Simple Contrastive Multi-View Clustering with Data-Level Fusion

Caixuan Luo1,2, Jie Xu3,*, Yazhou Ren3, Junbo Ma4 and Xiaofeng Zhu1,3
1School of Computer Science and Engineering, Guangxi Normal University, Guilin 541004, China
2Guangxi Key Lab of Multi-source Information Mining & Security, Guilin 541004, China
3School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
4School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

Abstract

Previous deep multi-view clustering methods usually design un-shared encoders to explore the cluster information among multi-view data, but they are difficult to customize the encoders for individual views and easily increase information loss. To address these issues, we propose a simple yet effective contrastive multi-view clustering framework. Specifically, different from using feature-level fusion in previous methods, we first propose a data-level fusion method to fuse multi-view information, which produces a fused data to replace all views and thus avoids customizing networks for different views. Then, we simulate the data noise and un-availability in multiple views to design two kinds of data augmentation for the fused data, making a shared encoder with simple contrastive learning to learn robust features and achieve the interaction across views. As a result, our method is a general framework and we base on it to conduct feature clustering and end-to-end clustering. Extensive experiments demonstrate that our method can explore the discriminative information in multi-view data and achieve superior clustering performance.

1 Introduction

Multi-View Clustering (MVC) can leverage rich information among multiple views to explore comprehensive cluster structures in multi-view datasets [Bickel and Scheffer, 2004; Xu et al., 2013; Zhang et al., 2024], so it has been become an important subdivision for the unsupervised clustering analysis. In MVC, learning representative features from multi-view data plays a decisive role in clustering performance, and works with different feature learning methods have been proposed [Ren et al., 2022; Zhang and He, 2023; Chen et al., 2022]. Motivated by the recent success of deep learning, research interest is largely paid to study deep MVC with neural networks [Wen et al., 2022; Fang et al., 2023].

According to the methodology for interacting multi-view information, existing deep MVC methods can be summarized into two categories, i.e., feature-level fusion in Figure 1(a) and feature-level consistency in Figure 1(b). Feature-level fusion first utilizes un-shared encoder networks to learn deep features of different views, and then establishes a fusion module at the feature-level to achieve information fusion across all views. On the fused feature, early work directly applies traditional single-view clustering to MVC, such as subspace methods and spectral methods [Abavisani and Patel, 2018; Huang et al., 2019], where decoder networks are usually stacked behind fusion modules to regularize the fused feature. Subsequent work incorporates weighting or attention strategies into fusion modules to quantify the importance of views [Zhou and Shen, 2020; Yin et al., 2020]. Feature-level consistency similarly employs different encoders to learn deep features for individual views, but then conducts consistency optimization objective among features to explore their mutual information, such as CCA methods and contrastive learning methods [Andrew et al., 2013; Wang et al., 2015; Tian et al., 2020]. Recently, contrastive learning based deep MVC showcases great success, where multiple views of a sample are used to construct positive pairs and their features’ consistency is encouraged by minimizing contrastive loss [Lin et al., 2021; Xu et al., 2023]. For example, [Trosten et al., 2021] first perform contrastive learning among multiple features and then fuse them to generate clustering predictions. [Xu et al., 2022] propose to learn multi-level features and clustering predictions for different views without feature fusion. The contrastive MVC methods could learn instance-discriminative features by leveraging views to self-supervise each others, and has inspired a lot of work to advance different issues in deep MVC [Lin et al., 2022; Trosten et al., 2023; Liu et al., 2023; Yan et al., 2023; Jin et al., 2023; Chen et al., 2023; Yang et al., 2023].

While remarkable progress has been made by existing deep MVC methods, deep MVC still faces significant challenges. First, the methods including both feature-level fusion and feature-level consistency need to build un-shared encoder networks for different views to learn features. However, the diversity of multi-view data (varying dimensions, sparsity, and data formats) renders the customization of encoders nearly impossible. Second, existing methods often employ encoders of the same structure for different views, resulting in suboptimal solutions and model redundancy. Third, the features obtained through more encoders might increase the risk of losing inherent information of data, thereby hindering the sub-
sequent feature-level fusion or consistency operations from exploring the useful discriminative information across views.

To address the aforementioned issues, we propose a novel framework entitled *SCM: Simple Contrastive Multi-view clustering with data-level fusion* as shown in Figure 1(c). Firstly, to avoid using multiple encoder networks as in previous methods, we propose shifting the fusion step from the feature-level to the data-level. In order to ensure that the fused data retains the information within multi-view data, we employ normalization and concatenation operations to achieve the data-level fusion without other operations. In this way, the discriminative information of different views can be encapsulated within different dimensions of the fused data, allowing us to search for data partitions in the fused data space, and thus handle multi-view learning problems as conveniently as single-view learning with a shared encoder network. Secondly, to enhance the robustness of deep model towards data noise and unavailability in real-world multi-view scenarios, we propose noise multi-view data augmentation and missing multi-view data augmentation to process the fused data. Furthermore, we employ instance-discriminative contrastive learning on the two types of augmented data, ensuring that the learned features are conducive to explore cross-view discriminative feature patterns. The fused data will establish a bridge to mitigate the gap between multi-view learning and single-view learning, and avoid the redundancy and customization issues of multi-view encoder networks.

Specifically, we express the data-level fusion as a function:

\[
X = \mathcal{F}(X^1, X^2, \ldots, X^V)
= [\mathcal{N}(X^1), \mathcal{N}(X^2), \ldots, \mathcal{N}(X^V)],
\]

where \(\mathcal{N}(\cdot)\) denotes the min-max normalization that brings the variables of different views into a uniform scale without distorting the ranges of values, \([\cdot]\) is the concatenation operation, and \(X \in \mathbb{R}^{N \times D}(D = \sum_v d_v)\) is defined as the fused data for all views. In this way, the discriminative information of different views can be encapsulated within different dimensions of the fused data. Traditional feature-level fusion after dimensionality reduction with different encoders might result in the loss of information due to data processing inequality.

---

**Figure 1**: Comparison of deep MVC frameworks. (a) Feature-level fusion. (b) Feature-level consistency. (c) Our SCM: Data-level fusion obtains the fused data \(\bar{X}\) which maintains the discriminative information across multi-view data \(\{X^1, \ldots, X^V\}\). Noise\&Missing multi-view data augmentation (DA\(v\) and DA\(m\)) produce \(\bar{X}^v\) and \(\bar{X}^m\), which are then fed into a shared encoder for obtaining features \(\bar{Z}^v\) and \(\bar{Z}^m\), respectively. Contrastive learning makes \(\bar{Z}^v\) and \(\bar{Z}^m\) interact with each other to learn discriminative information from multi-view data, as well as make the model robust to data noise and unavailability. The trained encoder obtains the final feature \(\bar{Z}\) of data \(\bar{X}\) for clustering.

---

2 Method

In this paper, we propose our SCM framework as shown in Figure 1(c), which not only can avoid the model customization for different views (this issue will exist in previous deep MVC methods as Figure 1(a-b)) by our data-level fusion with a shared network, but also can learn robust features by our multi-view data augmentation with contrastive learning.

2.1 Data-Level Fusion and Data Augmentation

To begin with, we introduce the basics in SCM framework, i.e., data-level fusion and multi-view data augmentation.

**Data-level fusion** Unlike previous methods that perform fusion operations at the feature-level, we advocate for data-level fusion of multi-view data to synergistically utilize the discriminative information of multiple views. The fused data will establish a bridge to mitigate the gap between multi-view learning and single-view learning, and avoid the redundancy and customization issues of multi-view encoder networks.

Specifically, we express the data-level fusion as a function:

\[
\bar{X} = \mathcal{F}(X^1, X^2, \ldots, X^V)
= [\mathcal{N}(X^1), \mathcal{N}(X^2), \ldots, \mathcal{N}(X^V)],
\]
Hence, in our data-level fusion, we refrain from employing complex mappings to ensure that the fused data retains the discriminative information from all views’ raw data.

**Multi-view data augmentation** Motivated by the beneficial effects of data augmentation [Shorten and Khoshgoftaar, 2019] in computer vision, we propose multi-view data augmentation techniques targeted for the fused data to increase model representation ability, which simulates the scenarios of noisy and unavailable data in practical multi-view learning.

Specifically, considering the presence of noisy data within multi-view datasets, we design the *noisy multi-view data augmentation* by adding noise on some views for each sample:

$$
\overline{X}_i^n = f_{DA}^n(X_i^n, \rho, \sigma) = [N_i^1, N_i^2, \ldots, N_i^V].
$$

(2)

To be specific, we denote the $i$-th view data in the fused data as $\overline{X}_i^n = N(X_i^n) \in \mathbb{R}^{N \times d_n}$. Then, for the $i$-th data $\overline{X}_i^n \in N_i^v$, its noise-augmented data $\overline{n}_i^n \in N_i^n$ is generated by

$$
\overline{n}_i^n = \begin{cases}
X_i^n + \epsilon, & \text{if } \overline{\delta}_i^n < p \\
X_i^n, & \text{else}
\end{cases}
$$

(3)

where $\overline{\delta}_i^n$ is randomly sampled from an uniform distribution, and $\epsilon \in \mathbb{R}^{d_v}$ is random noise sampling from a Gaussian distribution $N(0, \sigma^2)$. $p$ is a threshold that controls the proportion of noise-augmented data within multi-view data.

Further, given the case of data unavailability in multi-view datasets, we also design the *missing multi-view data augmentation* by masking values on some views for each sample:

$$
\overline{X}_i^m = f_{DA}^m(\overline{X}_i^m, \rho) = [M_i^1, M_i^2, \ldots, M_i^V].
$$

(4)

To be specific, we denote the $i$-th view data in the fused data as $\overline{X}_i^m = N(X_i^m) \in \mathbb{R}^{N \times d_m}$. Then, for the $i$-th data $\overline{X}_i^m \in N_i^v$, its missing-augmented data $\overline{m}_i^m \in M_i^m$ is generated by

$$
\overline{m}_i^m = x_i^v \cdot a_{iv}, \quad s.t. \sum_{v=1}^V a_{iv} > 0, a_{iv} \in A,
$$

(5)

where $A \in \{0, 1\}^{N \times V}$ is a random indicator matrix, which ensures that for the $i$-th sample, data from different views can be zeroed out to simulate the state of data unavailability while guaranteeing that at least one view remains available. We have $\sum_{v=1}^V \{a_{iv} < V\}/N = r$ where $\{\cdot\}$ denotes the indicator function, and $r$ is a threshold that controls the proportion of missing-augmented data within multi-view data.

The augmented data $\overline{X}^n$ and $\overline{X}^m$ are dynamically generated during training and thus will make SCM can be robust to the data noise and unavailability by the interaction among multiple views. Moreover, $\overline{X}^n, \overline{X}^m \in \mathbb{R}^{N \times D}$ are the same in data format and can be trained with a shared deep model.

### 2.2 Contrastive Clustering in SCM Framework

Given the fused data and its augmented data, we present our SCM equipped with contrastive learning, as well as its variants with reconstruction regularization and end-to-end prediction.

**Contrastive learning** For the augmented data $\overline{X}^n$ and $\overline{X}^m$, we utilize a shared encoder $E_\theta$ (parameterized by $\theta$) to extract their features $\overline{Z}^n \in \mathbb{R}^{N \times Z}$ and $\overline{Z}^m \in \mathbb{R}^{N \times Z}$, respectively:

$$
\begin{align*}
\overline{Z}^n &= E_\theta(\overline{X}^n), \\
\overline{Z}^m &= E_\theta(\overline{X}^m).
\end{align*}
$$

(6)

We then apply instance-discriminative contrastive learning on $\overline{Z}^n$ and $\overline{Z}^m$, to explore discriminative information across multiple views within the fused data. Specifically, for a mini-batch samples $B$, $\{\overline{z}_i^n \in \overline{Z}^n, \overline{z}_j^m \in \overline{Z}^m\}_{i=1, \ldots, |B|}$ are $|B|$ positive pairs. For each $\overline{z}_i^n$, its $(|B| - 2)$ negative pairs is $\{\overline{z}_j^n, \overline{z}_j^m\}_{j \neq i}^m$ and $\{\overline{z}_i^n, \overline{z}_j^m\}_{j \neq i}^m$ is denoted as a set $s^-$. The InfoNCE [Oord et al., 2018] loss for a single positive pair with multiple negative pairs is given by:

$$
\mathcal{L}^i_C = -\log \frac{\exp(C(\overline{z}_i^n, \overline{z}_i^m) / \tau)} {\sum_{\overline{z} \in s^-} \exp(C(\overline{z}_i^n, \overline{z}) / \tau)}.
$$

(7)

where $\tau$ denotes a temperature parameter and the distance between two sample features (e.g., $\overline{z}_i^n \in \overline{Z}^n$ and $\overline{z}_j^m \in \overline{Z}^m$) is measured by cosine similarity:

$$
C(\overline{z}_i^n, \overline{z}_j^m) = \frac{\overline{z}_i^n \cdot \overline{z}_j^m}{\|\overline{z}_i^n\| \cdot \|\overline{z}_j^m\|}.
$$

(8)

The overall InfoNCE loss for the batch is defined as follows:

$$
\mathcal{L}_{CO} = \frac{1}{|B|} \sum_{i=1}^{|B|} (\mathcal{L}^i_C + \mathcal{L}^m_C).
$$

(9)

In this SCM framework, contrastive learning encourages the encoder to map positive pairs closer together relative to negative pairs in the feature space, thus learning to discriminate different samples of the fused data. Specially, the feature $\overline{Z}^n$ can learn from noiseless views in data $\overline{X}^n$, and the feature $\overline{Z}^m$ can learn from available views in data $\overline{X}^m$, so we achieve the interaction of multi-view information in a shared model and aim to increase the model robustness. Then, we design model regularization and end-to-end prediction for clustering.

**Reconstruction regularization** Since clustering is an unsupervised learning task, using a decoder network to reconstruct the original data from learned features can naturally create a self-supervised signal, that encourages the capture of discriminative structures hidden within the data. This reconstruction regularization has been successfully applied to many deep MVC methods [Trost et al., 2023], which typically use unshared decoder networks for reconstructing different views. Within the context of our proposed multi-view data augmentation, we introduce a novel approach by employing a shared decoder network for achieving reconstruction regularization.

Specifically, we leverage a shared decoder network $D_\phi$ (parameterized by $\phi$) and establish three different reconstruction losses for the fused data and augmented data as follows:

$$
\begin{align*}
\mathcal{L}_{RE}^n &= \frac{1}{|B|} \sum_{i=1}^{|B|} \|\overline{x}_i - D_\phi(\overline{h}_i^n)\|^2_2, \overline{h}_i^n \in \overline{H}^n, \\
\mathcal{L}_{RE}^m &= \frac{1}{|B|} \sum_{i=1}^{|B|} \|\overline{x}_i - D_\phi(\overline{h}_i^m)\|^2_2, \overline{h}_i^m \in \overline{H}^m, \\
\mathcal{L}_{RE}^* &= \frac{1}{|B|} \sum_{i=1}^{|B|} \|\overline{x}_i - D_\phi(\overline{h}_i)\|^2_2, \overline{h}_i \in \overline{H},
\end{align*}
$$

(10)

where $\overline{H}^n, \overline{H}^m, \overline{h}$ are the hidden features between data $\overline{X}^n, \overline{X}^m, \overline{X}$ and features $\overline{Z}^n, \overline{Z}^m, \overline{Z}$, respectively. $\phi$ denotes partial network parameters within $\theta$. For example, we have $\overline{H}^n = E_\theta'(\overline{X}^n)$ and $\overline{Z}^n = E_\theta(\overline{X}^n) = E_{\theta, \theta'}(\overline{H}^n)$. It is noteworthy that we impose the reconstruction loss on $\overline{H}^n, \overline{H}^m, \overline{h}$ to avoid the inductive conflicts with the contrastive loss applied on $\overline{Z}^n, \overline{Z}^m$ [Xu et al., 2022]. Then, the
overall reconstruction loss is formulated as follows:

$$L_{RE} = L_{RE}^n + L_{RE}^m + L_{RE}^\perp.$$  

This overall reconstruction loss could be viewed as a combination of different reconstruction regularization [Xu et al., 2023]. Concretely, $L_{RE}^n$ encourages the hidden feature $\mathbf{H}^n$ to reconstruct the noiseless data $\mathbf{X}$ from noise-augmented data $\mathbf{X}^n$. $L_{RE}^\perp$ enables the hidden feature $\mathbf{H}^\perp$ to reconstruct the available data $\mathbf{X}$ from noiseless-augmented data $\mathbf{X}^\perp$. $L_{RE}^m$ regularizes the hidden feature $\mathbf{H}$ obtained by the noiseless and available data $\mathbf{X}$, for usage in subsequent clustering tasks.

**End-to-end clustering** Existing deep MVC methods for achieving end-to-end clustering primarily employ two strategies: I) In feature-level fusion methods, they often obtain cluster pseudo-labels on the fused features and then train a cluster network through self-training; II) For feature-level consistency methods, they usually set up separate cluster networks for different views and achieve consistent clustering through contrastive learning. It is worth noting that existing methods tend to require pre-setting the number of clusters $K$ to design the dimensionality of the model’s cluster network for each dataset, and our experiments in Section 3.2 find that fixed $K$ is harmful for clustering performance. To make our model architecture compatible with different $K$s of datasets, we propose end-to-end clustering on the basis of our SCM.

Specifically, we add a $H$-dimensional cluster network behind the feature $Z$ and obtain clustering labels $Q \in \mathbb{R}^{N \times H}$:

$$Q = \text{Softmax}(R_{\omega}(Z)), \quad (12)$$

where $R_{\omega}$ is a linear MLP network that organizes the dimensionality of clustering prediction to $H$. To extract known clustering structure information from $Z$, we utilize a clustering method that does not depend on class number, such as Density Peaks [Rodriguez and Laio, 2014], which automatically searches the feature space to obtain a set of anchor points $A = \{a_j\}_{j=1}^{|A|}, a_j \in \mathbb{R}^Z$. Further, through nearest neighbor assignment, we obtain the clustering labels $\tilde{P} \in \{0,1\}^{N \times H}$ ($p_{ij} \in \tilde{P}$) for $N$ samples as follows:

$$\begin{align*}
p_{ij} = 1, & \quad j^* = \arg \min_j \|z_i - a_j\|_2, \\
p_{ij} = 0, & \quad j \neq j^*.
\end{align*} \quad (13)$$

Actually, we have $\tilde{P} = \{(0,1)^{|A|}; \{0\}^{N \times (H-|A|)}\}$. As a result, the first $|A|$ dimensions of the matrix $\tilde{P}$ contain the cluster information in the learned features, and the last $(H - |A|)$ dimensions are all zeros. To achieve end-to-end clustering, we minimize the following mean squared error:

$$L_{EC} = \frac{1}{|B|} \sum_{i=1}^{|B|} \|\tilde{p}_i - q_i\|_2^2, \quad \tilde{p}_i \in \tilde{P}, q_i \in Q. \quad (14)$$

In implementation, the dimensionality number $H$ can be set much larger than the potential class number of dataset, thus decoupling the design of the model’s neural network structure from specific dataset. The final clustering prediction can still be obtained as follows, e.g., for the $i$-th sample:

$$y_i = \arg \max_{\tilde{q}_{ij}} q_{ij}, \quad q_{ij} \in Q. \quad (15)$$

---

**Algorithm 1:** The training steps of SCM framework

**Input:** Multi-view dataset $\{X^1, X^2, \ldots, X^V\}$

**Setting:** SCM($\lambda_1, \lambda_2 = 0$), SCM$_{RE}($$\lambda_1 = 1, \lambda_2 = 0$), SCM$_{EC}($$\lambda_1 = 0, \lambda_2 = 1$), SCM$_{EC+RE}($$\lambda_1, \lambda_2 = 1$), rates of data augmentation $p, r$, std $\sigma$, batch size $|B|$, network parameters $\theta, \phi, \omega$, learning rate $\eta$

**Data-level fusion** $X = F(X^1, X^2, \ldots, X^V)$

**while not converging do**

- **if $\lambda_1 = 0$ then**
  - Compute $L = L_{CO}$
  - Update $\theta \leftarrow \theta - \eta \nabla L(\theta)$
- **else**
  - Compute $L = L_{CO} + L_{RE}$
  - Update $\theta, \phi \leftarrow \theta, \phi - \eta \nabla L(\theta, \phi)$

**while not converging do**

- **if $\lambda_2 = 1$ then**
  - Compute $L = L_{CO}$
  - Update $\theta, \omega \leftarrow \theta, \omega - \eta \nabla L(\theta, \omega)$
- **else**
  - Compute $L = L_{CO} + L_{RE} + L_{EC}$
  - Update $\theta, \phi, \omega \leftarrow \theta, \phi, \omega - \eta \nabla L(\theta, \phi, \omega)$

**Output:** $Z$ for SCM/SCM$_{RE}$, $\tilde{Q}$ for SCM$_{EC/EC+RE}$

Finally, we summarize the loss function in our method as

$$L = L_{CO} + \lambda_1 L_{RE} + \lambda_2 L_{EC}. \quad (16)$$

We leverage $\lambda_1, \lambda_2 \in \{0,1\}$ to obtain different variants of SCM framework: SCM($\lambda_1, \lambda_2 = 0$), SCM$_{RE}($$\lambda_1 = 1, \lambda_2 = 0$), SCM$_{EC}($$\lambda_1 = 0, \lambda_2 = 1$), SCM$_{EC+RE}($$\lambda_1, \lambda_2 = 1$), whose training steps are shown in Algorithm 1.

**Complexity analysis** We let $E$ denote the total training epochs, $V$ and $N$ are the number of views and the sample size of a dataset, $|B|$ is the batch size in the mini-batch optimization. For each batch, the computational complexity of multi-view data augmentation is $O(|B|^2)$, that of contrastive loss, reconstruction loss, and end-to-end clustering loss are $O(|B|^2)$, $O(|B|)$, and $O(|B|)$, respectively. The computational complexity to obtain clustering labels is $O(N)$. For $E$ training epochs, the computational complexity approximates $O(N) + (EN/|B|)O(|B|^2)$ which is linear to sample size $N$. In terms of the memory consumption of deep model, the complexity in our SCM is $1/V$ of that in previous deep MVC methods, as the deep model in SCM is a shared network while that of other methods need $V$ individual networks.
3 Experiment

3.1 Experimental Setup

Datasets We conduct experiments on 8 public datasets, including BDGP [Cai et al., 2012], DIGIT [Peng et al., 2019], Fashion [Xiao et al., 2017], NGs [Hussain et al., 2010], VOC [Everingham et al., 2010], WebKB [Sun et al., 2007], DHA [Lin et al., 2012], and COIL-20 [Nene et al., 1996].

Baselines The comparison methods include K-Means [MacQueen, 1967], Density peak clustering [Rodriguez and Laio, 2014], and deep MVC methods CPSSPAN [Jin et al., 2023], CVCL [Chen et al., 2023], DSMVC [Tang and Liu, 2022a], DMSVC [Tang and Liu, 2022b], MFLVC [Xu et al., 2022].

Implementation details To facilitate a fair comparison, we employed the same network architecture in [Xu et al., 2022; Tang and Liu, 2022a] to implement SCM. It is important to note that whereas previous methods necessitated multiple encoder-decoder networks, our SCM framework requires only one. Specifically, the encoder network structure can be depicted as a fully connected (Fc) MLP [Rosenblatt and others, 1962] with the configuration of $X - F_{c500} - F_{c500} - F_{c2000} - H - Z - Q$, and the decoder network structure is $H - F_{c2000} - F_{c500} - F_{c500} - X$. The activation function for the cluster network $Q$ is Softmax, while ReLU [Nair and Hinton, 2010] is used for all other activation functions. For all datasets used in our experiments, the dimensions of $H$, $Z$, and $Q$ were set to 256, 128, and 64, respectively. The optimizer was Adam [Kingma and Ba, 2014] with a learning rate of 0.0003, and the batch size was set to 256. Both the noise and missing rates of multi-view data augmentation were set to 0.25, and the noise variance was 0.4. Our SCM is implemented by PyTorch and its code is available in https://github.com/SubmissionsIn/SCM.

3.2 Result Analysis

Tables 1 and 2 showcase the clustering results of all comparison methods with known and unknown class number, respectively, where mean values of 5 runs are reported. Evaluation metrics include clustering ACC, NMI, and ARI.

### Clustering with known class number

When the class number is known, we report clustering results of comparison methods as shown in Table 1. Obviously, our methods, including vanilla SCM and reconstruction regularized SCM$_{RE}$, achieve superior clustering performance. Specifically, the average results across 8 datasets indicate that our SCM achieved the improvement of 5% in ACC and 2% in NMI compared to the currently best-performing methods, while SCM$_{RE}$ achieved the improvement of 6% in ACC and 5% in NMI. In addition to superior clustering performance, our proposed data-level fusion makes SCM have simple contrastive multi-view learning paradigm, which can avoid the issues of model redundancy and network customization in previous methods.

### Clustering with unknown class number

To further explore the impact of the unknown class number, we report end-to-
end-to-end clustering results of different methods in Table 2. If the class number in models is fixed, previous end-to-end deep MVC methods often yield degraded results. This is because end-to-end clustering methods typically depend on the truth class number of datasets to design their model structures. In contrast, our methods (SCM$_{EC}$ and SCM$_{EC+RE}$) still achieve best performance, for instance, the ACC of SCM$_{EC}$ and SCM$_{EC+RE}$ have 9% and 13% improvements to the best comparison methods, respectively. The reason is that our SCM has the decoupled design between the model structure and the setting of class number, which transfers the problem of sensitive class number from the model structure to the density peak algorithm and is beneficial for its applicability.

### 3.3 Ablation Study

In this part, we first investigate the key contributions in our proposed noise and missing multi-view data augmentation, and then analyze the different components in loss function.

#### Multi-view data augmentation

As shown in Table 3, firstly, SCM w/o DA is a variant without any data augmentation, which achieves unsatisfactory clustering performance. Furthermore, SCM w/ $f_{DA}$ and SCM w/ $f_{DA}'$ are variants that incorporate the noise multi-view data augmentation and the missing multi-view data augmentation defined in this paper, respectively, and they both show significant improvements over SCM w/o DA. Finally, SCM combines two types of data augmentation and achieves further substantial improvements.
confirming the importance of our specially designed noise and missing multi-view data augmentation.

**Loss components** In Table 4, we conduct ablation studies on three losses using SCM$_{RE}$ and SCM$_{EC+RE}$ as baselines. Compared to SCM$_{RE}$, removing the reconstruction regularization loss in variant SCM$_{RE} w/o \mathcal{L}_{RE}$ results in a slight performance decline, while removing the contrastive loss in variant SCM$_{RE} w/o \mathcal{L}_{CO}$ leads to a severe performance decrease. Furthermore, compared to the end-to-end clustering setting of SCM$_{EC+RE}$, removing the end-to-end clustering loss in variant SCM$_{EC+RE} w/o \mathcal{L}_{EC}$ also significantly degrades model performance. These results indicate that $\mathcal{L}_{CO}$ plays the most crucial role in contrastive multi-view clustering, and $\mathcal{L}_{EC}$ is vital for end-to-end clustering, with $\mathcal{L}_{RE}$ serving a supporting role to regularize the feature learning.

### 3.4 Model Analysis

In this part, we conduct visualization analysis on the parameters and the training process in our SCM framework.

**Data augmentation rates** $\{p, r\}$ In SCM framework, we tune the noise multi-view data augmentation rate $p$ and the missing multi-view data augmentation rate $r$ within the range $[0, 0.25, 0.5, 0.75, 1.0]$, with results depicted in Figure 2. We observe that moderately increasing $p$ and $r$ significantly benefits the model in learning precise clustering structures within multi-view datasets. The underlying mechanism is that our noise and missing multi-view data augmentation compel the model to focus on the interaction and complementarity across views, thereby making the feature learning more robust to inherent noise and unavailable samples in multi-view data.

In comparison experiments, $p$ and $r$ were set to 0.25.

**Trade-off parameters** $\{\lambda_1, \lambda_2\}$ In our method, trade-off parameters $\lambda_1, \lambda_2 \in \{0, 1\}$ control the different settings in SCM framework. In Figure 3, we adopt the setting of SCM$_{EC+RE}$ and further explore the sensitivity of $\lambda_1$ and $\lambda_2$ within the range of $[10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3]$. The optimal values of $\lambda_1$ and $\lambda_2$ are different across different datasets and this outcome is within expectations. Trade-off parameters generally have a minimal sensitivity on the performance within the range of $[10^{-3}, 10^3]$. In all experiments, we did not specifically tune $\lambda_1$ and $\lambda_2$, and their values were fixed at 0 or 1 to implement different variants of SCM.

**Training loss and performance** In Figure 4, we record the loss and clustering accuracy curves during the training process of SCM. It is observed that the loss curve exhibits a smooth and consistent decline, indicating that SCM framework has well convergence. Concurrently, the steadily rising clustering accuracy suggests that the model is progressively learning the correct clustering structure of the dataset.

### 4 Conclusion

In this paper, we propose a novel contrastive multi-view clustering framework with data-level fusion, namely SCM. Specifically, our proposed data-level fusion effectively integrates multi-view information and avoids the issues of customization and redundancy of networks in previous methods. Moreover, we define two types of multi-view data augmentation approaches based on the data-level fusion, which enhances the robustness of model towards noisy and unavailable views in multi-view data. We apply the SCM framework to feature clustering and end-to-end clustering with known and unknown class number, and extensive experiments validate its effectiveness and superiority. Our SCM simplifies multi-view contrastive learning with a shared deep network, and we hope it could bring fresh insights into multi-view learning.
Acknowledgments
This work was supported in part by the National Key Research & Development Program of China under Grant 2022YFA1004100, Medico-Engineering Cooperation Funds from University of Electronic Science and Technology of China under Grant ZYGX2022YGRH009 and Grant ZYGX2022YGRH014, Guangxi Natural Science Foundation (2023GXNSFBA026010), Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (22-A-03-02), Innovation Project of Guangxi Graduate Education (XJCY2022009).

References


