Contrastive Multi-view Subspace Clustering via Tensor Transformers Autoencoder

Qianqian Wang[1*], Zihao Zhang[1], Wei Feng[2], Zhiqiang Tao[3], Quanxue Gao[1]

[1]School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi, China, 710071
[2]School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, China, 710049
[3]School of Information, Rochester Institute of Technology, Rochester, NY, USA, 14623
qqwang@xidian.edu.cn, zihaoz2021@stu.xidian.edu.cn, zhiqiang.tao@rit.edu, weifeng.ft@xjtu.edu.cn, qxgao@xidian.edu.cn

Abstract

Multi-view clustering aims to identify consistent and complementary information across multiple views to partition data into clusters, emerging as a popular unsupervised method for multi-view data analysis. However, existing methods often design view-specific encoders to extract distinct features from each view, lacking exploration of their complementarity. Additionally, current contrastive-based multi-view clustering methods may lead to erroneous negative sample pairs conflicting with the clustering objective. To address these challenges, we propose a novel Contrastive Multi-view Subspace Clustering via Tensor Transformers Autoencoder (TTAE). On the one hand, it facilitates information exchange between views by tensor transformers autoencoder, thereby enhancing complementarity. On the other hand, It learns a consistent subspace with a self-expression layer. Meanwhile, adaptive contrastive learning helps to provide more discriminative features for the self-expression learning layer, and the self-expression learning layer in turn supervises contrastive learning. Moreover, our method adaptively selects positive and negative samples for contrastive learning to mitigate the impact of inappropriate negative sample pairs. Extensive experiments on several multi-view datasets demonstrate the effectiveness and superiority of our model.

Introduction

Owing to the rapid development of information technology, multi-view data, which comprise multiple modalities, have become increasingly ubiquitous, such as in applications like instant music videos. The vastness and complexity of multi-view data make it challenging to obtain reliable data labels, resulting in significant obstacles for data processing. As a consequence, Multi-view Clustering (MVC) has gained considerable attention (Zhou et al. 2024; Huang et al. 2021; Cao et al. 2015). MVC is an unsupervised method that effectively leverages both inter-view and intra-view statistical properties to partition data into meaningful clusters. MVC has found wide application across a variety of domains. For example, in social network analysis, combining views such as user profiles, social relationships, and behavioral records enables a deeper understanding of user behavior (Fang et al. 2023). In image processing, integrating multiple views—such as color histograms and texture features—facilitates more nuanced content analysis, leading to improved clustering performance (Wang et al. 2022c). In data mining, merging features from diverse views can uncover hidden associations and trends, facilitating the extraction of valuable insights while reducing computational complexity. Prior research on MVC can be broadly categorized based on architecture: traditional MVC methods (Tao et al. 2017; Zhang et al. 2018, 2023a) and deep MVC methods (Lin et al. 2021; Zhao et al. 2024).

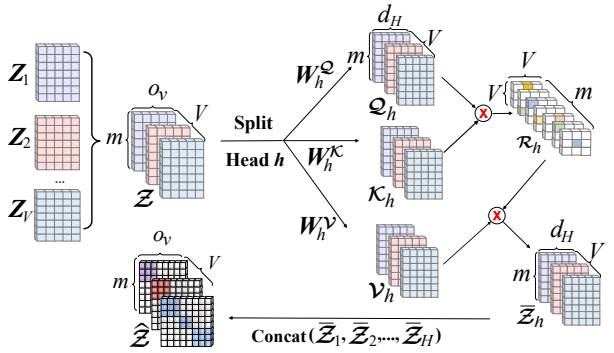

Figure 1: The flowchart of the Tensor Attention Layer. Due to the parallel computation of multiple attention heads, we depict only a single head h. These heads are concatenated after the same transformation i.e., to obtain the strengthened representation $\hat{\mathcal{Z}}$.

Traditional MVC methods include Non-negative Matrix Factorization (NMF) clustering (Li et al. 2023; Zong et al. 2017), graph constraint-based methods (Orecchia and Zhu 2014; Zhang et al. 2019), and subspace clustering methods (Cao et al. 2015; Li et al. 2019). However, these traditional methods primarily rely on shallow and linear embedding functions to capture the clustering structure, which limits their ability to explore the deep, nonlinear relationships within multi-view data. Although some approaches (Akaho 2001; Tzortzis and Likas 2012) attempt to address this limitation by incorporating multi-kernel learning strategies to capture nonlinear features, their clustering performance remains suboptimal.

In recent years, the emergence of deep neural networks

**Corresponding author

(DNNs) has led to remarkable advancements in representation learning (Bengio, Courville, and Vincent 2013). Many deep MVC methods (Ji et al. 2017; Wang et al. 2022b; Xie et al. 2020) have demonstrated impressive clustering performance, aided by feature extraction networks. Among these methods, deep multi-view subspace clustering (MVSC) has gained significant attention. This approach assumes that different modalities reside in distinct subspaces and aims to identify a shared subspace across multi-view data by maximizing the mutual agreement between different modalities. For example, Abavisani *et al.* (2018) explored various fusion strategies and proposed an unsupervised deep multimodal subspace clustering method (DMSC) based on convolutional neural networks.

Although these models have made significant strides in deep MVC, several challenges remain: 1) Many existing deep MVSC algorithms (e.g., (Abavisani and Patel 2018; Wang et al. 2020)) typically assign each view a view-specific encoder and impose constraints to maximize inter-view consistency. However, the features extracted by this approach often lack inter-view interaction, causing them to miss complementary information across views. 2) To explore the discriminative structure of multi-view data, some self-supervised methods (Wang et al. 2021, 2024) have been proposed, using clustering labels as supervised information to guide subspace learning. However, the inaccuracy of these cluster labels severely limits improvements in clustering performance. 3) Some current deep MVC algorithms incorporate contrastive learning to acquire discriminative features. Yet, these methods (e.g., (Xu et al. 2022)) typically treat pairs of corresponding views as positive pairs and non-corresponding samples as negative pairs. This strategy can lead to *noisy contrastive learning*, potentially pushing samples from the same cluster farther apart, which contradicts the goal of clustering.

To address these issues, we propose a novel method called Contrastive Multi-view Subspace Clustering via Tensor Transformers Auto-Encoder (*i.e.*, TTAE). TTAE effectively integrates Tensor Transformers Autoencoder, a Self-Expression Layer, and an Adaptive Contrastive Learning Module, forming a unified framework that facilitates both consistent and complementary learning. Specifically, TTAE incorporates the Tensor Attention Layer (illustrated in Figure 1), which captures feature information across views and fosters complementary learning. Additionally, TTAE employs contrastive learning to extract discriminative features. To further enhance the process, the Self-Expression Layer provides pairwise affinities, which help capture data point relationships. This layer also serves as a self-supervised method, mitigating the noise issues typically encountered in contrastive learning. The main contributions of TTAE are summarized below:

- We propose a novel multi-view subspace clustering method based on the Tensor Transformers Autoencoder, which enables effective feature extraction while facilitating cross-view information interaction for complementary learning.
- Our TTAE Module introduces a novel approach for constructing positive and negative pairs. By leveraging the self-expression coefficient matrix, it adaptively generates positive and negative samples, enabling the exploration of both consistent and discriminative structures in unlabeled multi-view data.
- We conduct extensive experiments on several benchmark datasets, demonstrating that TTAE significantly improves clustering performance compared to several baseline methods.

Related Work

Deep Multi-view Clustering

Since multiple views describe the same object, they inherently contain shared knowledge. The challenge in Multi-view Clustering (MVC) lies in extracting this shared information through consistent learning and leveraging it for clustering tasks. A common approach to achieving consistency is by maximizing the correlation between views to extract shared components. Canonical Correlation Analysis (CCA), and its deep variant, Deep CCA (Andrew et al. 2013), are representative methods that non-linearly transform data to obtain highly correlated representations. Building on this consistency principle, deep Multi-view Subspace Clustering (MVSC) methods aim to identify a common subspace across all views, enabling coherent clustering. These methods have demonstrated strong performance in various applications (Wang et al. 2020, 2023). In addition to consistency, different views also contain complementary information unique to each modality. While consistency focuses on shared knowledge, complementarity emphasizes the distinct perspectives each view provides. Recent deep MVSC methods (Zhang et al. 2023b; Wang et al. 2022a; Zhang et al. 2024a) aim to leverage both aspects—capturing both shared and complementary information. Unlike these methods, TTAE focuses on the interaction of information across multiple views and the discovery of discriminative features. It achieves this through the integration of a Tensor Transformers Autoencoder and an Adaptive Contrastive Learning Module, which together facilitate both consistent and complementary learning for improved clustering performance.

Contrastive Learning

Contrastive learning successfully enhances clustering performance as an effective technique for discriminative feature learning, leading to the development of specialized methods tailored to different data structures, such as images (Li et al. 2021) and graphs (Zhong et al. 2021). For example, Graph Contrastive Clustering (GCC) (Zhong et al. 2021) utilizes KNN-based graph contrastive learning to shift from instance-level consistency to cluster-level consistency, further improving clustering performance. However, due to the unsupervised nature of clustering, implementing label-guided contrastive learning (Khosla et al. 2020) presents a challenge, as it increases the risk of treating similar samples as negative pairs. To address this issue, various clustering methods have incorporated self-supervised signals during training. These approaches address the problem from multiple angles, including: 1) guiding the learning process

through pseudo-labeling (Jing et al. 2021; Feng et al. 2024), 2) controlling the quantity of sample pairs to mitigate errors (Yang et al. 2023), 3) adjusting the weights of samples based on confidence (Yan et al. 2023; Liu et al. 2024; Lu et al. 2024), and 4) refining the boundary conditions for constructing sample pairs (Tang and Liu 2022). A notable aspect of these methods is the emphasis on similarity metrics between samples. Unlike previous approaches, TTAE uniquely employs the self-expression coefficient matrix to effectively mitigate false-negative samples, achieving more accurate clustering results.

The Proposed Method

TTAE is composed of three parts: Tensor Transformers Autoencoder, Self-expression Layer, and Adaptive Contrastive Learning Module as depicted in Figure 2. In addition, Table 1 records the important notation of the paper.

Notion	Definition
m	Number of samples
V	Number of views
k	Number of Nearest samples taken
f	Number of cluster
d_v	Feature dimensions of original v-th data
o_v	Feature dimensions of latent v-th data
$\mathbf{X}_v \in \mathbb{R}^{m\times d_v}$	Input data matrix
$\mathbf{Z}_v \in \mathbb{R}^{m\times o_v}$	Latent features matrix of v-th data
$\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{m\times V\times o_v}$	Latent features tensor
$\hat{\boldsymbol{\mathcal{Z}}} \in \mathbb{R}^{m\times V\times o_v}$	Strengthened features tensor
$\bar{\mathbf{Z}}_v \in \mathbb{R}^{m\times o_v}$	Strengthened features matrix of v-th data
$\mathbf{S}$	Consistent representation matrix
$\mathbf{C}$	Affinity matrix $\mathbf{C} = \frac{1}{2}(\lvert\mathbf{S}\rvert + \lvert\mathbf{S}\rvert^T)$

Table 1: Main notations in this paper.

Tensor Transformers Autoencoder

Without loss of generality, we consider two views as examples. Given two heterogeneous views $\mathbf{X}_1 = \{x_1^1, x_1^2, x_1^3, ..., x_1^m\}^T \in \mathbb{R}^{m\times d_1}$, $\mathbf{X}_2 = \{x_2^1, x_2^2, x_2^3, ..., x_2^m\}^T \in \mathbb{R}^{m\times d_2}$, where m is the number of samples, d_v denotes the corresponding dimensions of the v-th views. To handle the high-dimensional and complex nature of raw data, we begin by transforming individual view data into a more compact representation through a feature extraction layer. This step reduces the dimensionality of the input data $\mathbf{X}_v$ and uncovers latent low-dimensional features $\mathbf{Z}_v$. Specifically, the feature extraction process follows the mapping: $\mathbf{Z}_v = \mathbf{E}_v(\mathbf{X}_v; \theta_{e_v}) \in \mathbb{R}^{m\times o_v}$, where θ_{e_v} denotes the parameters of the encoder $\mathbf{E}_v$, o_v represents the output dimension.

Through the feature extraction layer, it adeptly preserves unique information from each perspective, laying the groundwork for subsequent consistency learning—a measure often employed by many prominent clustering methods (Xu et al. 2022; Zhang et al. 2024b). However, as each view may provide distinct clustering value information, the challenge (Wang et al. 2024; Luo et al. 2018; Liu et al. 2023) lies in capturing complementary information across views to enhance clustering performance. Motivated by this, we explore inter-view interactions to capture complementary information within the prevailing Transformer framework (Vaswani et al. 2017).

Given $\mathbf{X}_v$ and its corresponding representation $\mathbf{Z}_v$ for view v, we expand the features $\mathbf{Z}_v$ from all views into a three-dimensional tensor $\boldsymbol{\mathcal{Z}} \in \mathcal{R}^{m\times V\times o_v}$, which is then passed into the subsequent attention module. Assuming the presence of attention heads (H) and their corresponding linear transformation weights for query ($\mathbf{W}^{\mathcal{Q}}$), key ($\mathbf{W}^{\mathcal{K}}$), and value ($\mathbf{W}^{\mathcal{V}}$), we take the input tensor $\boldsymbol{\mathcal{Z}}$, map it to feature representations, and perform the associated dimensional transformations. The resulting representations $\{\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{K}}, \boldsymbol{\mathcal{V}}\}$ belong to $\mathbb{R}^{m\times H\times V\times d_H}$, where the dimension d_H is determined as $d_H = o_v/H$. Computing multiple attention heads in parallel enables the division of feature dimensions into o_v/H, which enhances the effectiveness of feature representations. For each attention head h, the following operation is performed:

$$\boldsymbol{\mathcal{R}}_h = softmax(\ \Upsilon_{\{:,:,i,j\}}(\boldsymbol{\mathcal{Q}}_h, \boldsymbol{\mathcal{K}}_h^T)/\sqrt{d_h}\ ), \quad (1)$$

where the operator $\Upsilon_{:,:,i,j}(\cdot,\cdot)$ denotes matrix multiplication along the last two dimensions. By leveraging the parallel computation of multiple attention heads, the attention scores $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{m\times H\times V\times V}$ are obtained, capturing the similarity relationships among views.

Following this, the output features of the attention module are computed as $\bar{\boldsymbol{\mathcal{Z}}} = \Upsilon_{(:,:,i,j)}(\boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{V}}) \in \mathbb{R}^{m\times H\times V\times d_H}$, which represents the aggregated output from multiple attention heads. This representation undergoes further linear transformations and is refined through a fully connected layer to produce a strengthened representation:

$$\hat{\boldsymbol{\mathcal{Z}}} = \text{Concat}(\bar{\boldsymbol{\mathcal{Z}}}_1, \bar{\boldsymbol{\mathcal{Z}}}_2, ..., \bar{\boldsymbol{\mathcal{Z}}}_H) \in \mathbb{R}^{m\times V\times o_v}. \quad (2)$$

From this strengthened representation, the enhanced feature of each view $\bar{\mathbf{Z}}_v \in \mathbb{R}^{m\times o_v}$ can be extracted by decomposing $\hat{\boldsymbol{\mathcal{Z}}}$. This process is critical for facilitating effective multi-view interactions.

In addition, the decoder is responsible for performing reconstruction tasks. The modality-specific information, after undergoing view interactions, is passed to the corresponding modality decoders to reconstruct precise modality representations, thereby guiding the learning dynamics of Tensor Transformers Autoencoders. Specifically, the objective is to maximize the coherence between the original and reconstructed data while ensuring the effectiveness of the latent low-dimensional features within the hidden layers. This is typically achieved by minimizing the following loss function:

$$\mathcal{L}_{re} = \sum_{v=1}^{V} ||\mathbf{X}_v - \hat{\mathbf{X}}_v||_F^2, \quad (3)$$

where $\hat{\mathbf{X}}_v$ is the reconstructed view data and $\|\cdot\|_F$ stands for the Frobenius paradigm.

Self-expression Layer

The self-expression layer is employed to acquire a self-expression coefficient matrix exhibiting a distinct cluster

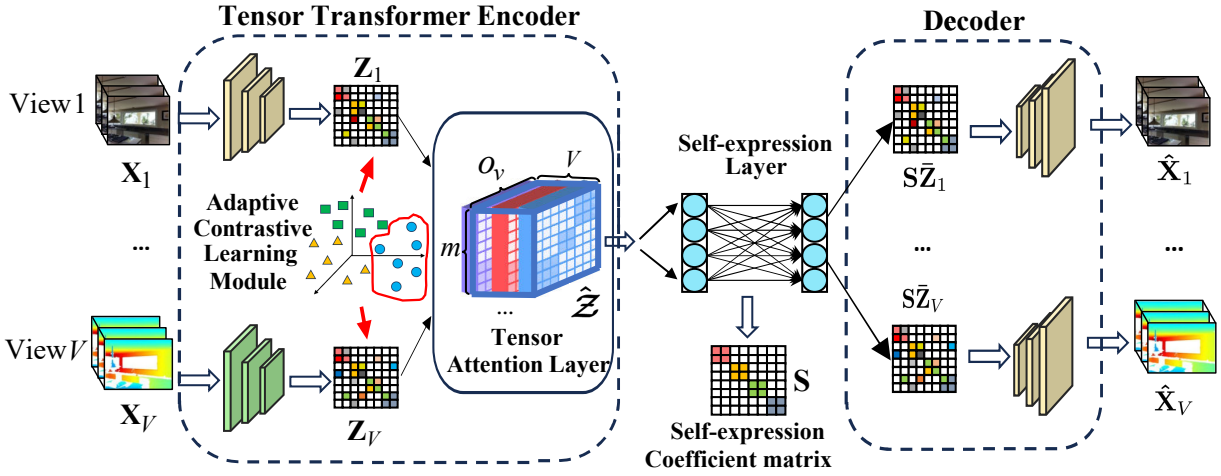

Figure 2: The framework of our proposed method (TTAE). For presentation purposes. The multi-view data $\mathbf{X}_v$, after passing through the feature extraction layer, yields latent features $\mathbf{Z}_v$, which encapsulate an underlying latent cluster structure. These features $\mathbf{Z}_v$ are concatenated into a tensor $\mathcal{Z}$ and fed into the Tensor Attention Layer to facilitate cross-view interactions and obtain enhanced features $\hat{\mathcal{Z}}$. Within each view, $\bar{\mathbf{Z}}_v$ undergoes a self-expression layer to acquire a self-expression coefficient matrix $\mathbf{S}$. In this case, $\mathbf{S}$ constructs relevant nearest neighbor graphs to assist in the selection of positive and negative sample pairs, achieving adaptive contrastive learning.

structure. This matrix reflects the interrelations of data points within the subspace, where a sample is expressed as a linear combination of other samples in the same subspace. The self-expression layer takes $\bar{\mathbf{Z}}_v$ as input and enforces $\bar{\mathbf{Z}}_v = \mathbf{S}\bar{\mathbf{Z}}_v$ to learn a shared self-expression coefficient matrix $\mathbf{S}$ for each view. To promote this self-expression property, the self-expression loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{se} &= \sum_{v=1}^{V} \|\bar{\mathbf{Z}}_v - \mathbf{S}\bar{\mathbf{Z}}_v\|_F^2 + \|\mathbf{S}\|_F^2 \\ s.t., &diag(\mathbf{S}) = 0, \end{aligned} \tag{4}$$

where $diag(\cdot)$ refers to extracting the diagonal elements of the matrix. To avoid trivial solution $\mathbf{S} = \mathbf{I}$, we constrain the diagonal elements of $\mathbf{S}$ to zero, i.e., $diag(\mathbf{S}) = 0$. Throughout network optimization, $\mathbf{S}$ can be interpreted as an adjacency matrix, representing the similarity relationships among samples.

Adaptive Contrastive Learning Module

By minimizing both $\mathcal{L}_{re}$ and $\mathcal{L}_{se}$, the model can effectively capture low-dimensional latent features, along with the embedded cluster structure information encoded in $\mathbf{S}$. However, due to the reconstruction task often yielding broad representations, insufficient for distinguishing relationships among clusters, it can adversely impact subsequent self-expression learning (Wang et al. 2021). Therefore, it requires further fine-tuning to apply to clustering tasks. Simultaneously, we recognize the consistency of high-level semantic information across different views. Hence, we introduce contrastive learning (Chen et al. 2020) to acquire discriminative features. In particular, to circumvent conflicts between the learning of reconstruction space and consistent common semantics (Xu et al. 2022), we engage in contrastive learning on the features obtained by passing $\mathbf{Z}_v$ through an MLP. This approach enhances the discriminative feature extraction of the view-specific encoder.

Given a sample $\mathbf{x}_1^i \in \mathbf{X}_1$ of object i and its corresponding representation $\mathbf{z}_1^i \in \mathbf{Z}_1$, we construct positive pairs using $\mathbf{z}_1^i$ and the feature representation of object i in another view, $\mathbf{z}_2^i$, *i.e.,* $(\mathbf{z}_1^i, \mathbf{z}_2^i)$. Negative pairs are formed by combining $\mathbf{z}_1^i$ with all feature samples other than $\mathbf{z}_1^i$ and $\mathbf{z}_2^i$, *i.e.,* $(\mathbf{z}_1^i, \mathbf{z}_2^k)$ for $k \in [1, m]$ and $(\mathbf{z}_1^i, \mathbf{z}_1^k)$ for $k \neq i$. All these sample pairs are uniformly processed through contrastive learning. For similarity measurement, we adopt the cosine similarity $sim(\cdot, \cdot)$, defined as $sim(\mathbf{z}_1, \mathbf{z}_2) = \frac{\mathbf{z}_1^T \mathbf{z}_2}{\|\mathbf{z}_1\| \|\mathbf{z}_2\|}$.

Combining sample selection with cross-entropy, contrastive learning is achieved by minimizing the following loss function:

$$\mathcal{L}_{cl} = D(\mathbf{Z}_1, \mathbf{Z}_2) = \sum_{i=1}^{m} log \frac{exp(sim(\mathbf{z}_1^i, \mathbf{z}_2^i)))}{\sum_{j=1}^{m} \{exp(sim(\mathbf{z}_1^i, \mathbf{z}_2^j)) + \sum_{v=1}^{2} \mathbb{L}_{(j \neq i)} exp[sim(\mathbf{z}_v^j, \mathbf{z}_v^i)]\}}, \tag{5}$$

where $\mathbb{L}_{(j \neq i)} \in \{0, 1\}$ is an indicator function evaluating to 1 if $j \neq i$.

However, directly selecting other samples as negative sample pairs inevitably leads to treating samples from the same cluster as negative pairs, which hinders the ability to learn discriminative features. To address this issue, some unsupervised approaches (Jing et al. 2021; Trosten et al. 2021) utilize pseudo-labels obtained from clustering algorithms (e.g., k-means) to introduce supervisory signals. Unlike these methods, We prioritize the similarity between paired data over pseudo-labels as self-supervised signals, as

Dataset	Size (Categories)	Dimensions
Fashion-MNIST	2,000 (10)	28×28
COIL-20	1,440 (20)	64×64
FRGC	2,462 (20)	32×32
Youtube-Faces (YTF)	2,000 (41)	55×55
ALOI-100	10,800 (100)	77, 125
Noisy-MNIST	20,000 (10)	784, 784

Table 2: Summary of Datasets.

it aligns more closely with the essence of contrastive learning. Given the reliability of the self-expression learning layer and the consistency embedded in the self-expression coefficient matrix S across views, it is reasonable to assume that the self-expression attributes of the same cluster can assist in identifying these affinity pairs. Specifically, for a given anchor, k affinity pairs (k-neighbor graph) will be excluded as the possibility of negative sample pairs, so the Eq. (5) becomes:

$$L_{acl} = D(\mathbf{Z}_1, \mathbf{Z}_2) = \sum_{i=1}^{m} log \frac{exp(sim(\mathbf{z}_1^i, \mathbf{z}_2^i)))}{\sum_{j=1, j \notin \mathbb{P}(i,k)}^{m} \{exp(sim(\mathbf{z}_1^i, \mathbf{z}_2^j)) + \sum_{v=1}^{2} \mathbb{L}_{(j \neq i)} exp[sim(\mathbf{z}_v^j, \mathbf{z}_v^i)]\}}, \tag{6}$$

where $\mathbb{L}_{(j \neq i)} \in \{0, 1\}$ is an indicator function that evaluated as 1 if $j \neq i$. $\mathbb{P}(i, k)$ refers to the k nearest sample points to anchor point i. The evolution of Eq. (5) to (6) means that the affinity between samples will be used to supervise its representation learning to learn more discriminative features. When integrated with its well-designed sequence of training steps, it effectively alleviates the potential noise effects arising from homogeneous samples.

Overall Loss Function and Optimization

Based on the above network structure, we adopt a joint loss composed of reconstruction loss $\mathcal{L}_{re}$, adaptive contrastive learning loss $\mathcal{L}_{acl}$, and self-expression loss $\mathcal{L}_{se}$, and the total loss $\mathcal{L}_{all}$ can be formalized as below:

$$\mathcal{L} = \mathcal{L}_{re} + \lambda_1 \mathcal{L}_{se} + \lambda_2 \mathcal{L}_{acl}. \tag{7}$$

We train the model in two stages: pre-training and overall training. Detailed steps are outlined in Algorithm 1. *Pre-training network:* TTAE uses reconstruction loss $\mathcal{L}_{re}$ and self-expression loss $\mathcal{L}_{se}$ for pre-training. This step accelerates training, produces robust representations $\mathbf{Z}$ for contrastive learning, and generates reliable guidance $\mathbf{S}$ for improved learning stability. *Overall training network:* The network is initialized with pre-trained parameters and optimized using Eq. (7). During training, the self-expression matrix is iteratively refined to support adaptive contrastive learning. The final shared matrix $\mathbf{S}$ is used to compute the affinity matrix: $\mathbf{C} = \frac{1}{2}(\mathbf{S} + \mathbf{S}^T)$ Spectral clustering is then applied to $\mathbf{C}$ to produce the final clustering results.

Algorithm 1: **TTAE**

Input: Multi-view data $\{\mathbf{X}_v\}_{v=1}^{V}$, hyperparameters λ_1, λ_2, k, and epochs e_1, e_2.

Stage 1: Pre-training (Epochs 1 **to** e_1**):**
for $t = 1$ to e_1 **do**
  Compute latent feature $\mathbf{Z}_v$ and $\boldsymbol{\mathcal{Z}}$
  Compute strengthened feature $\hat{\boldsymbol{\mathcal{Z}}}$ and $\hat{\mathbf{Z}}_v$
  Compute self-expression $\mathbf{S}\hat{\mathbf{Z}}_v$ and reconstruct $\hat{\mathbf{X}}_v$
  Update parameters with $\mathcal{L}_{re}$ and $\mathcal{L}_{se}$.
end for
Stage 2: Fine-tuning (Epochs $e_1 + 1$ **to** e_2**):**
for $t = e_1 + 1$ to e_2 **do**
  Repeat latent and strengthened feature computation.
  Construct k-nearest neighbor graph $\mathbb{P}(i, k)$ using $\mathbf{S}$.
  Reconstruct data and update parameters with $\mathcal{L}_{all}$.
end for
Compute affinity matrix $\mathbf{C} = \frac{1}{2}(\mathbf{S} + \mathbf{S}^T)$.

Output: Return predictions P from spectral clustering on affinity matrix $\mathbf{C}$.

Experienment

Experiment Setup

Dataset Settings: We evaluate our method on six benchmark datasets: Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), COIL-20 (Nene, Nayar, and Murase 1996), FRGC (Yang, Parikh, and Batra 2016), Youtube-Faces (YTF) (Wolf, Hassner, and Maoz 2011), ALOI-100 (Wang, Yang, and Liu 2019), and Noisy-MNIST (Hardoon, Szedmak, and Shawe-Taylor 2004). For large datasets, a random subset was selected, considering the runtime efficiency of most comparative algorithms. Each dataset comprises two complementary views, such as original images and edge maps. However, some datasets deviate from this general structure. For instance, ALOI-100 utilizes RGB features as the first view and Haralick texture features as the second. Similarly, Noisy-MNIST adopts grayscale images for the first view and noisy images with a Gaussian background for the second. Table 2 provides a comprehensive summary of the datasets, including key details such as size, category distribution, and dimensionality.

Implementation details: To ensure a fair comparison, the feature extraction backbone of TTAE is aligned with the majority of methods. Classic subspace algorithms, such as DMSC, employ their own CNNs for feature extraction, and we adopted a similar approach for consistency. For datasets like Fashion-MNIST, COIL-20, FRGC, and YTF, we use a three-layer convolutional network. For vector-based datasets, such as ALOI-100 and Noisy-MNIST, we referenced methods like MFLVC and DCCA, applying a three-layer fully connected network with dimensions {dim, 1024, 1024, 1024, 128}, where dim denotes the input dimension of the dataset. This ensures a reasonable and fair comparison across different datasets. All experiments were conducted on a Windows 11 platform equipped with NVIDIA 4090 Graphics Processing Units (GPUs) and 64 GB of memory. The ADAM optimizer (Kingma and Ba 2014) was em-

Methods	Fashion-MNIST		COIL-20		FRGC		YTF		ALOI-100		Noisy-MNIST	
	ACC	**NMI**	**ACC**	**NMI**	**ACC**	**NMI**	**ACC**	**NMI**	**ACC**	**NMI**	**ACC**	**NMI**
K-means (Hartigan and Wong 1979)	51.27	49.99	57.49	73.22	23.62	27.12	56.01	75.23	46.27	68.43	54.64	48.62
CC (Li et al. 2021)	49.53	46.75	65.27	76.27	34.62	43.08	58.27	77.65	65.31	84.12	79.42	72.16
LALMVC (Xie et al. 2019)	57.20	59.31	64.79	76.83	49.68	57.27	40.85	49.60	49.80	73.71	46.94	56.90
SMVSC (Sun et al. 2021)	65.45	58.97	72.64	81.67	50.73	58.85	62.00	80.14	50.69	73.70	61.76	51.67
EOMSC-CA (Liu et al. 2022)	52.70	56.72	57.43	74.69	31.72	32.45	62.05	83.46	24.21	58.24	59.96	50.03
DCCA (Andrew et al. 2013)	52.74	53.82	55.76	64.91	22.91	24.75	45.19	60.35	49.41	69.62	85.53	89.44
DMSC (Abavisani and Patel 2018)	59.55	65.07	74.10	86.82	72.83	80.96	62.80	80.16	53.40	73.43	69.43	61.72
CMSC-DCCA (Gao et al. 2020)	62.95	68.33	82.64	91.45	70.80	78.55	66.15	82.67	74.60	87.72	87.88	82.20
DMJC (Xie et al. 2020)	61.41	63.41	72.99	81.58	44.07	59.79	61.15	77.40	50.99	70.82	57.16	62.44
DMSC-UDL (Wang et al. 2020)	65.45	68.34	79.24	89.00	73.19	78.19	66.75	82.52	57.83	74.39	72.06	68.97
DMIM (Mao et al. 2021)	56.10	58.06	OOM	OOM	34.12	41.01	57.10	74.80	OOM	OOM	50.79	48.17
MFLVC (Xu et al. 2022)	55.05	54.63	73.12	82.90	45.79	52.28	56.75	70.24	74.90	85.70	92.82	88.79
D2MVSC (Wang et al. 2024)	63.80	67.73	83.77	92.06	73.46	80.29	66.28	82.88	60.91	75.34	72.15	68.55
TTAE	**65.75**	**69.36**	**84.93**	**92.87**	**74.45**	**82.81**	**68.10**	**83.54**	**79.31**	**88.72**	**94.53**	**90.08**

Table 3: The clustering accuracy rate(ACC)(%) and the normalized mutual information(NMI)(%) on six datasets. OOM means out of memory, Best results are highlighted in **bold**, and the second-best results are underlined.

ployed with a default learning rate set to 0.0001.

Experimental Results

Comparison with Existing Approaches: We compare our model with other popular clustering methods, including two classical single-view clustering, *i.e.,* K-means (Hartigan and Wong 1979), and CC (Li et al. 2021); three traditional multi-view clustering, *i.e.,* LALMVC (Xie et al. 2019), SMVSC (Sun et al. 2021), and EOMSC-CA (Liu et al. 2022); eight excellent deep multi-view clustering, *i.e.,* DMJC (Xie et al. 2020), DMSC (Abavisani and Patel 2018), DCCA (Andrew et al. 2013), CMSC-DCCA (Gao et al. 2020), DMSC-UDL (Wang et al. 2020), DMIM (Mao et al. 2021), MFLVC (Xu et al. 2022), D2MVSC (Wang et al. 2024). As k-means and CC can only handle single-view data, we record their best-view clustering performance.

Performance Evaluation: We evaluate the performance of these methods by testing the clustering accuracy (Kuhn 1955) (ACC), normalized mutual information (Xu, Liu, and Gong 2003) (NMI) according to the true labels. Table 3 summarizes the performance of all the methods on the six datasets, with the following findings: 1) Through joint learning of contrastive constraints and self-expression constraints, TTAE achieves the excellent performance of ACC and NMI on all six datasets. 2) Among deep MVSC methods, the proposed model TTAE achieves better results than DMSC, DMSC-UDL, CMSC-DCCA. These methods focus on pursuing more consistent subspace representations, but they ignore the exploration of complementary information, resulting in suboptimal performance. 3) In the case of two large datasets, ALOI-100 and Noisy-MNIST, methods utilizing CCA such as DCCA and CMSC-DCCA achieved suboptimal results. In contrast, contrastive learning methods such as MFLVC and the proposed method, TTAE, achieved relatively better results due to discriminative representation learning. As we accounted for noise in positive and negative sample pairs by considering self-pairwise supervision, TTAE attained optimal clustering performance.

Ablation Study: In this subsection, we conduct a series of experiments to investigate the effect of each component of the TTAE model. On the Fashion-MNIST and YTF datasets, we evaluate the performance of the model without $\mathcal{L}_{re}$, $\mathcal{L}_{se}$ and $\mathcal{L}_{acl}$ individually. The results are shown in Table 4. One could observe that all loss functions contribute to the improvement of clustering performance, and the combined losses yield optimal clustering performance. Additionally, we explored the impact of the Tensor Attention Layer. Using t-SNE visualization (Van der Maaten and Hinton 2008), we depict the features before and after passing through the Tensor Attention Layer as $\mathbf{Z}$ and $\hat{\boldsymbol{\mathcal{Z}}}$, respectively, as illustrated in Figure 3. Despite the introduction of some unavoidable noisy points into the clean clusters, it is evident that the formed clusters become more concentrated after passing through this module. This enhancement is attributed to the aggregative capacity of cross-view attention.

			Fashion-MNIST		YTF	
$\mathcal{L}_{re}$	$\mathcal{L}_{se}$	$\mathcal{L}_{acl}$	**ACC**	**NMI**	**ACC**	**NMI**
✓			53.50	52.76	54.45	71.56
	✓		52.81	58.06	60.18	78.32
		✓	51.96	51.12	61.10	77.23
	✓	✓	58.15	58.57	64.16	80.26
✓		✓	61.50	63.20	62.95	77.07
✓	✓		62.63	66.19	65.25	81.30
✓	✓	✓	**65.75**	**69.36**	**68.10**	**83.54**

Table 4: Ablation Study on Fashion-MNIST and YTF dataset in terms of ACC (%) and NMI (%).

Parameters Analysis: For the parameters λ, Figure 5 records the effect of λ on the Fashion-MNSIT dataset. In our analysis, the self-expression learning parameter λ_1 and the contrastive learning parameter λ_2 are changed in the range of $\{0.01, 0.1, 1, 10, 100\}$. When $\lambda_1 = 1$ and $\lambda_2 = 0.1$, TTAE achieves the best results on Fashion-MNIST datasets. Clearly, TTAE is robust to λ_2 and an appropriate λ_1 will contribute to achieving optimal clustering performance.

To address the variation in sample size m and the number of classes f across datasets, we propose a dynamic adjustment method for determining the number of nearest neighbors k. Assuming a uniform class distribution, k is defined as $k = \xi\frac{m}{f}$, where ξ is a proportional hyperparameter that

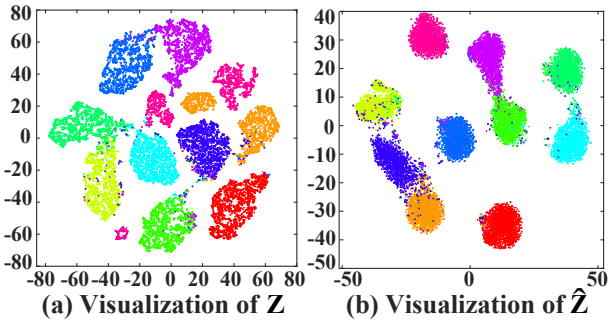

Figure 3: T-SNE visualization on the Noisy-MNIST with NMI = 0.89. The left image (a) corresponds to latent feature $\mathbf{Z}$, while the right image (b) corresponds to strengthened feature $\hat{\mathbf{Z}}$.

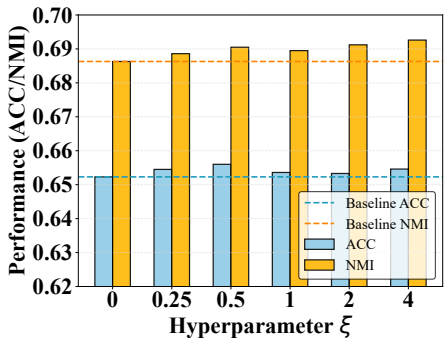

Figure 4: Clustering performance versus different hyperparameter values ξ on Fashion-MNIST, where nearest samples $k = \xi m/f$.

adjusts the threshold for k, thereby controlling the sample reduction ratio. This formulation ensures that k is adaptive to the dataset's characteristics, balancing the number of neighbors relative to the total sample size and class count. Figure 4 illustrates the clustering performance for different values of ξ across datasets. When $\xi = 0$, the adaptive contrastive loss function degenerates to its standard form without dynamic adjustment, as described in Eq. (6). For $\xi > 0$, the adjustment effectively filters irrelevant negative samples, significantly improving clustering performance. These results highlight the utility of the proportional threshold in selecting meaningful neighbors and the critical role of the self-expression layer in supervising the contrastive learning process.

Convergence and Runtime Analysis: We analyze the convergence of TTAE using the loss function in Eq. (7) on Fashion-MNIST. As shown in Figure 6, the loss decreases and converges steadily over iterations (5 epochs as one iteration), while ACC and NMI consistently improve. Although the attention mechanism introduces some computational overhead with a time complexity of $\mathcal{O}(N^2 o_v)$, this is compensated by improved clustering performance. Table 5 summarizes the runtime and performance of different methods after 200 training epochs.

Methods	NMI (%)	Runtime (s)
CMSC-DCCA	68.33	**18.21**
DMSC-UDL	68.34	19.68
MFLVC	54.63	24.17
D2MVSC	67.73	19.11
Ours	**69.36**	20.92

Table 5: Comparison of NMI and runtime for different methods on Fashion-MNIST.

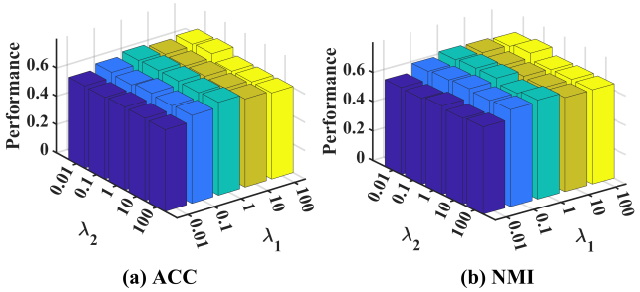

Figure 5: Influence of parameters λ_1, λ_2 changes on clustering performance on Fashion-MNIST dataset.

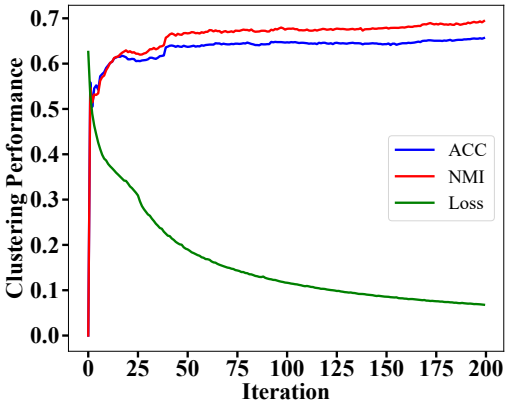

Figure 6: Clustering performance with increasing iteration on Fashion-MNIST.

Conclusion

In this work, we proposed a novel multi-view subspace clustering via Tensor Transformers Autoencoders (TTAE). Specifically, it integrates learning of consistency and complementarity. TTAE effectively captures complementary information by fostering interactive attention across diverse views. Simultaneously, it harmoniously combines self-expression with contrastive learning, where contrastive constraints are used to obtain discriminative features between samples to promote a clear self-expressive cluster structure, *i.e.,* sample affinity, which in turn can supervise pairs of negative samples. Extensive experiments demonstrate that TTAE achieves a significant improvement over several state-of-the-art clustering methods.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant 62176203 and Grant 62102306, the Fundamental Research Funds for the Central Universities, the Natural Science Basic Research Program of Shaanxi Province (Grant 2023-JC-YB-534), and the Science and Technology Project of Xi'an (Grant 2022JH-JSYF-0009), Initiative Postdocs Supporting Program (Grant BX20190262), Open Project of Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University (No. MMC202416), and the Xidian Innovation Fund (Project NoYISJ24017).

References

Abavisani, M.; and Patel, V. M. 2018. Deep multimodal subspace clustering networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(6): 1601–1614.

Akaho, S. 2001. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society*.

Andrew, G.; Arora, R.; Bilmes, J.; and Livescu, K. 2013. Deep canonical correlation analysis. In *ICML*, 1247–1255. PMLR.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8): 1798–1828.

Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *IEEE CVPR*, 586–594.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Fang, U.; Li, M.; Li, J.; Gao, L.; Jia, T.; and Zhang, Y. 2023. A Comprehensive Survey on Multi-view Clustering. *IEEE TKDE*, 35(12): 12350–12368.

Feng, W.; Sheng, G.; Wang, Q.; Gao, Q.; Tao, Z.; and Dong, B. 2024. Partial Multi-View Clustering via Self-Supervised Network. In *AAAI*, 11988–11995.

Gao, Q.; Lian, H.; Wang, Q.; and Sun, G. 2020. Cross-modal subspace clustering via deep canonical correlation analysis. In *AAAI*, volume 34, 3938–3945.

Hardoon, D. R.; Szedmak, S.; and Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12): 2639–2664.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1): 100–108.

Huang, Z.; Ren, Y.; Pu, X.; and He, L. 2021. Non-linear fusion for self-paced multi-view clustering. In *ACM Multimedia*, 3211–3219.

Ji, P.; Zhang, T.; Li, H.; Salzmann, M.; and Reid, I. D. 2017. Deep Subspace Clustering Networks. In *NeurIPS*, 24–33.

Jing, B.; Xiang, Y.; Chen, X.; Chen, Y.; and Tong, H. 2021. Graph-MVP: Multi-View Prototypical Contrastive Learning for Multiplex Graphs. *arXiv preprint arXiv:2109.03560*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. In *NeurIPS*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Li, J.; Gao, Q.; Wang, Q.; Yang, M.; and Xia, W. 2023. Orthogonal Non-negative Tensor Factorization based Multi-view Clustering. In *NeurIPS*.

Li, R.; Zhang, C.; Fu, H.; Peng, X.; Zhou, T.; and Hu, Q. 2019. Reciprocal multi-layer subspace learning for multi-view clustering. In *IEEE ICCV*, 8172–8180.

Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J. T.; and Peng, X. 2021. Contrastive clustering. In *AAAI*.

Lin, Y.; Gou, Y.; Liu, Z.; Li, B.; Lv, J.; and Peng, X. 2021. COMPLETER: Incomplete Multi-View Clustering via Contrastive Prediction. In *IEEE CVPR*, 11174–11183.

Liu, C.; Wen, J.; Wu, Z.; Luo, X.; Huang, C.; and Xu, Y. 2023. Information Recovery-Driven Deep Incomplete Multiview Clustering Network. *IEEE TNNLS*, 1–11.

Liu, S.; Wang, S.; Zhang, P.; Liu, X.; Xu, K.; Zhang, C.; and Gao, F. 2022. Efficient One-pass Multi-view Subspace Clustering with Consensus Anchors. In *AAAI*.

Liu, Y.; Yang, X.; Zhou, S.; Liu, X.; Wang, S.; Liang, K.; Tu, W.; and Li, L. 2024. Simple Contrastive Graph Clustering. *IEEE TNNLS*, 35(10): 13789–13800.

Lu, Y.; Lin, Y.; Yang, M.; Peng, D.; Hu, P.; and Peng, X. 2024. Decoupled Contrastive Multi-View Clustering with High-Order Random Walks. In *AAAI*, 14193–14201.

Luo, S.; Zhang, C.; Zhang, W.; and Cao, X. 2018. Consistent and Specific Multi-View Subspace Clustering. In *AAAI*, 3730–3737.

Mao, Y.; Yan, X.; Guo, Q.; and Ye, Y. 2021. Deep mutual information maximin for cross-modal clustering. In *AAAI*, 8893–8901.

Nene, S.; Nayar, S.; and Murase, H. 1996. Columbia university image library (coil-20). *Technical Report CUCS-005-96*.

Orecchia, L.; and Zhu, Z. A. 2014. Flow-based algorithms for local graph clustering. In *ACM-SIAM SODA*, 1267–1286. SIAM.

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

Tang, H.; and Liu, Y. 2022. Deep Safe Incomplete Multi-view Clustering: Theorem and Algorithm. In *ICML*, volume 162, 21090–21110.

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2017. From ensemble clustering to multi-view clustering. In *IJCAI*, 2843–2849.

Trosten, D. J.; Lokse, S.; Jenssen, R.; and Kampffmeyer, M. 2021. Reconsidering representation alignment for multi-view clustering. In *IEEE CVPR*, 1255–1265.

Tzortzis, G.; and Likas, A. 2012. Kernel-based weighted multi-view clustering. In *IEEE ICDM*, 675–684.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Wang, H.; Yang, Y.; and Liu, B. 2019. GMC: Graph-based multi-view clustering. *IEEE TKDE*, 32(6): 1116–1129.

Wang, J.; Wu, B.; Ren, Z.; and Zhou, Y. 2024. Decomposed deep multi-view subspace clustering with self-labeling supervision. *Information Sciences*, 653: 119798.

Wang, Q.; Cheng, J.; Gao, Q.; Zhao, G.; and Jiao, L. 2020. Deep multi-view subspace clustering with unified and discriminative learning. *IEEE TMM*, 23: 3483–3493.

Wang, Q.; Tao, Z.; Gao, Q.; and Jiao, L. 2022a. Multi-view subspace clustering via structured multi-pathway network. *IEEE TNNLS*.

Wang, Q.; Tao, Z.; Xia, W.; Gao, Q.; Cao, X.; and Jiao, L. 2022b. Adversarial multiview clustering networks with adaptive fusion. *IEEE TNNLS*.

Wang, Q.; Xia, W.; Tao, Z.; Gao, Q.; and Cao, X. 2021. Deep Self-Supervised t-SNE for Multi-modal Subspace Clustering. In *ACM Multimedia*, 1748–1755.

Wang, S.; Chen, Z.; Du, S.; and Lin, Z. 2022c. Learning Deep Sparse Regularizers With Applications to Multi-View Clustering and Semi-Supervised Classification. *IEEE TPAMI*, 44(9): 5042–5055.

Wang, S.; Li, C.; Li, Y.; Yuan, Y.; and Wang, G. 2023. Self-supervised information bottleneck for deep multi-view subspace clustering. *IEEE TIP*, 32: 1555–1567.

Wolf, L.; Hassner, T.; and Maoz, I. 2011. Face recognition in unconstrained videos with matched background similarity. In *IEEE CVPR*, 529–534.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, D.; Zhang, X.; Gao, Q.; Han, J.; Xiao, S.; and Gao, X. 2019. Multiview clustering by joint latent representation and similarity learning. *IEEE TCYB*, 50(11): 4848–4854.

Xie, Y.; Lin, B.; Qu, Y.; Li, C.; Zhang, W.; Ma, L.; Wen, Y.; and Tao, D. 2020. Joint deep multi-view learning for image clustering. *IEEE TKDE*, 33(11): 3594–3606.

Xu, J.; Tang, H.; Ren, Y.; Peng, L.; Zhu, X.; and He, L. 2022. Multi-level feature learning for contrastive multi-view clustering. In *IEEE CVPR*, 16051–16060.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *ACM SIGIR*, 267–273.

Yan, W.; Zhang, Y.; Lv, C.; Tang, C.; Yue, G.; Liao, L.; and Lin, W. 2023. GCFAgg: Global and Cross-view Feature Aggregation for Multi-view Clustering. In *IEEE CVPR*, 19863–19872.

Yang, J.; Parikh, D.; and Batra, D. 2016. Joint unsupervised learning of deep representations and image clusters. In *IEEE CVPR*, 5147–5156.

Yang, X.; Liu, Y.; Zhou, S.; Wang, S.; Tu, W.; Zheng, Q.; Liu, X.; Fang, L.; and Zhu, E. 2023. Cluster-Guided Contrastive Graph Clustering Network. In *AAAI*, 10834–10842.

Zhang, C.; Fu, H.; Hu, Q.; Cao, X.; Xie, Y.; Tao, D.; and Xu, D. 2018. Generalized latent multi-view subspace clustering. *IEEE TPAMI*, 42(1): 86–99.

Zhang, C.; Geng, Y.; Han, Z.; Liu, Y.; Fu, H.; and Hu, Q. 2024a. Autoencoder in Autoencoder Networks. *IEEE TNNLS*, 35(2): 2263–2275.

Zhang, C.; Lou, Z.; Zhou, Q.; and Hu, S. 2023a. Multi-view clustering via triplex information maximization. *IEEE TIP*, 32: 4299–4313.

Zhang, Y.; Yang, Y.; Li, T.; and Fujita, H. 2019. A multitask multiview clustering algorithm in heterogeneous situations based on LLE and LE. *Knowledge-Based Systems*, 163: 776–786.

Zhang, Z.; Wang, Q.; Pei, C.; and Gao, Q. 2024b. Deep cross-modal subspace clustering with Contrastive Neighbour Embedding. *Neurocomputing*, 576: 127318.

Zhang, Z.; Wang, Q.; Tao, Z.; Gao, Q.; and Feng, W. 2023b. Dropping Pathways Towards Deep Multi-View Graph Subspace Clustering Networks. In *ACM Multimedia*, 3259–3267.

Zhao, B.; Wang, Q.; Tao, Z.; Feng, W.; and Gao, Q. 2024. DFMVC: Deep Fair Multi-view Clustering. In *ACM Multimedia*, 8090–8099.

Zhong, H.; Wu, J.; Chen, C.; Huang, J.; Deng, M.; Nie, L.; Lin, Z.; and Hua, X.-S. 2021. Graph contrastive clustering. In *IEEE ICCV*, 9224–9233.

Zhou, L.; Du, G.; Lü, K.; Wang, L.; and Du, J. 2024. A Survey and an Empirical Evaluation of Multi-view Clustering Approaches. *ACM Computing Surveys*, 56(7): 1–38.

Zong, L.; Zhang, X.; Zhao, L.; Yu, H.; and Zhao, Q. 2017. Multi-view clustering via multi-manifold regularized non-negative matrix factorization. *Neural Networks*, 88: 74–89.