamics, the EMC-Nets involve two main processes: the instruction process and the approximation process. The instruction process establishes a correct direction of generation pseudo labels corresponding to the current common representation. The approximation process generates pseudo labels to force the network to approximate a reasonable cluster distribution. The main contributions of this paper are summarized as follows:

1. We propose a novel unsupervised multi-view clustering framework entitled Efficient Multi-view Clustering Networks (EMC-Nets), which efficiently integrates multiple views into a discriminative common representation and flexibly utilizes different clustering algorithms such as k-means or spectral clustering to adapt to real task requirements.

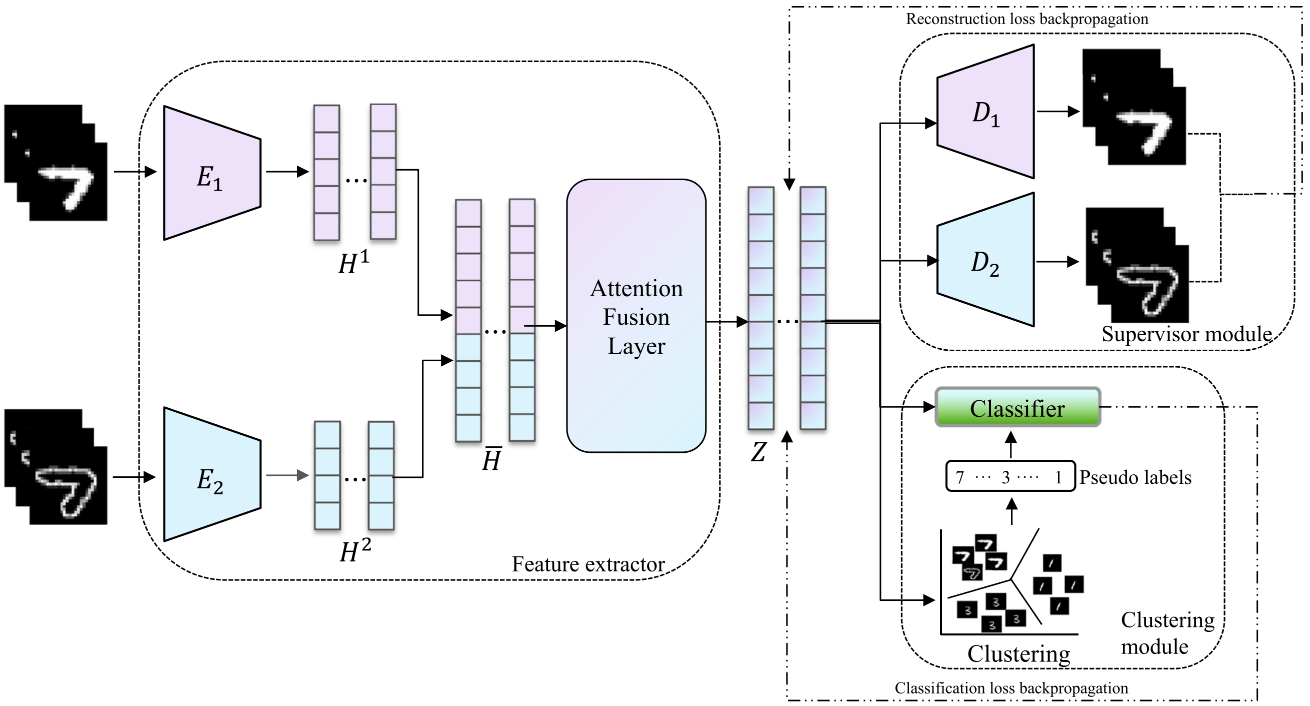


Fig. 1 Illustration of the Efficient Multi-view Clustering Network(EMC-Nets). EMC-Nets consists of the feature extractor, supervisor, and clustering modules. The feature extractor consists of V view-specific encoder networks E and an attention fusion layer. It outputs a specific low-dimensional standard representation Z that reconstructs each view in the supervisor module and generates pseudo labels through the clustering module. These pseudo labels are used to further enhance the feature extractor by the classifier.

2. We develop a robust alternating training process, involving an approximation process and an instruction process, to stimulate the process of multi-view feature fusion to obtain a comprehensive common representation. The instruction process, which adopts a reconstruction strategy, aims to generate pseudo labels. The approximation process uses pseudo labels to improve the feature extractor. These two processes are naturally complementary, ensuring the efficiency of unsupervised learning. Experimental results show that the proposed instruction process can effectively improve the quality of pseudo labels compared to [19].
3. We conduct experiments on four real-world datasets to measure the performance and convergence of the alternating process. Our experiments demonstrate that after a period of oscillation, the two processes are compatible with each other until they converge. In the comparative experiment, the representation learned by EMC-Nets outperforms state-of-the-art methods in terms of performance and effectiveness.

The remainder of the paper is as follows. Section 2 reviews related work, including self-supervised learning and multi-view clustering algorithms. Section 3 introduces the details of the proposed framework, while Section 4 presents the experimental results demonstrating the effectiveness and superiority of EMC-Nets against current methods on four real-world datasets. Finally, Section 5 discusses our findings and Section 6 concludes this work.

1 Related work

This section shortly introduces related works to pseudo label generation and multi-view clustering. Given that our work extends [19] to facilitate multi-view clustering. Section 2.1 highlights the differences between [19] and introduces the suggested EMC-Nets. Additionally, we also depict the difference between our work and current solutions in the field of MvC.

1.1 Pseudo label generation

Pseudo label generation is a vital sub-task in self-supervised learning [21]. that is widely used in the field of single-view clustering [18, 19, 22]. Simply expanding solutions from single-view to multi-view does not pose optimum clustering performance, while to the best of our knowledge, a limited number of papers consider a pseudo label generation technique to address the MvC problem. Given this research gap, we extend [19] to facilitate an MvC solution. The method in [19] utilizes batch k-means to iteratively generate the suitable pseudo label and then treats these labels to enhance the feature extractor similar to supervised learning. Managing an efficient pseudo label generation affords significant progress in unsupervised visual representation. However, [19] suffers from the pseudo label generation quality, and specifically, if the quality is poor, then the features learned later will also be poor. Due to the characteristics of the gradient descent algorithm, this is an irreparable issue in [19]. To address this issue, we introduce the supervisor module to reconstruct the original input and provide the correct direction during the pseudo labels learning process. Current literature offers some similar work to ours. For example, [18] designs dynamic memory modules to solve the inferior clustering assignment problem. This method stores the pseudo labels, features, and clustering centroids of each batch, and then the network perceives the global information to update the parameters and re-assign the clustering centroids. Despite this being a viable pseudo-label generation solution, the two additional memory modules impose a substantial computational burden. The work of [22] suggests a structure and attribute information fusion module to address graph clustering. It leverages the fusion graph information to generate target distribution instead of pseudo labels, but this method only deals with the graph clustering problem lacking generalization to other data forms, i.e., image and text. In contrast, the suggested EMC-Nets can theoretically deal with any form of data.

1.2 Multi-view clustering

The MvC algorithm taxonomy involves two categories: traditional methods and deep learning methods. Traditional MvC methods rely on statistical theory, e.g., multi-kernel learning methods [23, 24], where the characteristics of different views can be non-linearly mapped into a common representation. Nevertheless, these methods require selecting the appropriate kernel for each view carefully. Subspace-based methods [25–27] project the multi-view data into a low-dimensional shared subspace to obtain the standard representation. There are also methods exploiting non-negative matrix factorization [5] and graph-based models [28]. These statistical models have a common drawback, as they cannot extract complex structures within the data. Therefore, there have been many deep methods combined with traditional methods, such as [10, 12, 29]. These extract higher semantic features through neural networks and then apply on these features statistical models to extract a common representation. In recent years, complete deep methods have been proposed, such as [15, 16]. These methods have a more vital ability to express a data structure, obtaining higher performance on many benchmark datasets. These methods consider the reconstruction function as the main body of the objective function, allowing the network to understand the data. However, as discussed in Section 3.3, the learning range of the reconstruction function is too broad, and thus these methods utilize other constraints, e.g., adversarial constraints, to limit the learning range of the network. This strategy enlarges the model size, prohibiting practical applications. Hence, in this work, we reconsider network learning and propose a lightweight and efficient MvC framework described in detail in the next section.

2 Method

Consider the problem of clustering a set of n data points consisting of V views $\mathcal{D} = \{\mathbf{X}^1, \mathbf{X}^2, ..., \mathbf{X}^V\}$ into c clusters, where $\mathbf{X}^v \in \mathbb{R}^{d_v \times n}$ denotes the samples of the dimension d_v from the v-th views. In this work, we suggest an efficient deep multi-view clustering network to solve the aforementioned clustering problem. This section introduces the feature extractor and attention fusion layer and then depicts the clustering module and the process of pseudo labels generation. Next, we introduce the supervisor module and demonstrate its ability to avoid trivial pseudo-label generation solutions. Finally, we describe the proposed loss function and training procedure of EMC-Nets.

2.1 Feature extractor

This subsection introduces the feature extractor component and describes in detail the proposed attention fusion mechanism. Due to the diversity of the different views, the feature extractor, consisting of V view-specific encoder networks, transforms the data into low-dimensional latent space, while an attention [20] fusion layer fuses each specific latent to the joint representation (Fig. 1). Firstly, for the v-th view, we handle the encoder to extract the corresponding representation as $\mathbf{H}^v = E_v(\mathbf{X}^v;\theta_e^v)$, where $E_v(\cdot)$ refers to the v-th view's encoding network parameterized by θ_e^v. We utilize the fully-connected layer within the encoder's structure and then, we obtain $\bar{H}$ by concatenating each H^vas presented below:

$$\bar{H} = [H^1, H^2, ..., H^v] \tag{1}$$

where $[\cdot]$ denotes the concatenation operator.

The attention fusion layer is designed to fuse diverse information from the various views with the fusion mechanism presented in Fig. 2. According to the above strategy, we aim for the joint representation Z to consider the comprehensive information from each view. Based on this concept, we exploit the attention mechanism [20] to enhance each feature in $\bar{H}$ by using other features. Thus, first, we use $[W_Q, W_K, W_V]\bar{H}^T$ to obtain the *Query* Q, the *Key* K, and the *Value* V matrix, respectively. Then, the similarity matrix of Q and K are calculated through the $softmax\left(\frac{QK^T}{\sqrt{d}}\right)$, where d

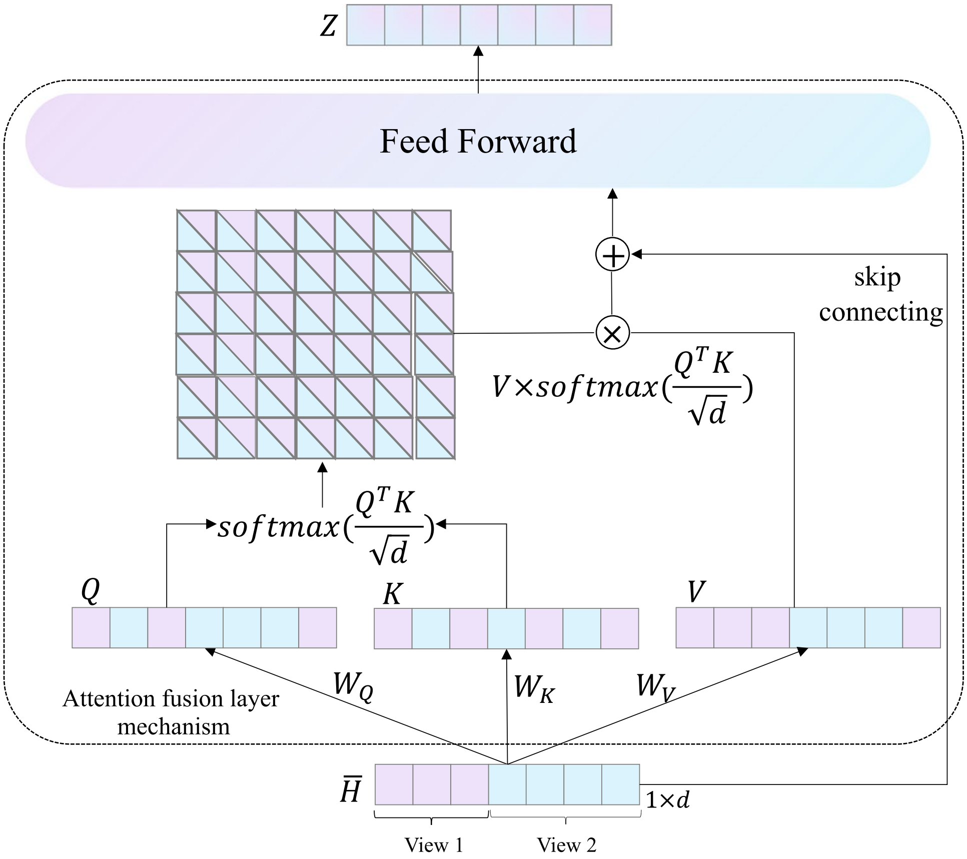

Z
Feed Forward
skip connecting
$V \times softmax(\frac{Q^T K}{\sqrt{d}})$
$softmax(\frac{Q^T K}{\sqrt{d}})$
Q
K
V
W_Q
W_K
W_V
Attention fusion layer mechanism
$\bar{H}$
$1 \times d$
View 1
View 2

Fig. 2 Illustration of the attention fusion mechanism. For convenience, we demonstrate that the input is one sample with d-dimensional feature. Note: we use a vector to demonstrate the attention fusion process in this example, but in practice, the $\bar{H}, Q, K, V, Z$ represent a matrix. Hence, we adopt $softmax(\frac{QK^T}{\sqrt{d}})V$ as the attention fusion core equation

is the feature dimension of $\bar{H}$. We believe that the similarity matrix is an essential factor for the success of the attention fusion method because, intuitively, it can provide a sufficient basis for each feature during the information fusion process. Next, we assign the information to be fused according to the similarity matrix utilizing $softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$. Finally, the fused information is added to the original input feature $\bar{H}$, and the output is sent to the fully-connected layer to obtain the comprehensive representation Z. For convenience, we denote the fusion process as the following formulas:

$$Z = Attn(\bar{H};\theta_\alpha) \tag{2}$$

where $Attn(\cdot)$ is a standard *TransformerEncoder* [20], as shown in Fig. 2, parameterized by θ_α. Finally, the standard representation Z is input to the clustering module and supervisor module, respectively.

2.2 Clustering module

This part discusses how to iteratively group the standard representation Z and leverage the pseudo labels to enhance the feature extractor. In [19], the authors employ standard clustering algorithms, such as k-means, to generate pseudo labels for each batch sample in the current feature space. Additionally, depending on the data properties, the clustering method can be replaced with other more appropriate clustering methods, such as spectral clustering. In this work, we extend the idea of generating pseudo labels [19] into the MvC domain, as [19] only deals with a single-view setup, and simply applying it on MvC does not pose an optimum solution (as discussed in Section 2.1). However, by exploiting the standard representation Z introduced in Section 3.1, it is feasible to extend the concept of [19] to MvC utilizing the following formula:

$$\min_{C\in\mathbb{R}^{d\times k}} \frac{1}{N}\sum_{i=1}^{N} \min_{y_i\in\{0,1\}^k} ||Z_i - C_{y_i}||_F^2 \quad \text{such that} \quad y_i^T 1_k = 1 \tag{3}$$

where $||\cdot||_F$ is the Frobenius norm, C represents a $d \times k$ centroid matrix, and y_i is the clustering assignment to each sample. By solving (3), we can cluster each common representation Z_i into k groups. After that, we obtain a series of optimal assignments $(y_i^*)_{i\leq N}$ and a centroid matrix C^*, which. These are then used as pseudo labels.

Considering a traditional supervised classification problem, the network $G(\cdot)$ can be generally divided into two parts: the feature extractor and the classifier. The parametrized classifier predicts the correct labels on top of the

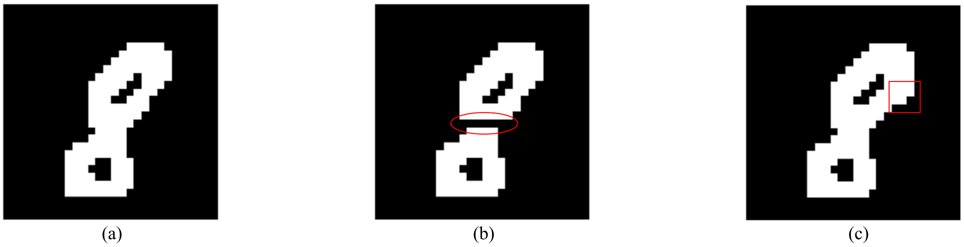


Fig. 3 Illustration of the limitation of reconstruction loss. (**a**) is the digit 8 from MNIST. In order to simplify the problem, we processed the image into a binary pixel map. We removed 6 pixels from (**b**) and added 6 pixels in (**c**). Obviously, (**b**) is not a recognized digit 8, but (**c**) can still be identified. Nevertheless, (**b**) and (**c**) have the same error in reconstruction loss

feature extractor, while the parameters θ_g of the network $G(\cdot)$ is learned by optimizing the following problem:

$$\min_{\theta_g} \frac{1}{N} \sum_{i=1}^{N} \ell(G(x_i), y_i) \tag{4}$$

where $\ell(\cdot)$ is a logistic loss, also known as the negative log-softmax function. Equation (5) is a straightforward metric to measure the distinction between the network's prediction and the ground truth distribution. In [19], the author applies (3) and (5) as an unsupervised network joint learning goal, which in this paper we extend to the case of MvC. Therefore, our method defines as a reconstruction function a more precise learning objective than the broad objective function of [19], defined as:

$$\mathcal{L}_{cls} = \min_{\theta_e^v, \theta_g, \theta_\alpha} \frac{1}{N} \sum_{i=1}^{N} \ell(G(Z), y_i) \tag{5}$$

where $G(\cdot)$ represents a linear classification neural network and y_i are the pseudo labels calculated by (3). Also, (5) is minimized using mini-batch stochastic gradient descent [30] and backpropagation to compute the gradient.

In short, EMC-Nets leverage the common representation Z to iteratively produce pseudo labels using (3), and updating the parameters of $E_v(\cdot;\theta_e^v)$ by predicting these pseudo labels using (5). However, this procedure is prone to trivial solutions, and thus in the following section, we describe how to avoid them.

2.3 Supervisor module

Although training the network with pseudo labels enhances the network's feature learning efficiency, it highly depends on the quality of pseudo labels. If their initial quality is poor, then the features learned later will also be poor. Therefore, in this part, we first discuss two trivial solutions that are often criticized: the unsupervised method with reconstruction being the primary learning goal is easy to learn trivial knowledge, and the pseudo label generation quality is highly dependent on the initial value. Then we explain how EMC-Nets solves these issues.

Reconstruction loss is one of the mainstream training indicators in unsupervised learning. It compresses the original information into a bottleneck representation and leverages these representations to reconstruct the original information. Due to its conceptual simplicity, reconstruction has a wide range of applications in networks utilizing an Auto-encoder [31] as the backbone. It can effectively measure the difference between the network's output and input but neglects the local structure causing the network to provide a trivial solution. For example, in Fig. 3, digit 8 in some parts is modified to simulate network reconstruction. We found that some modifications such as Fig. 3(c) are not crucial for recognition, but some changes, such as Fig. 3(b), may directly lead to indistinguishability. Although humans or some state-of-the-art classifiers can easily identify those insignificant changes and give the correct answer, this is challenging for the reconstruction loss. Therefore, for MvC type of problems, some methods introduce additional network modules to add constraints, i.e., [15, 16] use adversarial loss [17] as reconstruction constraints. However, the ability of these constraints to guide the network is limited, and lack of accurate guidance.

On the other hand, (5) indicates that a likelihood function can maximize the pseudo label distribution and the current cluster distribution. Therefore, if the initial pseudo labels contain many wrong labels, (5) will continue to magnify this error. As the number of training instances increases, the network will continue to accumulate error knowledge and eventually lead to an inferior generalization performance. This is caused due to, first, utilizing batch training methods prohibits the network from perceiving global information, and second, there is no additional error correction mechanism.

Therefore, we comprehensively consider the advantages and disadvantages of reconstruction and pseudo label generation and design a supervisor module combined with an alternate training strategy. The supervisor module is comprises V view-specific decoder networks, which considers the standard representation Z as input and reconstructs it into the V original views. This mapping could be represented as $\bar{X}^v = D_v(Z;\theta_d^v)$, where D_v refers to v-th view's decoding network parameterized by θ_d^v. Hence, we describe the loss function of supervisor module as:

$$\mathcal{L}_{recon} = \min_{\theta_e^v,\theta_d^v,\theta_\alpha} \sum_{v=1}^{V} ||D_v(Z), X^v||_F^2 \tag{6}$$

Overall, we believe that these two modules (clustering and supervisor) can naturally overcome the shortcomings of each other. Concretely, the clustering module (or pseudo labels) can provide explicit learning goals for the network and is more efficient in learning discernible features than the traditional unsupervised methods. On the other hand, the supervisor module (or reconstruction) can provide the correct direction to the clustering module and generate relatively correct pseudo labels.

2.4 Training procedure

This subsection discusses the EMC-Net training strategy, i.e., alternating process. In general, this alternating process is divided into two sub-processes: the approximation (led by a clustering module) and the instruction (led by a supervisor module). Finally, we discuss the convergence conditions of the alternating process and present the detailed training procedure (Algorithm 1). Concretely, we depict the alternating process as follows:

1. **Approximation process**: This process utilizes the pseudo labels generated by the clustering module to train the feature extractor, and then the network parameters are updated through (5). This strategy provides the network an explicit learning direction and improves the clustering performance by fitting pseudo labels to the greatest extent.
2. **Instruction process**: In this process, reconstruction is adopted as a network learning target to prevent the network from being trapped in a trivial solution. The instruction process updates the parameters of the feature extractor and the supervisor module through (6) and provides an appropriate generation direction for the cluster assignment to generate the corresponding pseudo labels based on the joint representation Z learned by the current network. The clustering module is quite flexible in terms of the clustering algorithm that can be utilized. To simplify the problem, for this module, we employ k-means as the clustering algorithm.

As shown in Algorithms1, we initially execute the instruction process to provide a good initialization for the network, and then the clustering module generates the pseudo labels to enhance the feature extractor through an approximation process. To evaluate the convergence capability of the alternating process, we define a general convergence condition, i.e., when the mutual information ratio of pseudo labels of any two continuous time-stamps is close enough (the difference is a small value ϵ) and remains unchanged for a while, we consider that convergence is reached. We provide a method to verify the convergence of the alternating process in practice during the alternating process, i.e., perform adequate training epochs and record the number of times that the mutual information ratio is continuously less than g. If this number reaches a certain threshold (default is 10), we consider that the alternate process has converged. These processes are performed alternately until convergence, and ultimately we obtain the optimal clustering result, i.e., pseudo labels.

Algorithm 1 Pseudocode of training EMC-Nets.

```
Input: Batch multi-view data $\mathcal{D}_b = \{\mathbf{X}_b^1, \mathbf{X}_b^2, ..., \mathbf{X}_b^V\} \in \mathcal{D}$; The error factor $\epsilon$, the number of
    cluster $k$, instruction process steps $r$ and approximation process steps $t$;
Output: Cluster assignment vector $A$
repeat
    for $r$ steps do
        Train the model by reconstruction and update network through Eq. 6
    end for
    $old\_label \leftarrow new\_label$;
    $new\_label \leftarrow ClusterAssignment(\mathcal{D}_b, k)$;
    // Calculate the ratio of Normalized Mutual Information (NMI) between old label and
    new label.
    $nmi\_ratio \leftarrow NMI(old\_label, new\_label)$
    for $t$ steps do
        Train the feature extractor of model by $new\_label$ and update network through Eq.
    5
    end for
until $|1 - nmi\_ratio| < \epsilon$
```

Table 1 Dataset Description

Dataset	type	#samples	#view	#class
BDGP	image-text	2,500	2	5
HW	image	2,000	6	10
CCV	video	6,773	3	20
MNIST	image	70,000	2	10

3 Experiments

Several experiments are conducted to evaluate the performance of EMC-Nets on various datasets. Specifically, firstly, we analyze the clustering performance of EMC-Nets and challenge it against current algorithms on real datasets. The comparative experiments on different dataset scales utilizing statistical and depth methods indicated that EMC-Nets perform better on pure visual tasks than hybrid view tasks. Furthermore, we study the convergence of EMC-Nets on different datasets and the impact of different hyper-parameters on clustering performance. Finally, we study the impact of different fusion methods and independent approximation and instruction processes on the final performance. The corresponding results highlight that the attention-based approach is better than other fusion strategies. However, the most critical finding is that the alternating process is more appealing than the independent process through the split experiment. To enhance readability, the clustering results of EMC-Nets are visualized.

3.1 Experimental Settings

This subsection describes four multi-view datasets, the evaluation metrics and the comparison methods utilized, the implementation details, and the parameter settings for the algorithms employed during the trials. In particular, to distinguish the methods of different characteristics, the competitor methods are divided into three categories, as shown in Table 3, i.e., single-view clustering, traditional statistical multi-view algorithms, and based on deep learning. The proposed EMC-Nets belong to the last category.

3.1.1 Datasets

To demonstrate the performance of the proposed framework, all methods are challenged on four multi-view benchmark datasets divided into three categories based on their data type. A brief dataset description is presented in Table 1.

1. Image dataset: Handwritten numerals (HW) dataset [32] consists of 2,000 samples from 0 to 9 ten digit classes and each class has 200 samples. In our experiment, we define view-1 being the 76 Fourier coefficients and view-2 being the 240 pixel averages in 2×3 windows; MNIST is a large-scale and widely-used benchmark dataset consisting of 70,000 handwritten digit images with 28×28 pixels. We adopt its advanced version provided by [33]. In our experiment, we treat view 1 as the original gray images and view 2 as the corresponding digit edge.
2. Image and text dataset: Berkeley Drosophila Genome Project (BDGP) [34] consists of 2, 500 data points about drosophila embryos belonging to 5 categories. Each data point is represented by a 1,750-D visual vector and a 79-D textual feature vector. We set the visual data as view 1 and textual data as view 2 in our experiment.
3. Video dataset: The Columbia Consumer Video (CCV) [35] dataset consists of 9, 317 YouTube videos with 20 diverse semantic categories. In our experiment, we adopt the available subset 6, 773 videos of CCV, along with three hand-crafted features: STIP features with 5, 000 dimensional Bag-of-Words (BoWs) representation, SIFT features extracted every two seconds with 5, 000 dimensional BoWs representation, and MFCC features with 4, 000 dimensional BoWs representation.

3.1.2 Evaluation metrics

The clustering performance is measured using three standard evaluation matrices, i.e., clustering Accuracy (ACC), Normalized Mutual Information (NMI), and Purity. We present these metrics as shown in (7), (8), and (9), respectively, where Y represents ground-truth labels and C denotes clustering labels; $\delta(\cdot,\cdot)$ is the indicator function; $map(\cdot)$ is the mapping function corresponding to the best one-to-one assignment of clusters to labels implemented by the Hungarian algorithm [48]; $I(\cdot;\cdot)$ and $H(\cdot)$ represent mutual information and entropy functionals, respectively. For further details on these evaluation metrics, the reader is referred to [36]. It should be noted that the validation process of the clustering methods involves only the cases where ground truth labels are available.

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(c_i))}{n} \tag{7}$$

$$NMI = \frac{I(Y;C)}{\frac{1}{2}(H(Y) + H(C))} \tag{8}$$

$$Purity = \frac{1}{N} \sum_{k} \max_{j} |Y_k \cap C_j| \tag{9}$$

3.1.3 Comparison models

The performance of the suggested EMC-Nets is challenged against:

1. Single-view Clustering (SC). The standard spectral clustering algorithm [37] is conducted on each view and the concatenated view.
2. Traditional (Statistics) Methods. 1) Robust Multi-view K-Means Clustering (RMKMC) [38] searches a consensus cluster indicator across multiple views. 2) Robust Multi-view Spectral Clustering (RMSC) [39] recovers a latent transition probability matrix from pair-wise similarity matrices of each view through a low-rank constraint. 3) Multiview Consensus Graph Clustering (MCGC) [40] learns a consensus graph with minimizing disagreement between different views and constraining the rank of the Laplacian matrix. 4) Auto-weighted Multiple Graph Learning (AMGL) [41] utilizes every single view to construct a graph and learns an optimal weight for each graph automatically with parameter-free. 5) Cross-view Matching Clustering (COMIC) [7] generates a view-specific connection graph to obtain view-consensus information, automatically measuring the cluster size. 6) Binary Multi-View Clustering (BMVC) [6] leverages the unifying compact collaborative discrete representation and binary clustering structure to obtain the common binary representation.
3. Deep Learning Methods. 1) Deep Canonical Correlation Analysis (DCCA) [10] learns nonlinear transformations of two views such that the representation are highly linearly correlated. 2) Deep Canonically Correlated Auto-encoders (DCCAE) [29] is an advanced version of DCCA. . It uses two auto-encoders to replace the fully connected network in DCCA and optimizes the canonical correlation between the learned bottleneck representations and the reconstruction errors of the auto-encoders. 3) Deep Multi-modal Subspace Clustering (DMSC) [12] presents convolutional neural network-based approaches for unsupervised multi-modal subspace clustering. 4) Deep Adversarial Multi-view Clustering (DAMC) [15] employs shared deep auto-encoders to learn latent representations across multiple views while leveraging adversarial networks to capture data distribution further. 5) Multi-View Clustering via Deep Matrix Factorization (MCDMF) [11] combines deep learning and matrix factorization to explore the hierarchical semantics of multi-view data and output the class information in the last layer by capturing the geometric structure in each view.

To compare the divergence of the above algorithms based on their type, these are divided into three categories, namely single-view clustering methods, multi-view clustering methods based on statistical models, and deep learning models, as shown in Table 3. We verify the performance of all algorithms on medium-scale datasets, while for the large-scale datasets evaluation, we involve only the algorithms that meet the requirements shown in Table 4.

Table 2 Hyper-parameters selection for all competitors

Model	Search range	Step	Best on (BDGP, HW, CCV, MNIST)
RMKMC	$\gamma \in [0.1, 2]$	0.1	(0.5, 0.5, 0.2, -)
RMSC	$\lambda \in [10^{-3}, 1]$	0.001	(0.01, 0.01, 0.005, -)
MCGC	$\beta \in [0, 10^{2}]$	0.1	(0.6, 0.6, 10, -)
COMIC	–		–
BMCV	$\gamma, \lambda \in [10^{-6}, 10^{-3}]$, $\beta, r \in [0, 10]$	10^{-6}, 0.1	$\gamma = 10^{-4}, \lambda = 10^{-5}$, $\beta = 0.5, r = 5$
AMGL	–		–
DCCA	$\lambda \in [10^{-4}, 0.1]$	10^{-4}	0.001
DCCAE	$\lambda \in [10^{-4}, 0.1]$	10^{-4}	0.001
DMSC	$\lambda_1, \lambda_2 \in [-10, 10]$	1	$\lambda_1 = \lambda_2 = 1$
MCDMF	$\gamma, \beta \in [10^{-2}, 1]$	0.01	$\gamma = 0.5, \beta = 0.1$
DAMC	$\lambda_1, \lambda_2 \in [-5, 5]$	1	$\lambda_1 = \lambda_2 = 1$

3.1.4 Implementation details and hyper-parameters setting

The proposed network architecture and the competitor non-linear methods are trained on the PyTorch platform, running on CentOS Linux 7 utilizing an NVIDIA Quadro P5000 Graphics Processing Unit (GPU) with 16 GB memory size. The experiments utilize the SGD solver [30] with a batch size of 64, and all activation functions are Rectified Linear Units (ReLU) [42]. All networks are trained with a learning rate of 10^{-3}, momentum 0.9, and weight decay 5×10^{-4}. The network weights are initialized utilizing the Xavier initialization [43] method, and to prevent overfitting, Dropout [44] is used to improve the network's generalization, with the random probability set to 0.5. By default, the instruction process steps are set to $r = 3$ and the approximation process step to $t = 2$. We run EMC-Nets for 10 runs and report the accuracy of the run with the lowest clustering loss. For the post-processing algorithms (DCCA, DCCAE, and DMSC), we run our method 20 times and report the result with the minimum loss. Since CCA-based methods, i.e., DCCA and DCCAE, can only deal with two views, the best two views are chosen on the CCV dataset according to their performance. The details on the proposed network architecture are given in the Appendix 1.

For a fair comparison, we uniformly use the grid search strategy to find the optimal hyper-parameters for all statistical methods, while for the deep learning methods, we

Table 3 Clustering results on BDGP, HW and CCV datasets

Categories	Dataset	BDGP			HW			CCV		
	Metrics	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
Single-view Clustering	SC$_{v=1}$	49.4	28.6	49.4	68.2	66.3	69.9	10.7	0.6	10.7
	SC$_{v=2}$	94.1	89.4	94.3	69.3	68.7	69.1	20.3	18.6	21.0
	SC$_{v=3}$	–	–	–	–	–	–	10.8	0.82	11.3
	SC$_{con}$	53.4	35.4	53.4	71.8	69.6	71.8	10.6	0.6	10.8
Statistic Multi-view Clustering	RMKMC	81.3	79.5	80.9	79.8	77.4	81.6	17.6	16.5	18.6
	RMSC	60.2	56.3	60.2	73.7	70.8	76.3	21.6	18.1	24.1
	MCGC	89.7	88.1	91.4	77.3	74.6	78.1	22.4	21.6	24.0
	AMGL	95.3	90.4	95.3	80.1	79.1	82.0	11.2	1.4	11.7
	COMIC	83.1	75.9	84.2	71.2	66.8	72.5	19.8	17.3	19.8
Deep Multi-view Clustering	DCCA	57.8	40.9	57.8	81.4	78.1	81.4	20.7	15.9	21.9
	DCCAE	45.3	39.1	44.1	68.2	59.7	68.2	16.1	11.8	18.3
	DMSC	68.1	50.6	73.8	91.6	85.5	91.6	17.5	13.5	25.1
	MCDMF	89.2	81.0	89.3	95.8	90.1	95.8	21.4	18.6	24.1
	DAMC	98.1	94.2	98.1	96.5	93.2	96.5	25.6	22.5	28.6
	EMC-Nets	**98.5**	**94.9**	**98.5**	**97.3**	**94.6**	**97.3**	**26.7**	**25.3**	**27.4**

Bold entries signify the best performance

exploit their default values. However, the key to the deep learning model is the weights initialization and the dimensionality of the joint representation. Hence, the network is initialized at five different random points, and the average value is considered the comparison baseline with the same dimensionality being preserved for a standard representation among all deep learning-based methods. The detail on the hyper-parameters selection is presented in Table 2.

3.2 Experimental results

The reported results on the four benchmark datasets are presented in Table 3. The findings are discussed from the aspect of comparing the methods against baseline techniques and clustering on a large-scale dataset.

3.2.1 Compared with baselines

The clustering performance of multi-view clustering methods (including statistical methods and deep methods) is significantly better than SC (based on only one or more view information), which indicates that it is necessary to fuse multi-view information for clustering. Compared with the statistical models, EMC-Nets has apparent advantages in the ACC, NMI, and Purity indicators. The principal reason is that these models utilize shallow and linear embedding functions and thus are unable to obtain complex information from the data. Even if compared with the deep methods, our framework can still achieve better results. Additionally, our framework is lighter than DAMC. Overall, we associate the success of EMC-Nets with the more precise guidance of the network during the training process.

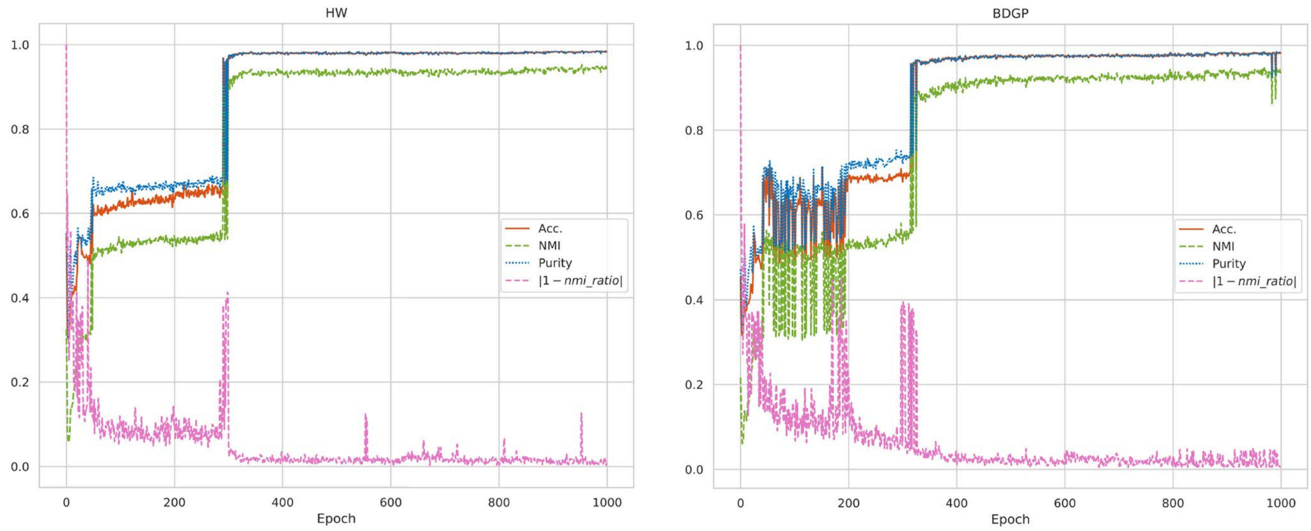
Table 4 Clustering result on MNIST dataset

Model	ACC	NMI	Purity
DCCA	47.6	44.3	49.2
DCCAE	43.8	39.3	45.1
BMVC	57.5	51.8	59.3
DMSC	65.3	59.4	65.7
DAMC	65.1	56.2	65.9
MCDMF	53.9	49.1	54.0
EMC-Nets	72.2	67.0	76.2

3.2.2 Clustering on Large-scale Dataset

To show that our approach is suitable for the large-scale dataset, we have conducted the experiments on the MNIST dataset. The challenged methods include four deep models, i.e., DCCA, DCCAE, DMSC, and DAMC. The results indicate that DMSC is not f scalable on this dataset due to its memory limitation, with the corresponding results presented in Table 4. The proposed approach manages a performance improvement of 6.9%(72.2 − 65.3),7.6%(67.0 − 59.4)and10.3%(76.2 − 65.9) against the second-best method in terms of ACC, NMI, and Purity metrics, respectively, demonstrating the effectiveness of the proposed approach on large-scale datasets.

According to the previous analysis, we found that EMC-Nets performs very well when processing pure visual tasks and can adapt to high-dimensional large-scale data. However, on some mixed views, such as CCV, the results are poor because the comprehensive information that EMC-Nets can capture is still limited. For example, the mixed features of audio, video, and subtitles are related in space and time



(a)

(b)

Fig. 4 The convergence of the proposed approach on the (**a**) HW and (**b**) BDGP datasets. The horizontal axis presents the number of training epochs. The red, green, and blue lines represent the cluster's ACC., NMI, and Purity metrics. The higher the value, the better. The pink line is the NMI ratio of the old and new pseudo labels, where the lower, the better

sequences. EMC-Nets is better at capturing space relationships rather than time sequences.

3.3 Further evaluation

In this subsection, we further discuss the proposed approach from the convergence perspectives, the impact of the hyper-parameters, component study, split process test, various fusion techniques, and visualization.

3.3.1 Convergence study

Our framework is trained alternately by (5) and (6), where (6) prevents the network from falling into the trivial solution, and (5) is to provide precise guidance for the network. Figure 4 shows the result of EMC-Nets in terms of ACC, NMI, and Purity on HW and BDGP. The solid red line shows the ACC value; the green dashed line the NMI of each method, and the blue dotted line the Purity. Before reaching 400 iterations, the network has a significant improvement, which means that the model undergoes a mutually compatible process during the alternate training. This process is pronounced on the BDGP dataset. For the convergence, as the pink dash line shows in Fig. 4, during the instantaneous improvement, the NMI ratio is also volatile. At about the end of the instantaneous improvement, i.e., during the stable phase of the alternate process, the pseudo label will undergo a large-scale change, and the NMI ratio is volatile. After that phase, the NMI ratio begins to stabilize and gradually converges. After 400 iterations, the network performance improves and enters a convergence phase. The results show that our method needs to go through a long period of oscillation to reach convergence.

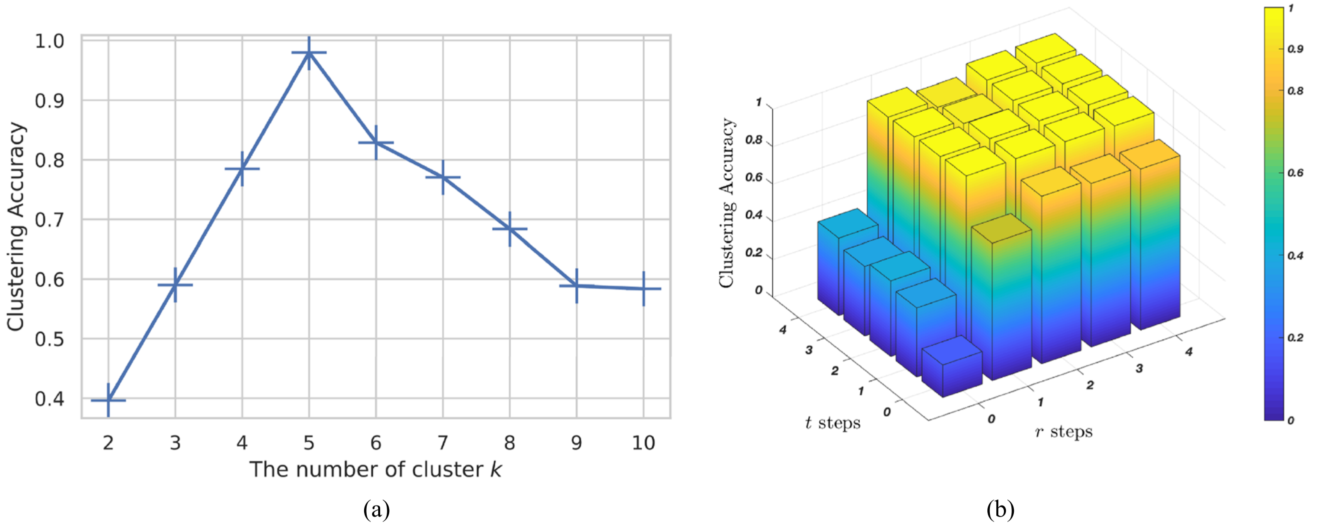
3.3.2 The impact of hyper-parameters

In the proposed framework, three hyper-parameters require tuning, i.e., the number of clusters k, the steps of the instruction process r, and the approximation process t. We discuss the influence of different values of these hyper-parameters on the BDGP dataset utilizing the metrics of Section 4.1.2 presented in Fig. 5. For the different clustering values k, Fig. 5(a) highlights that the proposed method reaches the highest value $k = 5$, where the elbow phenomenon [45] is evident.

Additionally, we evaluate different r and t values to investigate the network's performance. For each value pair, trials are run five times, and the mean ACC is reported. According to Fig. 5(b), when $r = 0$ or $t = 0$, the clustering accuracy is unstable, but as the number of instruction processes increases, the clustering accuracy first increases and then reduces. Additionally, when $t \leq r$, the performance is better than $t > r$. The optimum performance is reached for $r = 3$ and $t = 2$. This experiment also reveals that the instruction process provides to the network the correct guidance to prevent it from a trivial solution and that the approximation process enhances the network's capability to approximate the locally optimal solution efficiently.

3.3.3 Component study

We train three variants of our proposed method to examine the effect of the attention fusion layer, instruction, and



(a) (b)

Fig. 5 The influence of hyper-parameters of EMC-Nets on BDGP dataset, where (**a**) is the influence of different values of k on ACC, and (**b**) refer to the matrix of the changes of ACC when instruction process step r and approximating process step t set as $\{0, 1, 2, 3, 4\}$, respectively

Table 5 Component study on MNIST dataset

Model	ACC	NMI	Purity
EMC-Nets$_{con}$	62.3	61.3	69.5
EMC-Nets$_{appro}$	39.8	33.7	51.9
EMC-Nets$_{ins}$	63.1	61.5	69.5
EMC-Nets	**72.2**	**67.0**	**76.2**

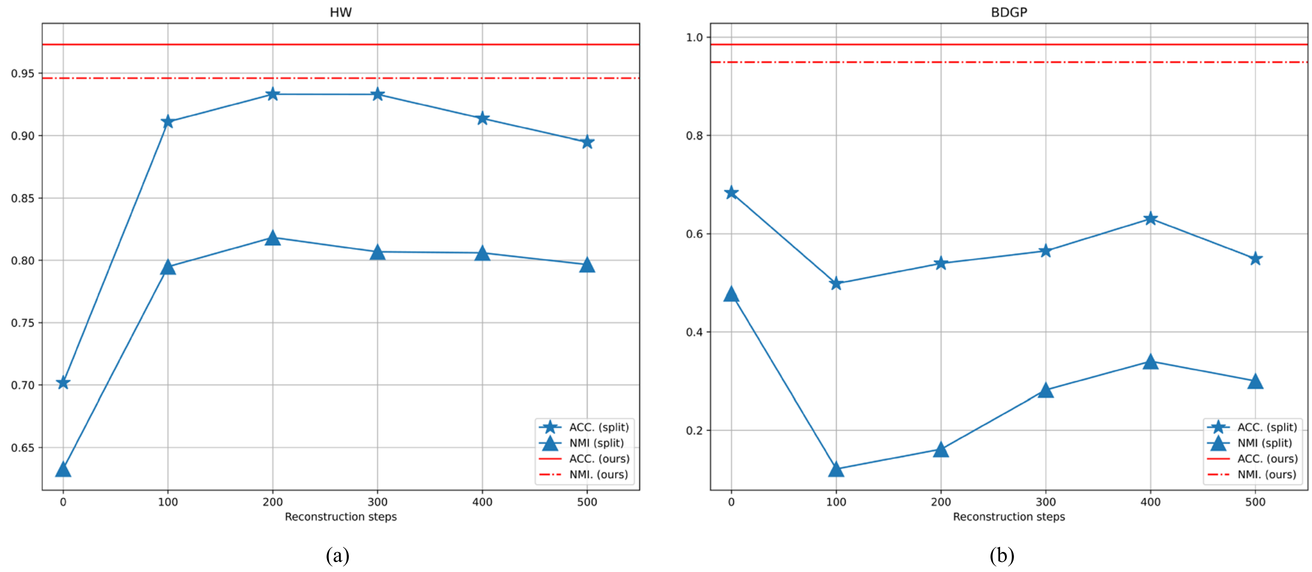
Bold entries signify the best performance

approximation process: (1) EMC$_{con}$ represents the EMC-Nets variant by removing the attention fusion layer and simply using concatenated features, (2) EMC$_{appro}$ is obtained by removing the supervisor module, i.e., this variant has only an attention fusion layer and an approximating process, (3) EMC$_{ins}$ is obtained by removing the approximating process in EMC-Nets. Table 5 shows the experimental results on the MNIST dataset, from which some critical observations can be made.

First, from the comparison of EMC$_{con}$ and EMC-Nets, the attention mechanism is better than the concatenated feature fusion. Second, from the results of EMC$_{appro}$, the performance without an instruction process is not ideal, as the pseudo labels generation can easily fall into a trivial solution. Correspondingly, the comparison of the EMC$_{ins}$ and EMC$_{appro}$ shows that the dual nature of this process conforms to our hypothesis, i.e., reconstruction is effective but not efficient, and pseudo labels can provide efficient guidance to the network. These results demonstrate that the proposed instruction process, approximation process, and attention fusion layer play a key role in improving the performance of multi-view clustering.

3.3.4 Split process test

To further study the efficiency of the proposed alternate training process, we comprehensively study the independent training scheme of the two processes. Concretely, the instruction process is first executed and then the approximation process to ensure that the two separations have no intersection. To ensure that the separation process and the alternate process can be trained 1000 times, during the separation process, we set the instruction process to execute 0, 100, 200, 300, 400, 500 times, respectively, and the corresponding approximation process to execute 1000, 900, 800, 700, 600, 500 times. Especially when the number of executions for the instruction process is 0, it is equivalent to the behavior of [19]. The corresponding experimental results are presented in Fig. 6. For the separation process, the number of instruction process training helps to improve the clustering performance, but as the number increases, the performance decreases. This phenomenon is consistent with our previous assumptions, i.e., reconstruction is effective but not efficient. Without executing the instruction process, the clustering performance is volatile, which is also consistent with our previous assumptions and is the weakness of [19]. In general, the performance of the split process is worse than the alternate process because the separation process does not entirely solve the problem or provides a trivial solution. However, the alternate process can solve these problems.



(a)

(b)

Fig. 6 The impact of split test, the red line refers the performance of our method, the blue line represents the performance of different degrees of split

Table 6 Different fusion techniques on MNIST dataset

Model	ACC	NMI	Purity
Concat.	62.3	61.3	69.5
Affinity	68.4	61.7	70.9
DNN	69.1	65.8	69.1
Attention	**72.2**	**67.0**	**76.2**

Bold entries signify the best performance

3.3.5 The impact of different fusion techniques

In this part, we study the impact of different view fusion methods on the EMC-Nets and examine four fusion methods:

1. 1. Concatenated, i.e., simply concatenating features extracted from each view, $Z_{con} = [H_1, H_2, \cdots, H_3]$.
2. Affinity Fusion [12], by establishing a self-expression layer to fuse different views as:

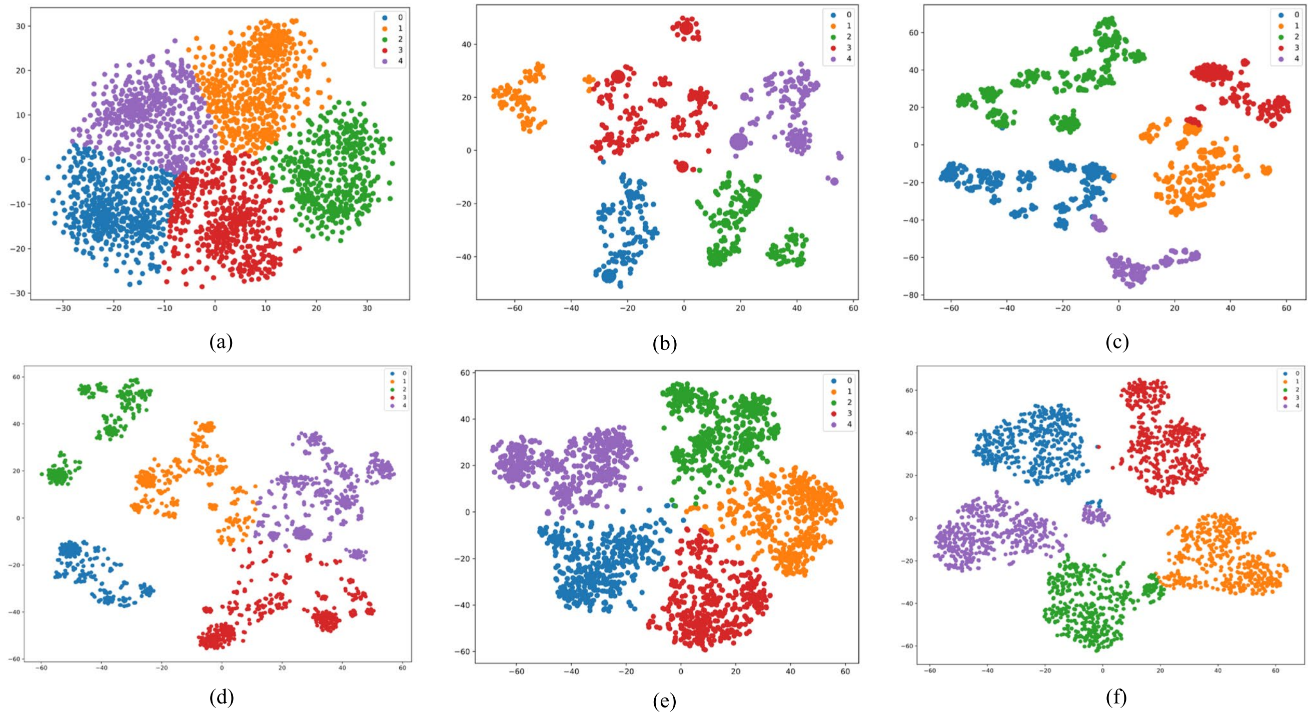
$$\min_{\Theta_s} \sum_{i=1}^{V} \|H_i - H_i\Theta_s\|$$

 where Θ_s is the self-expression parameters we need, and spectral clustering can be used directly to generate pseudo labels.
3. DNN fusion. Based on category one above, we build a fusion network $f(\cdot)$, where f comprises two fully connected layers.
4. Attention fusion, which used in EMC-Nets.

We conduct a fusion comparison experiment on MNIST, and for fairness, the feature dimensions output of all fusion methods is uniformly set to 200. The corresponding results are presented in Table 6, highlighting that in addition to the concatenation fusion strategy, the other three fusion methods are not far apart, while the attention fusion method manages the optimum performance. We believe that affinity fusion only considers the consistency between views and ignores complementarity. DNN fusion is better than affinity fusion, while due to the characteristics of the fully connected network, DNN fusion has the opportunity to consider more information, but it cannot distinguish it. In the process of attention fusion, it is necessary to calculate the similarity of each view, which ensures consistency and allocates the ratio of the fusion information. This fusion strategy considers the complementary information, and since the information is allocated according to its similarity, the model can distinguish the information importance Table 7.

3.3.6 Visualization

To demonstrate further the clustering effect of EMC-Nets, we use the t-SNE [46] visualization scheme to show the common representation extracted from each deep model on the BDGP dataset, i.e., DCCA, DMSC, and DAMC. As illustrated in Fig. 7, the proposed approach gives a more clear and compact cluster structure.



(a) (b) (c)

(d) (e) (f)

Fig. 7 Visualization of original features for each view and the common latent representations given by different algorithms with t-SNE [46] on BDGP dataset, where (**a**) original data of first view, (**b**) original data of second view, (**c**) DCCA, (**d**) DMSC, (**e**) DAMC, and (**f**) EMC-Nets

4 Discussion

The proposed approach poses several advantages, including its lower computational burden requirements than the existing state-of-the-art frameworks [16, 17]. Additionally, the experiments prove that the clustering quality can be improved by utilizing pseudo labels to train the feature extractor. Furthermore, the reconstruction procedure prevents the network from providing a trivial non-optimum solution. We have also proved that the proposed alternating training process can converge well after a sufficient number of down generations, and finally, the clustering module in EMC-Nets can be replaced with other clustering algorithms, such as spectral clustering, which makes our framework highly adaptable. Despite its advantages, our method has a few limitations. Specifically, it has an excessive period of shaking, as, during the experimental evaluation, our method presents a long oscillation period (about 400 iterations). This period is unfavorable for tight computing power, and this phenomenon is caused by the uncertainty of unsupervised learning and the compatibility of the alternate processes. Additionally, our method is not suitable for small sampled data. Our framework employs the attention mechanism as the core of the fusion layer, directly affecting the quality of the joint representation. The attention mechanism can exert a powerful performance for sufficient data volume, but its performance is weak compared to other forms of fusion mechanisms for small samples.

5 Conclusion

This paper proposes a novel framework, entitled Efficient Multi-view Clustering Networks (EMC-Nets), which includes a feature extractor, supervisor module, and a clustering module. Utilizing pseudo labels to enhance the feature extractor affords EMC-Nets learning discriminative common representation efficiently. Furthermore, we design robust alternating processes to make EMC-Nets more efficient than other reconstruction-based algorithms. We treat the instruction process to prevent the model from providing a trivial non-optimum solution during training and the approximation process to fit the reasonable cluster distribution. Experimental results on four real-world datasets demonstrate the superiority of our model over several state-of-the-art multi-view clustering methods. Our future work shall focus on a more stable training process. This is quite challenging as it requires fully understanding the training dynamics of the alternate process and studying the network's learning behavior. Additionally, we also consider the theoretical relationship of the core concepts exploited. Intuitively, we consider that the instruction process and pseudo labels training process are closer to the E-step and M-step ideas in the Expectation-Maximization (EM) algorithm [47]. The future aim shall be to prove the relationship between them.

Table 7 EMC-Nets in the BDGP experiments

Module	Layer	Input	Output	Hyper-parameters
View-1 Encoder	Linear	1750	500	bias=False
	Linear	500	300	
	Linear	300	100	
View-2 Encoder	Linear	79	300	bias=False
	Linear	300	100	
Fusion Layer	TransformerEncoder ×1	200	200	head=1, dff=256
Classifier	Linear+Sigmoid	200	5	
View-1 Decoder	Linear	200	300	
	Linear	300	500	
	Linear+Sigmoid	500	1750	
View-2 Decoder	Linear	200	300	
	Linear+Sigmoid	300	79	
The total number of parameters: 2, 750, 446				

Table 8 EMC-Nets in the HW experiments

Module	Layer	Input	Output	Hyper-parameters
View-1 Encoder	Linear	240	150	bias=False
View-2 Encoder	Linear	76	50	bias=False
Fusion Layer	TransformerEncoder ×1	200	200	head=1, dff=1024
Classifier	Linear+Sigmoid	200	10	
View-1 Decoder	Linear	200	200	
	Linear+Sigmoid	200	240	
View-2 Decoder	Linear+Sigmoid	200	76	
The total number of parameters: 1, 290, 424				

Table 9 EMC-Nets in the CCV experiments

Module	Layer	Input	Output	Hyper-parameters
View-1 Encoder	Linear	4000	2000	bias=False
	Linear	2000	1000	
	Linear	1000	200	
View-(2,3) Encoder	Linear	5000	3000	bias=False
	Linear	3000	2000	
	Linear	2000	800	
	Linear	800	400	
Fusion Layer	TransformerEncoder ×2	1000	1000	head=1, dff=2048
Classifier	Linear+Sigmoid	1000	20	
View-1 Decoder	Linear	1000	2000	
	Linear	2000	4000	
View-(2,3) Decoder	Linear	1000	2000	
	Linear	2000	3000	
	Linear	3000	5000	
The total number of parameters: 126, 410, 764				

Table 10 EMC-Nets in the MNIST experiments

Module	Layer	Input	Output	Hyper-parameters
View-(1,2) Encoder	Linear	784	400	bias=False
	Linear	400	500	
	Linear	500	100	
Fusion Layer	TransformerEncoder ×2	200	200	head=1, dff=1024
Classifier	Linear+Sigmoid	200	10	
View-(1,2) Decoder	Linear	200	400	
	Linear	400	500	
	Linear+Sigmoid	500	784	
The total number of parameters: 4, 114, 650				

Appendix: Network architectures

In this section, we report the details of the network architecture, includes network layer, input and output size, hyper-parameters of the layer, and the total number of parameters, used in the experiments. Note that all the layer of network are used Pytorch-style API. For the fully-connected (linear) layer, we use *bias* as default, and we report the *head* and the dimension of feedforward (dff) of the TransformerEncoder (Tables 7, 8, 9, 10).

Acknowledgements This work was supported by Natural Science Foundation of Guangdong Province, China (No. 2020A1515011468; 2016A0303010003); Guangdong University Scientific Research Project, China (No. 2019KTSCX189; 2018KTSCX235; 2018WZDXM014); Joint Research and Development Fund of Wuyi University and Hong Kong and Macau (2019WGALH21).

References

1. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2):91–110
2. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. CVPR 1:886–893
3. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE TPAMI 24(7):971–987
4. Yang Y, Wang H (2018) Multi-view clustering: A survey. Big Data Mining and Analytics 1(2):83–107
5. Liu J et al (2013) Multi-view clustering via joint nonnegative matrix factorization, In: Proceedings of the 2013 SIAM international conference on data mining, society for industrial and applied mathematics
6. Zhang Z et al (2018) Binary multi-view clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 41(7):1774–1782
7. Peng X, Huang Z, Lv J, Zhu H, Zhou JT (2019) COMIC: Multi-view clustering without parameter selection. International Conference on Machine Learning 97(ICML'19):5092–5101
8. Chaudhuri K et al (2009) Multi-view clustering via canonical correlation analysis. In: Proceedings of the 26th annual international conference on machine learning, ACM, pp 129–136
9. Melzer T, Reiter M, Bischof H (2001) Nonlinear feature extraction using generalized canonical correlation analysis. In: Proc. intern. conf. artificial neural networks (ICANN2001), pp 353–360
10. Galen A, Raman A, Jeff B, Livescu K (2013) Deep canonical correlation analysis. In: International conference on machine learning, pp 1247–1255
11. Zhao H, Zhengming D, Yun F (2017) Multi-view clustering via deep matrix factorization. In: Proceedings of the AAAI conference on artificial intelligence, vol 31, No 1
12. Abavisani M, Patel VM (2018) Deep multimodal subspace clustering networks. IEEE Journal ofSelected Topics in Signal Processing 12(6):1601–1614
13. Zhang C, Liu Y, Fu H (2019) AE^2-Nets: Autoencoder in autoencoder networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2577–2585
14. Zhang Y et al (2019) A multitask multiview clustering algorithm in heterogeneous situations based on LLE and LE. Knowledge-Based Systems 163:776–786
15. Zhaoyang L, Wang Q, Tao Z, Gao Q, Yang Z (2019) Deep Adversarial multi-view clustering network. In: IJCAI, pp 2952–2958
16. Zhou R, Shen Y-D(2020) End-to-End adversarial-attention network for multi-modal clustering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14619–14628
17. Goodfellow IJ et al (2014) Generative adversarial nets. In: Proceedings of the 27th international conference on neural information processing systems - volume 2 (NIPS'14). MIT Press, Cambridge, MA, USA, pp 2672–2680
18. Zhan X, Xie J, Liu Z, Ong Y-S, Loy CC (2020) Online deep clustering for unsupervised representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6688–6697
19. Caron M, Piotr B, Joulin A, Douze M (2018) Deep clustering for unsupervised learning of visual features. In: Proceedings of the european conference on computer vision (ECCV), pp 132–149
20. Vaswani A et al (2017) Attention is all you need. In: NIPS
21. Jing L, Tian Y (2020) Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence
22. Tu W et al (2021) Deep fusion clustering network. In: AAAI
23. Xu LL, Neufeld J, Larson B, Schuurmans D (2005) Maximum margin clustering. In: Proc. 18th annu. conf. advances in neural information processing systems 17, Cambridge, MA, USA, pp 1537–1544
24. Du L, Zhou P, Shi L, Wang HM, Fan MY, Wang WJ, Shen YD (2015) Robust multiple kernel k-means using '2;1-norm.

In: Proc. 24th int. conf. artificial intelligence, Buenos Aires, Argentina, pp 3476–3482

25. Chaudhuri K, Kakade SM, Livescu K, Sridharan K (2009) Multi-view clustering via canonical correlation analysis. In: Proc. 26th annu. int. conf. machine learning, Montreal, Canada, pp 129–136
26. Zhao XR, Evans N, Dugelay JL (2014) A subspace co-training framework for multi-view clustering. Pattern Recogn. Lett. 41:73–82
27. Guo YH (2013) Convex subspace representation learning from multi-view data. In: Proc. 27th AAAI conf. artificial intelligence, Bellevue, WA, USA, pp 387–393
28. Xue Z, Li GR, Wang SH, Zhang CJ, Zhang WG, Huang QM (2015) (2015) GOMES: A group-aware multi-view fusion approach towards real-world image clustering. In: Proc. IEEE int. conf. multimedia and expo, Turin, Italy, pp 1–6
29. Wang W et al (2015) On deep multi-view representation learning. In: International conference on machine learning, PMLR, pp 1083–1092
30. Bottou L (2012) Stochastic gradient descent tricks. In: Neural networks: Tricks of the trade, Springer, pp 421–436
31. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
32. Asuncion A, Newman D (2007) Uci machine learning repository
33. Shang C, Palmer A, Sun J, Chen K-S, Lu J, Bi J (2017) VIGAN: missing view imputation with generative adversarial networks. arxiv:1708.06724
34. Cai X, Wang H, Huang H, Ding C (2012) Joint stage recognition and anatomical annotation of drosophila gene expression patterns. Bioinformatics 28(12):i16–i24
35. Jiang Y-G, Ye G, Chang S-F, Ellis D, Loui AC (2011) Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In: ICMR, pp 29
36. Kumar A, Rai P, Daume H (2011) Co-regularized multi-view spectral clustering. In: NIPS, pp 1413–1421
37. Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: Analysis and an algorithm. In: Proceedings of the 14th international conference on neural information processing systems: natural and synthetic
38. Cai X, Nie F, Huang H (2013) Multi-view k-means clustering on big data. In: Twenty-Third international Joint conference on artificial intelligence
39. Xia R, Pan Y, Du L, Yin J (2014) Robust multi-view spectral clustering via low-rank and sparse decomposition. In: AAAI, pp 2149–2155
40. Zhan K, Nie F, Wang J, Yang Y (2019) Multiview consensus graph clustering. In: A publication of the IEEE signal processing society, IEEE transactions on image processing
41. Nie F, Li J, Li X et al (2016) Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In: IJCAI, pp 1881–1887
42. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, pp 315–323
43. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR workshop and conference proceedings, pp 249–256
44. Hinton GE, Nitish S, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580
45. Ketchen DJ, Shook CL (1996) The application of cluster analysis in strategic management research: an analysis and critique. Strategic Management Journal 17(6):441–458
46. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. Journal of Machine Learning Research 9:2579–2605
47. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 39(1):1–22
48. Kuhn HW (1955) The Hungarian method for the assignment problem. Naval research logistics quarterly 2(1–2):83–97

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

Guanzhou Ke received his B.S degree in communication engineering from Wuyi University, Jiangmen, China, in 2019. He is currently pursuing a Master's degree in system engineering from Wuyi University, Jiangmen, China. His research interests include deep learning, computer vision, multi-view learning, and unsupervised learning.

Zhiyong Hong received the B.S. and M.S. degrees in computer science and technology from the Shenyang Institute of Technology, Shenyang, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science and technology from Southwest Jiaotong University, Chengdu, China, in 2014. From 2006 to 2014, he was a Lecturer with the School of Computer, Wuyi University, Jiangmen, China. Since 2021, he has been a Professor. His research interests include intelligent information processing, block chain, and rough set theory.