Bridging Optimization and Neural Networks for Efficient Multi-view Clustering

Huilang Xu[1], **Xiang-Xiang Su**[1], **Simin Chen**[1], **Guang-Yong Chen**[1*], **Xing Chen**[1*]

[1]College of Computer and Data Science, Fuzhou University, Fuzhou, China
xuhuilang1999@outlook.com, sxxdyx0619@163.com, 241020071@fzu.edu.cn,
cgykeda@mail.ustc.edu.cn, chenxing@fzu.edu.cn

Abstract

Multi-view clustering (MVC) seeks to uncover the intrinsic group structures embedded in multi-view data, which has attracted considerable attention in recent years. Existing approaches predominantly concentrate on incorporating suitable model priors to capture consistency across views. However, these explicit constraints often fail to hold in scenarios involving significant modal differences between views or the presence of noise, thereby limiting the efficacy of these methods in more complex contexts. To address these issues, this paper introduces BONE, a lightweight and interpretable MVC framework that Bridges Optimization and Neural networks for Efficient MVC. By leveraging learnable parameters to extract high-level features from low-level features derived through classical optimization, BONE integrates the consistency information across views without the need for explicit prior constraints, while eliminating the necessity for pre-training or post-processing. Extensive experiments show that BONE achieves clustering performance comparable to or even better than existing deep MVC methods, while using only 1% of the parameters, offering a new perspective for designing efficient MVC algorithms.

Code — https://github.com/NuclearLemon/BONE

Introduction

Multi-view data refers to data collected from different perspectives of the same object. In an era of increasingly diverse data sources, effectively leveraging the complementary information across such views has become a key research focus in recent years (Yu et al. 2025; Tang et al. 2024). In this context, multi-view clustering (MVC) has gained considerable attention for its ability to uncover the intrinsic cluster structures underlying multi-view data (Guo et al. 2025; Ji and Feng 2025; Cai et al. 2024; Chao, Jiang, and Chu 2024; Sun et al. 2024).

Existing MVC methods can be broadly categorized into traditional methods and deep MVC methods. Traditional methods rely on rigorous formulations, solved iteratively through classical optimization techniques. To achieve discriminative decompositions, these methods frequently incorporate prior constraints or regularization terms designed to enforce consistency across views (Fang et al. 2023). However, such explicit assumptions (e.g., presupposing fully or nearly identical representations across all views) often become invalid when significant modal differences or substantial noise exist within the data, consequently limiting their representational power (Wang et al. 2022b). Additionally, certain priors introduce computationally intensive matrix operations, such as singular value decomposition (SVD) arising from orthogonality constraints, restricting scalability and performance in large-scale settings. Moreover, most traditional methods depend on post-processing clustering algorithms (e.g., K-means or spectral clustering) to obtain final assignments, further hindering their efficacy in handling complex scenarios (Chen et al. 2024).

Deep MVC methods have garnered considerable attention due to their superior feature representation capabilities (Zhou et al. 2024). Unlike traditional methods that explicitly design models, these methods typically employ autoencoders to implicitly capture features from multi-view data through training. They generally adopt a two-stage training strategy: first, pre-training the autoencoder using reconstruction loss, followed by training the network with consistency losses to capture shared semantic information across views. While these methods often achieve superior performance, they require a substantial number of learnable parameters and multiple rounds of pre-training, resulting in significant computational costs (Long et al. 2025). Additionally, these "black-box" models often suffer from low interpretability, posing challenges for analysis and model refinement.

Recently, considerable research has focused on integrating classical model priors with the powerful representational capacity of neural networks (Yan et al. 2025; Wang et al. 2025b; Liu et al. 2023; Wu et al. 2024; Li et al. 2024a). The key motivation behind these hybrid methods is that neural networks can effectively compensate for representational limitations arising from invalid classical priors under challenging conditions, while classical priors mitigate issues associated with excessive parameterization and poor interpretability inherent to deep architectures. Therefore, a natural question arises: *Can we design a lightweight and interpretable framework that efficiently captures cross-view consistency without relying on explicitly crafted prior constraints or post-processing feature fusion?*

To this end, we propose BONE, a lightweight and inter-

*Corresponding authors.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

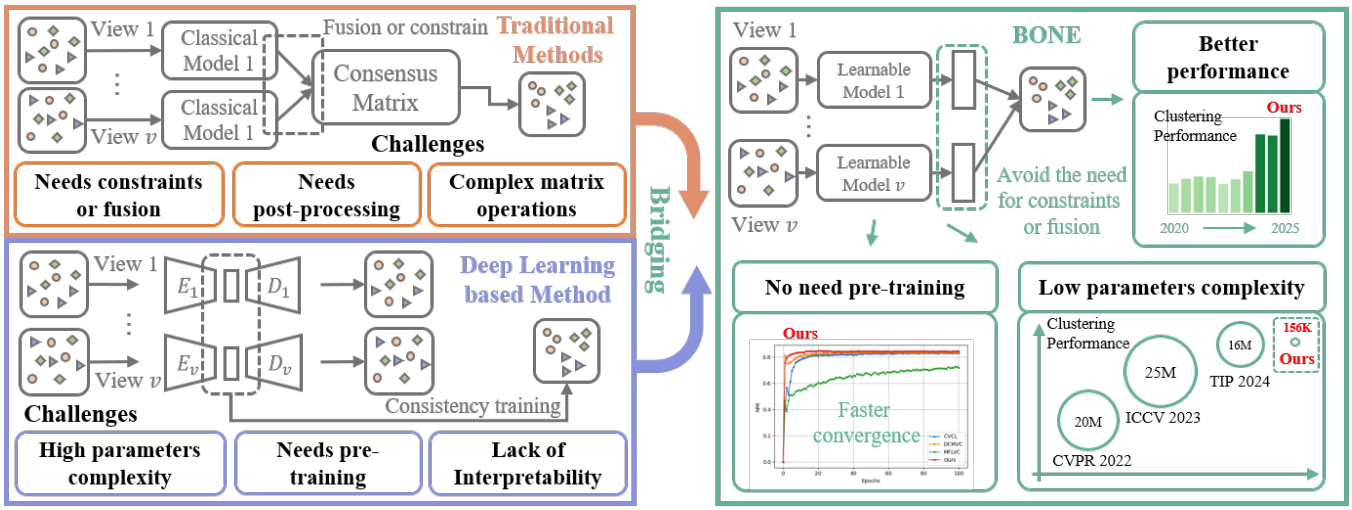


Figure 1: A flexible optimization-guided framework for MVC. Our BONE alleviates reliance on strict prior assumptions and post-processing feature fusion, achieving *better clustering performance*, *faster convergence without pre-training*, and *substantially fewer parameters*. Additionally, due to its optimization-based design, our method provides improved *interpretability* compared to black-box deep methods.

pretable MVC method that Bridges Optimization and Neural networks for Efficient MVC. BONE leverages learnable parameters to extract high-level semantic representations from low-level semantic representations obtained via classical optimization models, thereby efficiently integrating cross-view consistency without relying on explicit priors or post-processing feature fusion, and eliminating the necessity for pre-training or post-processing clustering procedures. Specifically, our method offers the following advantages (as shown in Figure 1):

- Compared to traditional methods, (a) it avoids explicit consistency constraints or post-processing feature fusion by effectively leveraging learnable parameters to integrate consistency information, thereby enhancing clustering performance; (b) it directly obtains clustering assignments in an end-to-end manner, eliminating dependence on post-processing clustering steps; (c) it requires only a small part of the data to train the network, circumventing the computational overhead associated with complex matrix operations in large-scale scenarios.
- Compared to deep learning methods, (a) it extracts high-level semantic representations from low-level representations obtained via classical optimization, achieving comparable or better performance with substantially fewer parameters (approximately 1%); (b) it effectively leverages classical model priors, enabling rapid convergence without the need for complex pre-training, thereby significantly reducing training cost; (c) it offers higher interpretability than existing "black-box" models, facilitating further evaluation and refinement.

The contributions of this paper are summarized as follows:

- **Lightweight & Interpretable One-step MVC Method.** We propose a lightweight and interpretable one-step MVC method, which effectively integrates consistency information across views through learnable parameters, thereby avoiding the need for explicit prior constraints or post-processing feature fusion.
- **Optimization-Guided Neural Network Framework.** We introduce a flexible optimization-guided neural network framework for MVC, bridging classical optimization and deep learning methods, thus providing a novel perspective for designing efficient MVC methods.
- **Compelling Empirical Evidence.** Experiments conducted on widely-used MVC datasets demonstrate that our method achieves clustering performance comparable or superior to existing methods, while using only 1% of the parameters and accelerating convergence.

Related Work

In this section, we review the current research on MVC methods. Consider a multi-view dataset denoted as $\{\mathbf{X}^v \in \mathbb{R}^{m_v \times n}\}_{v=1}^{V}$, comprising V views and n samples, where m_v denotes the feature dimensionality of the v-th view. Let k denote the number of clusters, r the dimensionality of the learned representation. The core challenge in MVC is to effectively balance consistency and complementarity across views. Current MVC approaches can be broadly categorized into traditional and deep MVC methods.

Traditional MVC Methods

Traditional MVC methods typically solve the MVC problem by designing models with prior assumptions. These methods can be further divided into four main categories: subspace learning-based methods (Zhang et al. 2024; Liu et al. 2024b), matrix factorization-based methods (Chen et al. 2024; Wan et al. 2023), graph learning-based methods (Liang et al. 2025; Zhao et al. 2024a; Yu et al. 2024; Li et al. 2024b; Zhao et al. 2024b), and kernel-based methods (Liu

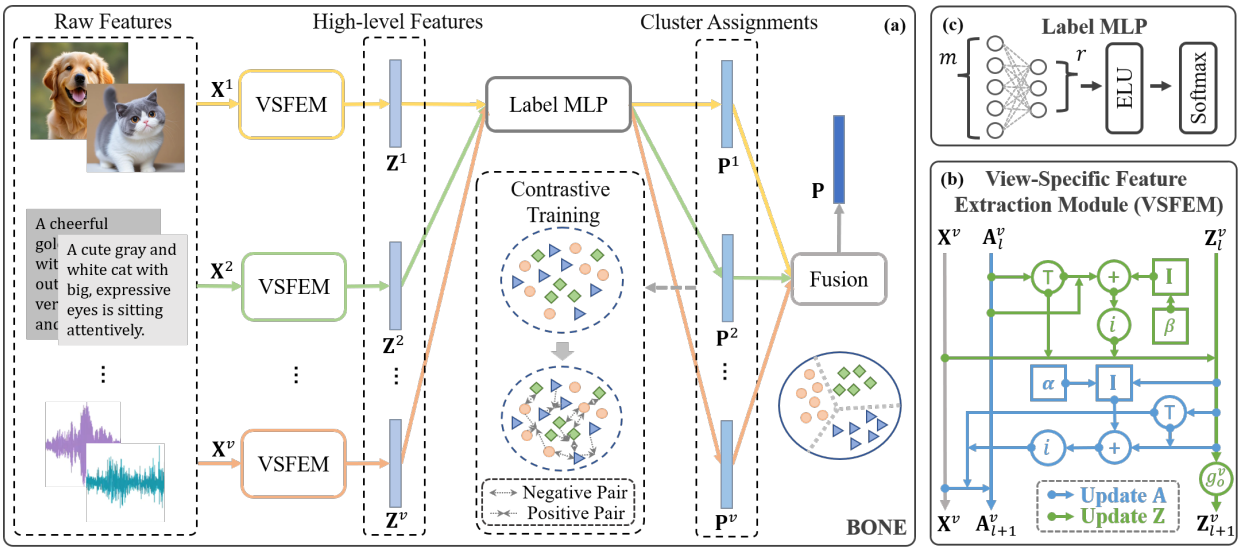


Figure 2: The framework of BONE. We first independently obtain high-level semantic information for each view using view-specific feature extraction modules. Then, we utilize a parameter-sharing label MLP to derive the clustering assignments for each view. By employing contrastive training, we compel the network to capture consistency information across different views. Finally, the clustering assignments are achieved end-to-end.

et al. 2023; Su et al. 2024). A broad category of traditional models can be expressed as:

$$\begin{aligned} \min_{\mathbf{A}^v, \mathbf{Z}^v} & \sum_{v=1}^{V} f_v(\mathbf{X}^v, \mathbf{A}^v, \mathbf{Z}^v) + \phi_v(\mathbf{A}^v) + \psi_v(\mathbf{Z}^v), \\ \text{s.t.} \quad & \mathbf{A}^v \in \mathcal{X}_v, \mathbf{Z}^v \in \mathcal{Y}_v, \end{aligned} \tag{1}$$

where $\mathbf{X}^v \in \mathbb{R}^{m_v \times n}$, and ϕ and ψ represent different regularization terms. The sets $\mathcal{X}_v$ and $\mathcal{Y}_v$ denote the feasible regions for $\mathbf{A}^v$ and $\mathbf{Z}^v$, respectively. For models based on different prior assumptions, $\mathbf{A}^v$ and $\mathbf{Z}^v$ have varying interpretations. For instance, in anchor-based subspace MVC methods, $\mathbf{A}^v$ represents the anchor matrix, and $\mathbf{Z}^v$ represents the representation matrix. In matrix factorization-based MVC methods, $\mathbf{A}^v$ represents the basis matrix, and $\mathbf{Z}^v$ represents the coefficient matrix.

Traditional methods generally achieve consistency between views in two ways. The first approach explicitly constrains the representations or assignment matrices across different views to be identical (Chen et al. 2025; Zhang et al. 2024; Chen et al. 2024; Wang et al. 2021b). However, this strict constraint may overlook the noise and discrepancies naturally occurring between views, potentially limiting the effectiveness of the representations (Zhou et al. 2024).

The second approach adopts relaxed constraints, allowing each view to have its own representation matrix $\mathbf{Z}^v$ (Kong et al. 2025; Zhang et al. 2023; Kang et al. 2020). After obtaining view-specific representations, an additional fusion step is often performed to achieve consistency (e.g., concatenating columns and applying spectral clustering). Although these methods achieve some degree of consistency between views, they fail to fully exploit the inter-view relationships and suffer from the computational overhead of the fusion step like SVD (Wan et al. 2024; Liu et al. 2022).

Another challenge for traditional methods is obtaining clustering assignments after achieving consistent representations. Existing methods typically rely on clustering algorithms (like K-means) to obtain the final assignments. However, this decoupling of the learning process from the final assignment can lead to performance degradation (Liu et al. 2021). Moreover, K-means is computationally expensive on large-scale datasets and highly sensitive to initializations, which limits its applicability in complex scenarios.

Furthermore, several works have explored the incorporation of learnable parameters into the update processes of classical methods to enhance performance (Du et al. 2025; Lin et al. 2025; Wang et al. 2021a). However, they still rely on explicit prior constraints to extract view-consistent representations and depend on extra post-processing steps, making them inadequate for handling complex MVC scenarios.

Deep Learning-based Methods

Deep learning-based methods typically adopt a data-driven approach, training autoencoders to extract features from the original multi-view data (Fang et al. 2023). The pre-training phase is usually guided by reconstruction loss (Trosten et al. 2023), defined as:

$$\mathcal{L}_{\text{pre}} = \sum_{v=1}^{V} \sum_{i=1}^{n} \|\boldsymbol{x}_i^v - f_d^v(f_e^v(\boldsymbol{x}_i^v; \mathbf{W}_e^v); \mathbf{W}_d^v)\|_2^2, \tag{2}$$

where f_e^v and f_d^v represent the encoder and decoder for the v-th view, respectively, and $\mathbf{W}_e^v$ and $\mathbf{W}_d^v$ are learnable parameters. These methods first pre-train the autoencoder using reconstruction loss (Trosten et al. 2021), then apply various training strategies (e.g., contrastive loss and self-

representation loss (Wang et al. 2025b)) to capture the consistency between multi-view data.

Based on the training strategy, deep MVC methods can be divided into two categories: non-contrastive (Wang et al. 2025a, 2024; Xu et al. 2021) and contrastive methods (Xu et al. 2022; Chen et al. 2023; Cui et al. 2024; Lu et al. 2024; Wang et al. 2025b). Due to the remarkable success of contrastive learning in unsupervised learning (Hu et al. 2024), contrastive deep MVC methods have gained more attention in recent years and typically achieve better clustering results (Cui et al. 2024). However, these methods heavily rely on the assumption that autoencoders can effectively extract semantic features, which requires a complex training process and results in higher computational costs (Trosten et al. 2023).

The Proposed Method

This section provides a detailed description of BONE (shown in Figure 2). Our goal is to develop a lightweight and interpretable one-step MVC method that bridges the classical models' priors with the feature extraction capabilities of neural networks. The proposed method requires no preprocessing or post-processing clustering steps and thus is better adapted to handle complex MVC tasks.

View-Specific Feature Extraction Module

The primary challenge in MVC lies in effectively extracting view-specific representations. Traditional methods typically leverage explicit prior models, whereas deep learning-based methods rely on data-driven feature extraction. However, the former often encounter representational limitations due to rigid or invalid priors, while the latter suffer from excessive parameter complexity and computationally expensive pretraining process. To address these issues, we bridge these two approaches and design a lightweight, interpretable view-specific feature extraction module. Specifically, we consider the following classical MVC model:

$$\min_{\mathbf{A}^v,\mathbf{Z}^v} \sum_{v=1}^{V} \|\mathbf{X}^v - \mathbf{A}^v\mathbf{Z}^v\|_F^2 + \alpha\|\mathbf{A}^v\|_F^2 + \beta\|\mathbf{Z}^v\|_F^2, \quad (3)$$

where $\mathbf{A}^v \in \mathbb{R}^{m_v \times r}$, $\mathbf{Z}^v \in \mathbb{R}^{r \times n}$, which is an unconstrained version of (1). We leverage the prior knowledge from this model to design the view-specific feature extraction module. By differentiating and applying algebraic transformations, we derive the following update rules:

$$\mathbf{Z}^v = (\mathbf{A}^{v\top}\mathbf{A}^v + \beta\mathbf{I})^{-1}\mathbf{A}^{v\top}\mathbf{X}^v, \quad (4)$$

$$\mathbf{A}^v = \mathbf{X}^v\mathbf{Z}^{v\top}(\mathbf{Z}^v\mathbf{Z}^{v\top} + \alpha\mathbf{I})^{-1}, \quad (5)$$

where $\mathbf{I} \in \mathbb{R}^{r\times r}$ denotes the identity matrix.

From (4), we observe that the operation $(\mathbf{A}^{v\top}\mathbf{A}^v + \beta\mathbf{I})^{-1}\mathbf{A}^{v\top}$ essentially performs a basis transformation that maps the original m_v-dimensional data to an r-dimensional space. This can be viewed as an optimization-based encoder extracting r-dimensional representations from the original data without any learnable parameters. Instead of directly using data-driven methods to implicitly learn latent structures, we first employ model priors to extract low-level semantic features and then use a limited number of learnable parameters for high-level feature extraction. This approach is expected to significantly reduce the model parameter complexity while obtaining discriminative low-dimensional representations. The process can be expressed as:

$$\mathbf{Z}^v = g_e^v(g_o^v(\mathbf{X}^v), \mathbf{W}_e^v), \quad (6)$$

where g_e^v represents the learnable encoding process for the v-th view, and $g_o^v = (\mathbf{A}^{v\top}\mathbf{A}^v + \beta\mathbf{I})^{-1}\mathbf{A}^{v\top}$ represents the optimization-guided encoding process. This approach has several characteristics:

- The encoding process g_o^v serves as an initial low-level semantic representation extraction step, eliminating the need for complex pre-training and significantly reducing parameter complexity, while maintaining competitive performance.
- By incorporating the feature extraction capabilities of neural networks, this approach mitigates the representational limitations of classical models.
- The output of each layer is interpretable, facilitating further model evaluation and refinement.

We interpret (4) and (5) as iterative updates within each network layer, stacking L layers to construct the final view-specific feature extraction module. Given multi-view input data $\{\mathbf{X}^v\}_{v=1}^V$, the module alternately applies these updates across L layers, ultimately producing $\{\mathbf{A}^{v,L}, \mathbf{Z}^{v,L}\}_{v=1}^V$ (shown in Figure 2 (a) and (b)).

Additionally, since the initialization of $\{\mathbf{A}_0^v\}_{v=1}^V$ significantly influences the performance of traditional methods (Liu et al. 2024a; Wang et al. 2022a), we set the initialization matrices $\{\mathbf{A}_0^v\}_{v=1}^V$ as learnable parameters. Specifically, these learnable matrices are initialized by solving the following standard low-rank matrix factorization problem:

$$\min_{\mathbf{A}_{\text{init}}^v,\mathbf{Z}_{\text{init}}^v} \sum_{v=1}^{V} \|\mathbf{X}^v - \mathbf{A}_{\text{init}}^v\mathbf{Z}_{\text{init}}^v\|_F^2, \text{ s.t. } \mathbf{A}_{\text{init}}^{v\top}\mathbf{A}_{\text{init}}^v = \mathbf{I}. \quad (7)$$

This optimization problem can be efficiently solved via alternating least squares. Given $\mathbf{A}_{\text{init}}^v$, the optimal solution for $\mathbf{Z}_{\text{init}}^v$ is directly obtained by $\mathbf{Z}_{\text{init}}^v = \mathbf{A}_{\text{init}}^{v\top}\mathbf{X}^v$. Subsequently, given $\mathbf{Z}_{\text{init}}^v$, solving for $\mathbf{A}_{\text{init}}^v$ becomes an orthogonal Procrustes problem (Gower and Dijksterhuis 2004), which admits a closed-form solution via SVD:

$$\mathbf{X}^v\mathbf{Z}_{\text{init}}^{v\top} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad \mathbf{A}_{\text{init}}^v = \mathbf{U}\mathbf{V}^\top. \quad (8)$$

We then use the resulting $\mathbf{A}_{\text{init}}^v$ as initial values for the learnable parameters $\mathbf{A}_0^v$.

Remark: The problem in (7) is intended to provide an approximate initialization rather than an exact solution. It can efficiently be solved through random sampling and limited iterations, making it well-suited for large-scale scenarios.

Consistency Feature Fusion Module

After obtaining the view-specific representations $\{\mathbf{Z}^v\}_{v=1}^V$, the key challenge is to effectively fuse them and obtain the final clustering assignments. Traditional methods typically employ explicit consistency constraints or post-processing feature fusion. However, these methods either impose strict

Method	Handwrite			BDGP			MRSC			COIL20		
	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
LMVSC	0.671	0.651	0.708	0.556	0.304	0.556	0.352	0.308	0.375	0.692	0.792	0.706
SFMC	0.757	0.868	0.782	0.378	0.352	0.379	0.600	0.603	0.638	0.748	**0.893**	0.788
FPMVS-CAG	0.823	0.792	0.823	0.556	0.376	0.590	0.805	0.688	0.805	0.602	0.780	0.633
FDAGF	0.823	0.794	0.823	0.525	0.368	0.547	0.739	0.657	0.749	0.740	0.837	0.760
AWMVC	0.875	0.793	0.877	0.478	0.297	0.478	0.666	0.556	0.668	**0.772**	0.848	0.774
RCAGL	0.794	0.827	0.840	0.518	0.345	0.526	0.776	0.680	0.776	0.651	0.843	**0.874**
CFMC	0.864	0.837	0.864	0.626	0.428	0.626	0.738	0.651	0.771	0.693	0.815	0.729
ALPC	0.632	0.619	0.702	0.922	0.814	0.922	0.643	0.495	0.643	0.764	0.841	0.779
TLRLF4	0.764	0.919	0.800	0.927	0.799	0.927		N/A		0.763	0.877	0.801
Ours	**0.977**	**0.944**	**0.977**	**0.992**	**0.973**	**0.992**	**0.938**	**0.881**	**0.938**	0.769	0.844	0.790
Method	Caltech5V			MNIST-USPS			ALOI			CCV		
	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
LMVSC	0.461	0.349	0.484	0.560	0.529	0.613	0.603	0.781	0.630	0.143	0.113	0.182
SFMC	0.468	0.550	0.484	0.989	0.971	0.989	0.672	0.757	0.682	0.105	0.006	0.106
FPMVS-CAG	0.809	0.682	0.809	0.783	0.725	0.790	0.348	0.664	0.358	0.222	0.179	0.236
FDAGF	0.813	0.697	0.813	0.561	0.508	0.567	0.662	0.804	0.684	0.221	0.184	0.247
AWMVC	0.785	0.701	0.791	0.587	0.539	0.600	0.670	0.802	0.691	0.218	0.181	0.243
RCAGL	0.836	0.725	0.836	0.609	0.637	0.626	0.398	0.664	0.417	0.209	0.194	0.240
CFMC	0.799	0.693	0.799	0.761	0.719	0.783	0.576	0.794	0.595	0.194	0.168	0.225
ALPC	0.461	0.257	0.473	0.502	0.496	0.520		N/A		0.183	0.127	0.217
TLRLF4	0.755	0.713	0.794	**0.992**	**0.977**	**0.992**	0.737	0.822	0.757	0.214	0.216	0.263
Ours	**0.906**	**0.831**	**0.906**	0.982	0.952	0.982	**0.784**	**0.856**	**0.799**	**0.345**	**0.304**	**0.366**

Table 1: Clustering performance of different traditional methods across various datasets, where N/A indicates program errors or timeouts. It can be observed that our method generally achieves better performance compared to existing methods.

priors or inadequately exploit inter-view relationships, thus limiting applicability to complex MVC scenarios. Alternatively, to avoid reliance on explicit constraints or post-processing feature fusion, we employ a contrastive strategy to extract discriminative consistency information.

To achieve this, we adopt a view-shared label projection head $f_l(\{\mathbf{Z}^v\}_{v=1}^V; \mathbf{W}_l)$ to derive the clustering assignment matrix $\mathbf{P}^v \in \mathbb{R}^{k\times n}$ from $\{\mathbf{Z}^v\}_{v=1}^V$:

$$\mathbf{P}^v = f_l(\mathbf{Z}^v; \mathbf{W}^l). \tag{9}$$

Subsequently, contrastive learning is employed to enhance consistency of $\mathbf{P}^v \in \mathbb{R}^{k\times n}$ across views. The primary goal of contrastive learning is to maximize the similarity between positive pairs (features from the same sample across different views) and minimize the similarity between negative pairs (features from different samples across all views), thus encouraging discriminative and consistent multi-view representations. To this end, we construct positive pairs from features of the same sample across different views, and negative pairs from features of different samples across all views. Formally, for each sample i, pairs $\{(\boldsymbol{p}_i^a, \boldsymbol{p}_i^b) \mid a \neq b\}$ constitute positive samples, whereas pairs $\{(\boldsymbol{p}_i^a, \boldsymbol{p}_j^b) \mid i \neq j\}$ are treated as negative samples. The contrastive loss for $\{\mathbf{P}^v\}_{v=1}^V$ is then formulated as:

$$s(i,v,v') = -\log \frac{\exp\big(d(\boldsymbol{p}_i^v, \boldsymbol{p}_i^{v'})/\tau\big)}{\sum_{j=1,u=1}^{n,V} \mathbf{1}_{(j,u)\neq(i,v)} \exp\big(d(\boldsymbol{p}_i^v, \boldsymbol{p}_j^u)/\tau\big)},$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{V}\sum_{v=1}^{V}\Big(\frac{1}{k}\sum_{i=1}^{k} q_i^v \log q_i^v\Big),$$

$$\mathcal{L}_{\mathbf{P}} = \frac{1}{nV}\sum_{\substack{i=1\\v=1}}^{n,V}\Big(\frac{1}{V-1}\sum_{\substack{v'=1\\v'\neq v}}^{V} s(i,v,v')\Big) + \lambda_r \mathcal{L}_{\text{reg}}, \tag{10}$$

where $d(\boldsymbol{p}_i, \boldsymbol{p}_j) = \frac{\langle \boldsymbol{p}_i, \boldsymbol{p}_j\rangle}{\|\boldsymbol{p}_i\|\|\boldsymbol{p}_j\|}$ denotes the cosine similarity between two clustering assignment vectors. The function $s(i,v,v')$ denotes the contrastive score, and the temperature coefficient τ controls the scale of similarity values (Chen et al. 2020). The empirical cluster-assignment distribution $q_i^v = \frac{1}{n}\sum_{j=1}^{n} p_{ij}^v$, which measures the average fraction of samples assigned to cluster i in view v. The regularization term $\mathcal{L}_{\text{reg}}$ is an entropy-based regularizer, encouraging a balanced use of clusters and thereby preventing degenerate solutions where most samples collapse into only a few clusters. The regularization coefficient λ_r balances the trade-off between the contrastive loss and the regularization term.

Finally, the overall clustering assignment $\mathbf{P}$ is obtained by a weighted sum of the individual view assignments:

$$\mathbf{P} = \sum_{v=1}^{V} \gamma_{\mathbf{P}}^v \mathbf{P}^v, \tag{11}$$

where $0 \leq \gamma_{\mathbf{P}}^v \leq 1$ is the weight for the v-th view, and $\sum_{v=1}^{V} \gamma_{\mathbf{P}}^v = 1$. We simply set the view weights equally, as is commonly done in deep MVC methods. The complete optimization procedure for BONE is outlined in Algorithm 1.

Dataset	MFLVC				CVCL				DCMVC				Ours			
	ACC	NMI	Purity	Params	ACC	NMI	Purity	Params	ACC	NMI	Purity	Params	ACC	NMI	Purity	Params
HandWrite	0.859	0.864	0.859	28M	0.968	0.929	0.968	27M	0.901	0.822	0.901	21M	0.977	0.944	0.977	**55K**
BDGP	0.990	0.967	0.990	11M	0.990	0.967	0.990	9M	0.991	0.971	0.991	8M	0.992	0.973	0.992	**18K**
MRSC	0.795	0.711	0.795	19M	0.910	0.820	0.910	11M	0.586	0.501	0.610	15M	0.938	0.881	0.938	**31K**
COIL20	0.628	0.733	0.647	24M	0.750	0.816	0.767	28M	0.757	0.841	0.782	21M	0.769	0.844	0.790	**227K**
Caltech5V	0.747	0.681	0.747	26M	0.738	0.641	0.753	17M	0.901	0.832	0.901	21M	0.906	0.831	0.906	**116K**
MNIST-USPS	0.996	0.988	0.996	10M	0.997	0.990	0.997	15M	0.990	0.971	0.990	8M	0.982	0.952	0.982	**69K**
ALOI	0.494	0.778	0.510	18M	0.895	0.932	0.903	23M	0.928	0.956	0.932	14M	0.784	0.856	0.799	**397K**
CCV	0.290	0.302	0.330	27M	0.278	0.275	0.322	25M	0.347	0.335	0.396	24M	0.345	0.304	0.366	**863K**

Table 2: Performance of different deep MVC methods across various datasets. It can be observed that our method achieves comparable or even better performance to existing methods, while utilizing only 1% of their parameter complexity.

Algorithm 1: The optimization process of BONE.

Input: Multi-view dataset $\{\mathbf{X}^v\}_{v=1}^V$; Number of clusters k, layers L; Dimension of representation r; Model penalty coefficients α and β; Contrastive loss temperature coefficient τ; Cluster assignment regularization coefficient λ_r; Batch size b; Number of training epochs E.

Initialize $\{\mathbf{A}_0^v\}_{v=1}^V$ by solving (7).
for $e = 1$ to E **do**
  for $l = 1$ to L **do**
    Solve for semantic features $\{\mathbf{Z}^{v,l}\}_{v=1}^V$ using (4).
    Update the base matrix $\{\mathbf{A}^{v,l}\}_{v=1}^V$ using (5).
    Extract high-level semantic features using (6).
  end for
  Solve for the assignments $\{\mathbf{P}^v\}_{v=1}^V$ using (9).
  Compute the loss using (10) and backpropagate.
end for
Obtain the final cluster assignments from (11).

Output: Cluster assignments $\mathbf{P}$.

Parameter Complexity

In our proposed method, the parameters to be optimized mainly consist of two components: the view-specific feature extraction modules $\{\{g_e^{v,l}\}_{l=1}^L, \mathbf{A}_0^v\}_{v=1}^V$ and the label projection head f_l. Specifically, the label projection head is a single fully connected layer structured as $r \rightarrow k$. Consequently, the total number of parameters is $r(VLr + \sum_{v=1}^V m_v + k)$. Given that r is typically set as a small integer (commonly a few times the number of clusters k), the overall parameter complexity of BONE remains substantially lower compared to existing deep MVC methods.

Experiments

This section evaluates the effectiveness of the proposed method through clustering experiments conducted on several common datasets, including BDGP (Liu et al. 2024b), HandWrite (Chen et al. 2024), MNIST-USPS, COIL20, Caltech5V (Xu et al. 2022), MSRC (Winn and Jojic 2005), ALOI (Zhang, Huang, and Wang 2023), CCV (Jiang et al. 2011), YouTubeFace50 (Wan et al. 2023). All experiments are performed on a Windows 11 laptop equipped with an AMD Ryzen 7 7840HS CPU and 64GB of memory.

Metric	MFLVC	CVCL	DCMVC	Ours
ACC	0.5672	0.705	0.7093	0.7001
NMI	0.7235	0.8289	0.8366	0.8426
Purity	0.4655	0.7431	0.7405	0.7544
Time Cost	83+3009s	582+374s	424+7498s	7+80s
Parameters	20M	25M	16M	156K

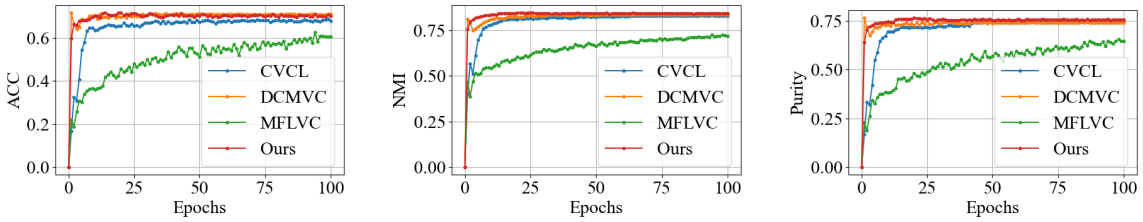
Table 3: Clustering performance of various deep MVC methods on the YouTubeFace50 dataset. Our method demonstrates comparable performance to existing methods while significantly reducing time expenditure.

Experimental Setup

For our method, the number of layers L is set between 2 and 5. The setting of r follows the strategy commonly used in traditional methods, with values set as $k\times$ [1, 2, 3, 4, 5]. The batch size b is set to values in [128, 256, 512]. We perform grid search to determine the optimal values for these three parameters. All remaining parameters were fixed consistently across datasets: regularization coefficients $\alpha = 0.1$ and $\beta = 0.1$, contrastive loss temperature parameter $\tau = 1$, clustering assignment regularization coefficient $\lambda_r = 1$, number of training epochs $E = 100$, and maximum iterations for the initialization matrix optimization set to 100. To validate the proposed method, we compare it with classical and deep MVC methods, including: LMVSC (Kang et al. 2020), SFMC (Li et al. 2020), FPMVS-CAG (Wang et al. 2021b), FDAGF (Zhang et al. 2023), AWMVC (Wan et al. 2023), RCAGL (Liu et al. 2024b), CFMC (Chen et al. 2024), ALPC (Chen et al. 2025), TLRLF4 (Long et al. 2025), MFLVC (Xu et al. 2022), CVCL (Chen et al. 2023) and DCMVC (Cui et al. 2024). For fair comparison, the hyperparameters of these methods are optimized via grid search as per the authors' recommendations. The clustering performance of all methods is evaluated using three commonly used metrics in MVC: Accuracy (ACC), Normalized Mutual Information (NMI), and Purity. Higher values of these metrics correspond to better clustering performance.

Performance Evaluation

In Table 1, we present the performance of the proposed method compared with competitive traditional methods across different datasets, with the best and second-best re-



(a) ACC (b) NMI (c) Purity

Figure 3: Detailed training processes of different deep methods on the YouTubeFace50 dataset, where all methods except ours are pre-trained for 200 epochs. It can be observed that BONE can quickly achieve clustering performance comparable to existing deep methods, without the need for any pre-training procedure.

sults highlighted in bold and underlined, respectively. It can be observed that our method outperforms existing methods on the majority of datasets, while showing comparable results on the remaining datasets, even without the introduction of explicit consistency or additional constraint assumptions. This improvement is attributed to the feature extraction capabilities of the neural network, which alleviates the representational limitations of classical methods by extracting high-level semantic representation from representations obtained by optimization. Our method does not rely on explicit consistency assumptions or post-processing feature fusion to obtain view consistency information, making it more robust to complex MVC scenarios characterized by large modality discrepancies or the presence of noise.

More importantly, the proposed method does not rely on a decoupled post-processing clustering process. Instead, it employs a contrastive strategy that forces the consistency feature fusion module to extract a consistent clustering assignment matrix from the multi-view high-dimensional semantic information, directly yielding the final clustering assignment. This end-to-end approach avoids the performance degradation typically caused by the decoupling of consistency learning and final clustering assignment, while also eliminating the computational overhead associated with additional post-processing clustering algorithms, making it well-suited for large-scale MVC tasks.

Furthermore, Table 2 compares the performance of the proposed method with representative deep MVC methods. Even without pre-training and with only approximately 1% of the parameter complexity, our method achieves performance that is comparable to or even surpasses existing methods. This is because our method does not solely rely on a data-driven approach to implicitly learn latent representations from the data. Instead, it incorporates model priors from classical methods to obtain semantic features. This strategy not only significantly reduces parameter complexity but also eliminates the need for pre-training, thereby greatly simplifying the model's training cost.

Training Efficiency

To provide a more intuitive observation of the training process, we evaluate the performance of various deep methods on YouTubeFace50 dataset (including 126,054 samples). Due to the significant computational cost of full training, we restrict all methods to a maximum of 10 batches per epoch and train for 100 epochs. All deep methods except ours are pre-trained for 200 epochs. For BONE, 1% of the samples are randomly sampled for solving the initial matrix. Table 3 presents detailed performances for each method. The time complexity of BONE is composed of the initialization of $\{\mathbf{A}_0^v\}_{v=1}^V$ and contrastive training, while the other methods include both pre-training and contrastive training.

As shown by the results, BONE achieves satisfactory performance with significantly reduced computational cost, without the need for any pre-training process. In contrast, the other methods require more training epochs to capture consistency information, leading to higher computational costs. Figure 3 illustrates the changes in clustering metrics during the training process for each method. Even without any pre-training process, our approach yields satisfactory clustering performance in the early epochs of training.

Conclusion

This paper introduces BONE, a lightweight and interpretable framework for MVC. By bridging optimization and neural networks, BONE extracts high-level features from low-level features derived through the classical optimization process, thereby integrating consistency across views without relying on explicit prior constraints. The experimental results demonstrate that BONE efficiently integrates view consistency information without the need for explicit prior constraints or post-processing feature fusion, achieving satisfactory performance compared to existing competitive methods. This work provides a novel perspective for designing efficient MVC algorithms, showcasing significant potential for practical applications.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62173091, in part by the Natural Science Foundation of Fujian Province, China under Grant 2024J0902, in part by the Qishan Scholar Program of Fuzhou University under Grant XRC-22014, and in part by the Postgraduate Education Teaching Reform Project of Fuzhou University under Grant FYAI2025011.

References

Cai, Y.; Che, H.; Pan, B.; Leung, M.-F.; Liu, C.; and Wen, S. 2024. Projected cross-view learning for unbalanced incomplete multi-view clustering. *Information Fusion*, 105: 102245.

Chao, G.; Jiang, Y.; and Chu, D. 2024. Incomplete contrastive multi-view clustering with high-confidence guiding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11221–11229.

Chen, J.; Mao, H.; Woo, W. L.; and Peng, X. 2023. Deep multiview clustering by contrasting cluster assignments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16752–16761.

Chen, M.-S.; Wang, C.-D.; Huang, D.; Lai, J.-H.; and Yu, P. S. 2024. Concept factorization based multiview clustering for large-scale data. *IEEE Transactions on Knowledge and Data Engineering*, 36(11): 5784–5796.

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

Chen, Y.; Wang, H.; Peng, J.; and Wang, Y. 2025. Anchor learning with potential cluster constraints for multiview clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 15939–15947.

Cui, J.; Li, Y.; Huang, H.; and Wen, J. 2024. Dual contrast-driven deep multi-view clustering. *IEEE Transactions on Image Processing*.

Du, S.; Wu, C.; Fang, Z.; Zhao, W.; Wu, Y.; Wang, C.; and Wang, S. 2025. LargeMvC-Net: anchor-based deep unfolding network for large-scale multi-view clustering. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 1714–1723.

Fang, U.; Li, M.; Li, J.; Gao, L.; Jia, T.; and Zhang, Y. 2023. A comprehensive survey on multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(12): 12350–12368.

Gower, J. C.; and Dijksterhuis, G. B. 2004. *Procrustes Problems*, volume 30. OUP Oxford.

Guo, W.; Che, H.; Leung, M.-F.; Jin, L.; and Wen, S. 2025. Robust mixed-order graph learning for incomplete multiview clustering. *Information Fusion*, 115: 102776.

Hu, H.; Wang, X.; Zhang, Y.; Chen, Q.; and Guan, Q. 2024. A comprehensive survey on contrastive learning. *Neurocomputing*, 610: 128645.

Ji, J.; and Feng, S. 2025. Anchors crash tensor: efficient and scalable tensorial multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jiang, Y.-G.; Ye, G.; Chang, S.-F.; Ellis, D.; and Loui, A. C. 2011. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proceedings of the 1st ACM international conference on multimedia retrieval*, 1–8.

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

Kong, J.; Liu, J.; Shang, R.; Zhang, W.; Xu, S.; and Li, Y. 2025. Robust multi-view subspace clustering via neighbor embedding on manifold and low-rank representation learning. *Expert Systems with Applications*, 267: 125831.

Li, C.; Zhang, B.; Hong, D.; Yao, J.; Jia, X.; Plaza, A.; and Chanussot, J. 2024a. Interpretable networks for hyperspectral anomaly detection: A deep unfolding solution. *IEEE Transactions on Geoscience and Remote Sensing*.

Li, L.; Pan, Y.; Liu, J.; Liu, Y.; Liu, X.; Li, K.; Tsang, I. W.; and Li, K. 2024b. BGAE: Auto-encoding multi-view bipartite graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 36(8): 3682–3696.

Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2020. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1): 330–344.

Liang, J.; Dong, X.; Wang, P.; Xu, J.; Wu, D.; and Nie, F. 2025. Multi-view graph clustering via dual view-cluster-order interactivity mining. *IEEE Transactions on Circuits and Systems for Video Technology*.

Lin, R.; Li, J.; Du, S.; Wang, S.; and Zhang, L. 2025. OIMGC-Net: optimization-inspired interpretable multi-view graph clustering network. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 1957–1966.

Liu, J.; Liu, X.; Yang, Y.; Liao, Q.; and Xia, Y. 2023. Contrastive multi-view kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 9552–9566.

Liu, S.; Liang, K.; Dong, Z.; Wang, S.; Yang, X.; Zhou, S.; Zhu, E.; and Liu, X. 2024a. Learn from view correlation: An anchor enhancement strategy for multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26151–26161.

Liu, S.; Liao, Q.; Wang, S.; Liu, X.; and Zhu, E. 2024b. Robust and consistent anchor graph learning for multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 36(8): 4207–4219.

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

Liu, X.; Liu, L.; Liao, Q.; Wang, S.; Zhang, Y.; Tu, W.; Tang, C.; Liu, J.; and Zhu, E. 2021. One pass late fusion multi-view clustering. In *International Conference on Machine Learning*, 6850–6859. PMLR.

Long, Z.; Wang, Q.; Ren, Y.; Liu, Y.; and Zhu, C. 2025. TL-RLF4MVC: tensor low-rank and low-frequency for scalable multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 14193–14201.

Su, P.; Liu, Y.; Li, S.; Huang, S.; and Lv, J. 2024. Robust contrastive multi-view kernel clustering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 4938–4945.

Sun, Y.; Qin, Y.; Li, Y.; Peng, D.; Peng, X.; and Hu, P. 2024. Robust multi-view clustering with noisy correspondence. *IEEE Transactions on Knowledge and Data Engineering*.

Tang, J.; Yi, Q.; Fu, S.; and Tian, Y. 2024. Incomplete multi-view learning: Review, analysis, and prospects. *Applied Soft Computing*, 153: 111278.

Trosten, D. J.; Lokse, S.; Jenssen, R.; and Kampffmeyer, M. 2021. Reconsidering representation alignment for multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1255–1265.

Trosten, D. J.; Løkse, S.; Jenssen, R.; and Kampffmeyer, M. C. 2023. On the effects of self-supervision and contrastive alignment in deep multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23976–23985.

Wan, X.; Liu, J.; Gan, X.; Liu, X.; Wang, S.; Wen, Y.; Wan, T.; and Zhu, E. 2024. One-step multi-view clustering with diverse representation. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3): 5774–5786.

Wan, X.; Liu, X.; Liu, J.; Wang, S.; Wen, Y.; Liang, W.; Zhu, E.; Liu, Z.; and Zhou, L. 2023. Auto-weighted multi-view clustering for large-scale data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10078–10086.

Wang, B.; Zeng, C.; Chen, M.; and Li, X. 2025a. Towards learnable anchor for deep multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 21044–21052.

Wang, J.; Feng, S.; Lyu, G.; and Yuan, J. 2024. Surer: Structure-adaptive unified graph neural network for multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15520–15527.

Wang, Q.; Zhang, Z.; Feng, W.; Tao, Z.; and Gao, Q. 2025b. Contrastive multi-view subspace clustering via tensor transformers autoencoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 21207–21215.

Wang, S.; Chen, Z.; Du, S.; and Lin, Z. 2021a. Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5042–5055.

Wang, S.; Liu, X.; Liu, S.; Jin, J.; Tu, W.; Zhu, X.; and Zhu, E. 2022a. Align then fusion: Generalized large-scale multi-view clustering with anchor matching correspondences. *Advances in Neural Information Processing Systems*, 35: 5882–5895.

Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; and Zhu, E. 2021b. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31: 556–568.

Wang, Y.; Chang, D.; Fu, Z.; Wen, J.; and Zhao, Y. 2022b. Graph contrastive partial multi-view clustering. *IEEE Transactions on Multimedia*, 25: 6551–6562.

Winn, J.; and Jojic, N. 2005. Locus: Learning object classes with unsupervised segmentation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, 756–763. IEEE.

Wu, F.; Zhang, T.; Li, L.; Huang, Y.; and Peng, Z. 2024. RPCANet: Deep unfolding RPCA based infrared small target detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4809–4818.

Xu, J.; Ren, Y.; Tang, H.; Pu, X.; Zhu, X.; Zeng, M.; and He, L. 2021. Multi-VAE: Learning disentangled view-common and view-peculiar visual representations for multi-view clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9234–9243.

Xu, J.; Tang, H.; Ren, Y.; Peng, L.; Zhu, X.; and He, L. 2022. Multi-level feature learning for contrastive multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16051–16060.

Yan, J.; Zhang, K.; Sun, Q.; Ge, C.; Wan, W.; Sun, J.; and Zhang, H. 2025. Spatial–spectral unfolding network with mutual guidance for multispectral and hyperspectral image fusion. *Pattern Recognition*, 161: 111277.

Yu, S.; Wang, S.; Dong, Z.; Tu, W.; Liu, S.; Lv, Z.; Li, P.; Wang, M.; and Zhu, E. 2024. A non-parametric graph clustering framework for multi-view data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16558–16567.

Yu, Z.; Dong, Z.; Yu, C.; Yang, K.; Fan, Z.; and Chen, C. P. 2025. A review on multi-view learning. *Frontiers of Computer Science*, 19(7): 197334.

Zhang, C.; Jia, X.; Li, Z.; Chen, C.; and Li, H. 2024. Learning cluster-wise anchors for multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16696–16704.

Zhang, G.-Y.; Huang, D.; and Wang, C.-D. 2023. Facilitated low-rank multi-view subspace clustering. *Knowledge-Based Systems*, 260: 110141.

Zhang, P.; Wang, S.; Li, L.; Zhang, C.; Liu, X.; Zhu, E.; Liu, Z.; Zhou, L.; and Luo, L. 2023. Let the data choose: Flexible and diverse anchor graph fusion for scalable multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11262–11269.

Zhao, X.; Wang, S.; Liu, X.; and Liang, J. 2024a. Multi-view clustering via dynamic unified bipartite graph learning. *Pattern Recognition*, 156: 110715.

Zhao, Z.; Nie, F.; Wang, R.; Wang, Z.; and Li, X. 2024b. An balanced, and scalable graph-based multiview clustering method. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, L.; Du, G.; Lue, 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.