Bayesian Debiasing for Robust Contrastive Multi-view Clustering

1st Zhehan Zhou
School of Computer Science
South China Normal University
Guangzhou, China
leslielalala12@gmail.com

2nd Zhonghua Li
School of Computer Science
South China Normal University
Guangzhou, China
Atonioy@163.com

3rd Na Tang
School of Computer Science
South China Normal University
Guangzhou, China
tangna@scnu.edu.cn

4th Xiaolin Xiao
School of Computer Science
South China Normal University
Guangzhou, China
shellyxiaolin@gmail.com*

***Abstract*—Contrastive Multi-view Clustering (MvC) learns discriminative representations by distinguishing positive and negative pairs, but its performance often suffers from false positives and negatives. Existing approaches mitigate this issue with k-means pseudo-labels, yet these deterministic labels are prone to misclassifying hard samples near cluster boundaries. To address this issue, we propose a Bayesian Debiasing framework for robust contrastive multi-view clustering (BDMvC). First, BDMvC employs a Bayesian Gaussian mixture model to generate pseudo-labels, enabling soft assignments of hard samples. Second, two mixing strategies are designed to synthesize diverse hard samples, ensuring smoother and more robust cluster boundaries. Finally, unlike prior methods, BDMvC provides the first unified treatment of both hard positives and negatives within a principled framework, substantially enhancing the robustness of contrastive learning. Experiments on benchmark datasets demonstrate that BDMvC consistently outperforms state-of-the-art methods.**

***Index Terms*—Contrastive multi-view clustering, Bayesian pseudo-label, hard sample mixing**

I. Introduction

In many real-world applications, each instance is represented by multi-view data from multiple perspectives or feature spaces. This motivates MvC as an effective paradigm to integrate complementary information for unsupervised representation learning [1], [2]. Traditional MvC methods rely on graph learning, subspace learning, or matrix factorization [3], while recent progress in deep learning has enabled more expressive representations and scalable solutions [4], [5]. Among them, contrastive MvC has gained significant attention because it learns discriminative representations without supervision by aligning positive pairs and separating negative ones [6], showing promising results across diverse domains.

Despite these advances, contrastive MvC still suffers from unreliable contrastive supervision caused by ambiguous samples with uncertain semantic assignments, which are prevalent near cluster boundaries [7], [8]. Such samples, commonly referred to as hard samples, tend to induce false positives and false negatives, severely degrading the quality of contrastive pairs. To mitigate this issue, pseudo-labels are often introduced to debias contrastive learning [9], [10]. However, most existing methods rely on deterministic pseudo-labels generated by k-means [11], [12], which are particularly unreliable in the early training stage and struggle to model the inherent uncertainty of hard samples. Furthermore, although mining hard samples has been shown to enhance discriminability [13], [14], existing approaches typically select a fixed set of hard samples from predefined positives or negatives, resulting in limited diversity, nonsmooth cluster boundaries, and prohibitive computational cost when scaling up [15], [16].

To overcome these limitations, we propose a novel BDMvC framework, as shown in Fig. 1. The key contributions of this work are as follows. First, we introduce Bayesian debiasing by employing a Bayesian Gaussian mixture (BGM) model to generate Bayesian pseudo-labels, providing soft cluster assignments that more reliably identify hard samples near cluster boundaries. Second, we design two hard sample mixing strategies to synthesize diverse hard positives and negatives, improving robustness and producing smoother cluster boundaries. Third, unlike prior approaches that handle hard positives and negatives separately, our framework offers a unified contrastive learning approach, enhancing robustness and stability in MvC. In addition, extensive experiments on six benchmark datasets validate the superiority and robustness of BDMvC, showing consistent improvements over state-of-the-art contrastive MvC methods.

II. Method

Our BDMvC framework first integrates probabilistic debiasing to remove false pairs and identify hard samples. It then performs hard sample mixing and is optimized via a unified instance- and cluster-level contrastive loss.

Suppose a multi-view dataset with N samples across V views, is denoted as $\{X^v = [x_1^v, \cdots, x_N^v] \in \mathbb{R}^{D_v \times N}\}$, where D_v is the dimension of the v-th view.

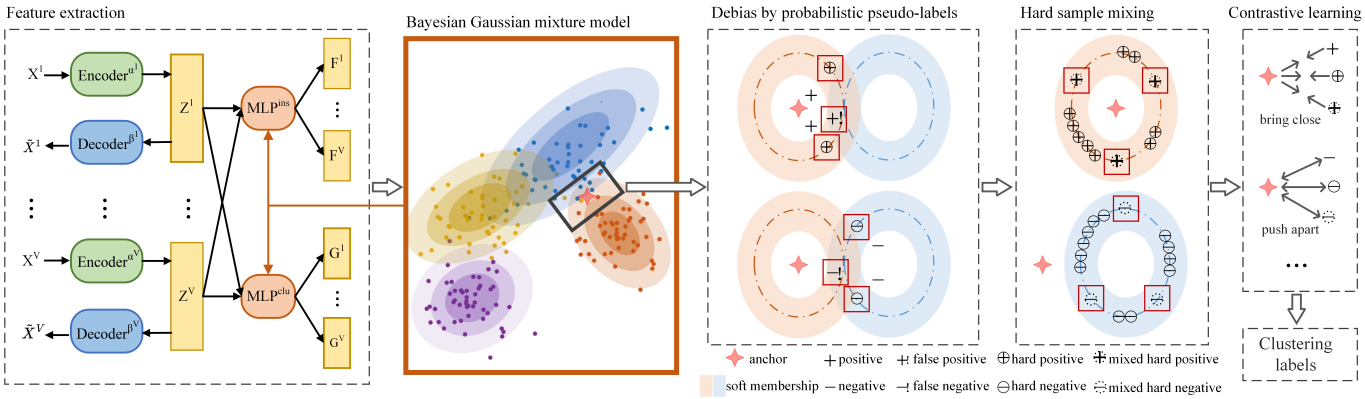

Fig. 1: Overview of BDMvC. The Bayesian Gaussian mixture model generates probabilistic pseudo-labels, which form the basis of our debiasing strategy and help in learning high-level features. Our debiasing strategy involves false sample removal and hard sample identification. Afterward, hard sample mixing is applied to enhance the diversity of contrastive sample pairs. Finally, contrastive learning is performed on the reliable and diverse sample pairs to derive the clustering results.

A. Bayesian Debiasing

1) False Sample Removal: To mitigate bias from false positives and negatives, we employ a Bayesian Gaussian mixture (BGM) model to assign probabilistic cluster memberships. For sample x_i^v in view v, its posterior probability of belonging to cluster k is:

$$\gamma_{ik}^v = \frac{\pi_k \mathcal{N}(x_i^v|\mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_i^v|\mu_j, \Sigma_j)}, \tag{1}$$

where K is the number of components, and π_k, μ_k, Σ_k are the mixture weight, mean, and covariance. The parameters are updated iteratively by maximizing the posterior. The joint probability that x_i^v and x_j^u belong to the same cluster is:

$$\Gamma_{ij}^{vu} = \sum_{k=1}^{K} \gamma_{ik}^v \gamma_{jk}^u. \tag{2}$$

Initial positive pairs with low Γ_{ij}^{vu} are discarded as false positives, while initial negative pairs with high Γ_{ij}^{vu} are removed as false negatives.

2) Hard Sample Identification: For an anchor x_i^v, we compute the similarity set $\mathcal{S}_i = \{s_{ij}\}_{j=1}^N$, where

$$s_{ij} = \frac{(x_i^v)^\top x_j^v}{\|x_i^v\| \, \|x_j^v\|}. \tag{3}$$

Hard positive candidates are selected from low-similarity samples while considering probabilistic pseudo-labels to exclude false negatives:

$$\mathcal{H}_i^+ = \underset{\mathcal{C}\subset\mathcal{S}_i, |\mathcal{C}|=n}{\arg\min} \sum_{s_{ij}\in\mathcal{C}} \left(s_{ij} - \delta\, \Gamma_{ij}^{vu}\right), \tag{4}$$

where $\mathcal{C}$ is a subset of $\mathcal{S}_i$ consisting of n similarity scores, and minimizing the objective selects the n least similar samples to the anchor. The objective jointly accounts for feature-level hardness and semantic consistency. Specifically, minimizing s_{ij} encourages selecting samples that are less similar to the anchor, while the term Γ_{ij}^{vu} measures the joint posterior probability that x_i^v and x_j^u belong to the same cluster under the Bayesian Gaussian mixture model. The balancing coefficient δ controls the trade-off between selecting hard positives and suppressing false negatives, ensuring that selected samples are both challenging and semantically reliable.

The final hard positive set for anchor x_i^v is denoted as $\mathcal{P}_i^v$. Hard negatives are selected in the same manner and denoted as $\mathcal{N}_i^v$. By using Bayesian probabilistic assignments rather than deterministic k-means labels, our approach flexibly handles hard samples near cluster boundaries, improving the reliability of contrastive pairs.

B. Hard Sample Mixing

Although hard samples improve cluster boundary learning, fixed hard samples limit diversity and increase computation. To address this, we propose two mixing strategies that generate a small set of synthetic hard samples per iteration, improving generalization.

For anchor x_i^v, let $\mathcal{P}_i^v$ be its hard positives. The first mixing strategy combines two randomly selected hard positives $\tilde{p}_j^v, \tilde{p}_k^v \in \mathcal{P}_i^v$ to generate a synthetic sample:

$$h_r^{(1),v} = \frac{\lambda_r \tilde{p}_j^v + (1-\lambda_r)\tilde{p}_k^v}{\|\lambda_r \tilde{p}_j^v + (1-\lambda_r)\tilde{p}_k^v\|}, \tag{5}$$

where $\lambda_r \in (0,1)$ is random. The resulting set is $\mathcal{H}_i^{(1),v} = \{h_1^{(1),v}, \ldots, h_{T_1}^{(1),v}\}$.

The second strategy mixes the anchor with a single hard positive $\tilde{p}_j^v$:

$$h_r^{(2),v} = \frac{\mu_r x_i^v + (1-\mu_r)\tilde{p}_j^v}{\|\mu_r x_i^v + (1-\mu_r)\tilde{p}_j^v\|}, \tag{6}$$

where $\mu_r \in (0, 0.5)$ ensures the hard sample dominates. The resulting set is $\mathcal{H}_i^{(2),v} = \{h_1^{(2),v}, \ldots, h_{T_2}^{(2),v}\}$.

The synthetic hard positives for x_i^v are merged as

$$\mathcal{SP}_i^v = \mathcal{H}_i^{(1),v} \cup \mathcal{H}_i^{(2),v} = \{sp_j^v\}_{j=1}^{T_p}, \tag{7}$$

where $T_p = T_1 + T_2$ denotes the number of synthetic hard positives.

Similarly, synthetic hard negatives are generated as $\mathcal{SN}_i^v = \{sn_j^v\}_{j=1}^{T_n}$.

C. Unified Contrastive Learning Loss

1) Intra-view Reconstruction: To reduce redundancy and noise, we incorporate an autoencoder for each view to extract view-specific features. Specifically, given anchor x_i^v, we encode it as z_i^v and decode it as $\tilde{x}_i^v$. Let L_{rec} denote the total reconstruction loss across all views:

$$L_{rec} = \sum_{v=1}^{V} \sum_{i=1}^{N} \|x_i^v - \tilde{x}_i^v\|_2^2. \tag{8}$$

This reconstruction ensures that view-specific features retain essential information while filtering out noise.

2) Instance-level Contrastive Loss: To effectively capture cross-view relationship, we stack a shared MLP on $\{Z^v = [z_1^v, \cdots, z_N^v]\}_{v=1}^V$ to obtain instance-level features $\{F^v = [f_1^v, \cdots, f_N^v]\}_{v=1}^V$. Typically,$f_i^v$ possesses $(VN-1)$ feature pairs, i.e., $\{(f_i^v, f_j^u)\}_{j=1,\cdots,N}^{u=1,\cdots,V}$, where the pair (f_i^v, f_i^v) is excluded. Bayesian pseudo-labels p_{ij}^{vw} and synthetic samples are incorporated by assigning probabilistic weights.

For an anchor feature f_i^v, all contrastive candidates are collected into a unified set:

$$\begin{aligned} \mathcal{S}_i^v = &\Big\{(f_j^u, \Gamma_{ij}^{vu})\Big\}_{j,u} \cup \Big\{(sp_r^w, \Gamma_{ir}^{vw})\Big\}_{r=1}^{T_p^f} \\ &\cup \Big\{(sn_r^w, \Gamma_{ir}^{vw})\Big\}_{r=1}^{T_n^f}, \end{aligned} \tag{9}$$

where Γ denotes the joint posterior probability produced by the Bayesian debiasing module, sp_r^w and sn_r^w represent synthetic hard positive and negative cluster samples, respectively.

The instance-level contrastive objective of the v-th view and the u-th view is defined as a probabilistically weighted InfoNCE loss over the augmented sample space:

$$l_f^{vu} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\sum\limits_{(y,\gamma)\in\mathcal{S}_i^v} \gamma\, \delta(y)\, \exp\Big(\frac{\mathrm{sim}(f_i^v, y)}{\tau_f}\Big)}{\sum\limits_{(y,\gamma)\in\mathcal{S}_i^v} \gamma\, \exp\Big(\frac{\mathrm{sim}(f_i^v, y)}{\tau_f}\Big) - e^{1/\tau_f}}, \tag{10}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, τ_f is the temperature parameter, and $\delta(y)$ is an indicator function that activates positive and synthetic positives while suppressing negatives.

Finally, the accumulated instance-level contrastive loss across all views can be expressed as:

$$L_{ins} = \frac{1}{2} \sum_{v=1}^{V} \sum_{u\neq v}^{V} l_f^{vu}. \tag{11}$$

This instance-level design combines Bayesian pseudo-labels with synthetic hard samples, unifying hard pair treatment and improving robustness.

Algorithm 1 BDMvC: Bayesian Debiasing for Robust Contrastive Multi-view Clustering

Input: Multi-view data $\{X^v \in \mathbb{R}^{D_v \times N}\}_{v=1}^V$; Cluster number K.

Output: The clustering result R.

```
Compute the intra-view reconstruction loss.
Compute the probabilistic pseudo-labels using BGM.
for x_i^v, i = 1 to N do
    Detect false samples using probabilistic pseudo-labels by Eq. 2.
    Identify hard samples using probabilistic pseudo-labels and cosine similarity by Eq. 4.
    Mix hard samples by Eq. 5 and Eq. 6.
end for
Calculate instance-level contrastive loss by Eq. 11.
for G_{j,:}^v, j = 1 to K do
    Detect false samples using probabilistic pseudo-labels by Eq. 2.
    Identify hard samples using probabilistic pseudo-labels and cosine similarity by Eq. 4.
    Mix hard samples by Eq. 5 and Eq. 6.
end for
Calculate cluster-level contrastive loss by Eq. 14.
Update model by minimizing the contrastive loss L with the guidance of joint probability Γ_ij^uv.
Obtain R by the accuracy of cluster labels.
return R
```

3) Cluster-level Contrastive Loss: To exploit cluster-level consistency, samples from the same cluster are treated as positives, while samples from different clusters are treated as negatives, guided by Bayesian pseudo-labels. A shared MLP is applied to $\{Z^v\}_{v=1}^V$ to obtain semantic-level assignments $\{G^v \in \mathbb{R}^{K\times N}\}_{v=1}^V$, where $G_{i,:}^v$ denote the i-th row of G^v, representing the i-th cluster in the v-th view, $G_{i,:}^v$ possess $(VK-1)$ sample pairs, i.e., $\{(G_{i,:}^v, G_{j,:}^u)\}_{j=1,\cdots,C}^{u=1,\cdots,V}$, where the pair $(G_{i,:}^v, G_{i,:}^v)$ is excluded. For each anchor cluster representation $G_{i,:}^v$, we construct an augmented contrastive set

$$\begin{aligned} \mathcal{C}_i^v = &\Big\{(G_{j,:}^u, \Gamma_{ij}^{vu})\Big\}_{j,u} \cup \Big\{(sp_r^w, \Gamma_{ir}^{vw})\Big\}_{r=1}^{T_p^g} \\ &\cup \Big\{(sn_r^w, \Gamma_{ir}^{vw})\Big\}_{r=1}^{T_n^g}, \end{aligned} \tag{12}$$

where Γ denotes the joint posterior probability produced by the Bayesian debiasing module, sp_r^w and sn_r^w represent synthetic hard positive and negative cluster samples, respectively.

The cluster-level contrastive objective of the v-th view and the u-th view is defined as a unified probabilistically weighted InfoNCE loss:

$$l_g^{vu} = -\frac{1}{K} \sum_{i=1}^{K} \log \frac{\sum\limits_{(y,\gamma)\in\mathcal{C}_i^v} \gamma\, \delta(y)\, \exp\Big(\frac{\mathrm{sim}(G_{i,:}^v, y)}{\tau_g}\Big)}{\sum\limits_{(y,\gamma)\in\mathcal{C}_i^v} \gamma\, \exp\Big(\frac{\mathrm{sim}(G_{i,:}^v, y)}{\tau_g}\Big) - e^{1/\tau_g}}, \tag{13}$$

TABLE I: The statistics of the multi-view datasets.

Datasets	Samples	Views	Classes
Prokaryotic	551	3	4
Caltech101	2386	6	20
BDGP	2500	2	5
MNIST-USPS	5000	2	10
Hdigit	10000	2	10
Handwritten	2000	6	10

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, τ_g is the temperature parameter, and $\delta(\cdot)$ activates positive and synthetic positive cluster samples.

Finally, the accumulated cluster-level contrastive loss across all views can be expressed as:

$$L_{clu} = \frac{1}{2}\sum_{v=1}^{V}\sum_{u\neq v}^{V} l_g^{vu} + \sum_{v=1}^{V}\sum_{j=1}^{K} w_j^v \log w_j^v, \tag{14}$$

where $w_j^v = \frac{1}{N}\sum_{i=1}^{N}\gamma_{ij}^v$ and γ_{ij}^v denotes the probability that the i-th sample in the v-th view belongs to the j-th cluster. The second term regularizes the contrastive loss to prevent assigning all samples to a single cluster.

This cluster-level design, also combined with Bayesian pseudo-labels and synthetic samples, unifies hard positive and negative treatment and enhances global robustness.

4) Overall Objective: The final loss combines reconstruction, instance-level, and cluster-level objectives:

$$L = L_{rec} + \lambda_{ins}L_{ins} + \lambda_{clu}L_{clu}, \tag{15}$$

where, λ_{ins} and λ_{clu} are the weight hyperparameters of the high-level and the semantic-level contrastive loss, respectively. By minimizing this total loss function, the model can effectively improve the contrastive learning effect.

The detailed process of our proposed BDMvC is summarized in Algorithm 1.

III. Experiments

A. Experiments Setup

1) Datasets: We employ six benchmark multi-view datasets for performance comparison, with the dataset statistics summarized in Table I. Among these, the Handwritten dataset consists of six views, and, following the approach of [10], we created five variations by selecting different numbers of views to evaluate the robustness of different models across varying view counts.

2) Comparison Methods: We compare our BDMvC model with nine state-of-the-art MvC approaches, including four mathematical methods (SMVSC [18], PLCMF [19], FastMICE [20], OMVCDR [21]) and five contrastive learning methods (CoMVC [22], MFLVC [10], CVCL [23], SCM [25], MCoCo [26]).

3) Implementations: The experiments were conducted on a Linux system with an i9-13900K CPU, 128GB RAM, and an NVIDIA 4090Ti GPU. For our model, the hyper-parameters were set as follows: batch size sets as 256, instance-level temperature is 0.5, cluster-level temperature is 1.0, and learning rate is 0.003.

4) Evaluation Metrics: The clustering performance was assessed using three metrics: clustering accuracy (ACC), normalized mutual information (NMI), and purity (PUR). For a fair comparison, all methods are evaluated using the same experimental protocol. Each experiment is repeated multiple times with different random initializations, and the average results are reported.

B. Comparison Results and Analysis

1) Results on General Datasets: The clustering performance of our BDMvC model is compared with nine state-of-the-art methods on five benchmark datasets, as shown in Table II. The key observations are as follows:

Comparison with mathematical methods: BDMvC consistently outperforms traditional mathematical clustering methods such as SMVSC, PLCMF, and FastMICE across all datasets and metrics. These methods learn representations from raw features, which may contain noise or redundancy and do not distinguish easy from hard samples, limiting cluster quality. For instance, on Prokaryotic, BDMvC achieves 0.627 ACC, higher than 0.558 of MFLVC and 0.533 of FastMICE.

Comparison with contrastive MvC methods: BDMvC surpasses existing contrastive multi-view clustering methods on all five datasets. The largest gain is on Prokaryotic, improving ACC by 0.073 over the second-best MCoCo (0.627 vs. 0.554). On BDGP, it attains 0.992 ACC, outperforming CVCL (0.990) and MFLVC (0.980), and on MNIST-USPS and Hdigit, it reaches 0.997 and 0.976 ACC, respectively. While methods like CoMVC, SiMVC, and SCM rely on view fusion, and MFLVC or CVCL employ clustering constraints, they either lose view-specific features or fail to handle false positives/negatives and hard sample diversity. In contrast, BDMvC combines probabilistic pseudo-label debiasing with hard sample mixing, ensuring both reliable and diverse contrastive pairs, and achieves consistently superior performance.

2) Comparison across Different Number of Views: To further evaluate our method, we constructed five new datasets from Handwritten by selecting different numbers of views. The results on these synthetic datasets are presented in Table III. From these results, we can draw the following observations:

Consistent superiority of BDMvC: Our BDMvC model consistently achieves top performance across all Handwritten datasets with varying numbers of views. As views increase from 2V to 6V, ACC rises from 0.848 to 0.934, an improvement of 8.5%, with similar gains in NMI and PUR, showing that BDMvC effectively leverages additional views for better clustering while maintaining robust feature learning.

Performance drop in some baselines: Methods like FastMICE, CVCL, and SCM exhibit fluctuating results as views increase. For example, FastMICE attains 0.868 ACC on

TABLE II: Clustering results of competing methods on five benchmark datasets. Bold values denote the best results and underlined values denote the second-best.

	Prokaryotic			Caltech101			BDGP			MNIST-USPS			Hdigit		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
SMVSC [18]	0.559	0.182	0.572	0.275	0.351	0.340	0.746	0.762	0.751	0.754	0.688	0.754	0.863	0.768	0.863
PLCMF [19]	0.445	0.020	0.568	0.328	0.407	0.328	0.698	0.713	0.700	0.623	0.659	0.667	0.905	0.796	0.905
FastMICE [20]	0.533	0.269	0.650	0.430	0.455	0.414	0.881	0.902	0.899	0.957	0.933	0.957	0.933	0.926	0.942
OMVCDR [21]	0.518	0.373	0.522	0.396	0.371	0.401	0.570	0.356	0.594	0.795	0.812	0.799	0.905	0.874	0.905
CoMVC [22]	0.414	0.188	0.670	0.164	0.256	0.237	0.802	0.670	0.803	0.987	0.976	0.989	0.903	0.871	0.903
MFLVC [10]	0.430	0.221	0.598	0.213	0.286	0.282	0.980	0.948	0.980	0.993	0.982	0.993	0.944	0.875	0.944
CVCL [23]	0.546	0.235	0.596	0.475	0.459	0.426	0.990	0.967	0.990	0.995	0.987	0.995	0.957	0.923	0.957
SCM [25]	0.383	0.269	0.568	0.271	0.427	0.557	0.971	0.903	0.971	0.986	0.963	0.986	0.943	0.920	0.955
MCoCo [26]	0.558	0.276	0.611	0.472	0.460	0.455	0.987	0.959	0.972	0.995	0.986	0.990	0.960	0.932	0.959
BDMvC	**0.627**	**0.377**	**0.729**	**0.507**	**0.466**	**0.618**	**0.992**	**0.968**	**0.992**	**0.997**	**0.990**	**0.997**	**0.976**	**0.937**	**0.976**

TABLE III: Clustering results of competing methods on Handwritten with different views. "-XV" denotes X views.

	Handwritten-2V			Handwritten-3V			Handwritten-4V			Handwritten-5V			Handwritten-6V		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
SMVSC [18]	0.698	0.512	0.698	0.702	0.683	0.702	0.755	0.763	0.783	0.761	0.788	0.794	0.786	0.801	0.799
PLCMF [19]	0.717	0.732	0.744	0.768	0.780	0.674	0.822	0.832	0.822	0.811	0.836	0.819	0.829	0.836	0.831
FastMICE [20]	0.804	0.763	0.809	0.850	0.815	0.868	0.797	0.845	0.829	0.855	0.874	0.862	0.842	0.858	0.852
OMVCDR [21]	0.726	0.711	0.732	0.783	0.796	0.782	0.766	0.803	0.779	0.803	0.821	0.814	0.851	0.897	0.856
CoMVC [22]	0.778	0.761	0.800	0.836	0.812	0.847	0.819	0.805	0.825	0.786	0.809	0.799	0.793	0.811	0.793
MFLVC [10]	0.792	0.760	0.792	0.831	0.802	0.831	0.854	0.858	0.854	0.821	0.815	0.821	0.824	0.819	0.824
CVCL [23]	0.837	0.812	0.838	0.859	0.828	0.862	0.751	0.778	0.783	0.892	0.878	0.906	0.856	0.869	0.856
SCM [25]	0.838	0.820	0.838	0.822	0.761	0.822	0.910	0.832	0.910	0.879	0.786	0.879	0.821	0.748	0.821
MCoCo [26]	0.753	0.766	0.760	0.771	0.781	0.779	0.809	0.841	0.811	0.815	0.803	0.834	0.897	0.863	0.870
BDMvC	**0.848**	**0.844**	**0.848**	**0.871**	**0.866**	**0.871**	**0.917**	**0.868**	**0.917**	**0.925**	**0.906**	**0.925**	**0.934**	**0.900**	**0.934**

Handwritten-3V but falls to 0.852 on 6V; SCM drops from 0.910 ACC on 4V to 0.821 on 6V. This indicates that adding views can introduce noisy or inconsistent information. In contrast, BDMvC consistently improves or sustains high accuracy, demonstrating its robustness and effective multi-view integration.

These results highlight both the effectiveness of our Bayesian debiasing and hard sample mixing, and its scalability in leveraging complementary multi-view information.

C. Model Analysis

1) Visualization of Features during the Training Process: To better understand BDMvC, we visualize learned features on MNIST-USPS and BDGP using t-SNE, as shown in Fig. 2. The left column visualizes the features in the initial state, while the right column shows learned features after training, where clear cluster separation is observed.

2) Ablation Studies: We further conduct ablation studies on representative datasets to evaluate the contribution of each module, as summarized in Table IV. On Prokaryotic, removing BGM, HNM, HPM, or the cluster-level contrastive learning (CL) module reduces ACC to 0.608, 0.554, 0.564, and 0.586, respectively, compared to 0.627 for the full BDMvC. On MNIST-USPS, the corresponding drops are 0.005, 0.094, 0.100, and 0.092. Similarly, on Handwritten-2V, ACC decreases from 0.848 to 0.825 (w/o-BGM), 0.756 (w/o-HNM), 0.733 (w/o-HPM), and 0.738 (w/o-CL). These results demonstrate that all modules contribute meaningfully: BGM provides reliable probabilistic pseudo-labels, HNM and HPM enhance the discriminability by synthesizing hard negatives and positives, and CL improves feature-level contrastive alignment.

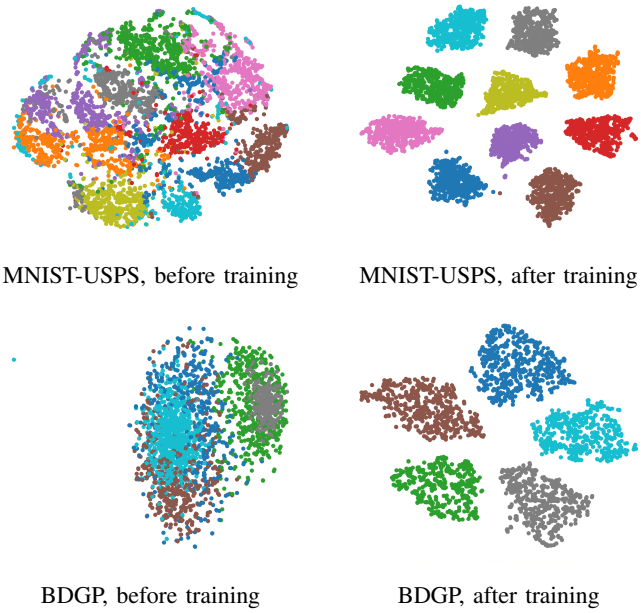

Fig. 2: Visualization of learned features F^v on MNIST-USPS and BDGP.

TABLE IV: Ablation study on representative datasets.

Dataset	Method	ACC	NMI	PUR
Prokaryotic	w/o-BGM	0.608	0.358	0.661
	w/o-HNM	0.554	0.283	0.612
	w/o-HPM	0.564	0.302	0.623
	w/o-CL	0.586	0.314	0.623
	BDMvC	**0.627**	**0.377**	**0.729**
MNIST-USPS	w/o-BGM	0.992	0.990	0.992
	w/o-HNM	0.903	0.955	0.903
	w/o-HPM	0.897	0.958	0.897
	w/o-CL	0.905	0.885	0.906
	BDMvC	**0.997**	**0.990**	**0.997**
Handwritten-2V	w/o-BGM	0.825	0.806	0.825
	w/o-HNM	0.756	0.794	0.765
	w/o-HPM	0.733	0.802	0.743
	w/o-CL	0.738	0.780	0.738
	BDMvC	**0.848**	**0.844**	**0.848**

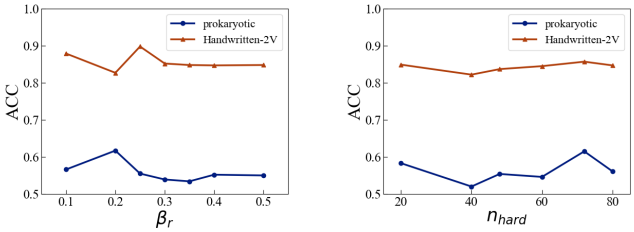

Fig. 3: Parameter sensitivity analysis on ACC with respect to mixing coefficient β_r and number of synthetic hard samples n_{hard}.

The combined effect of all components enables BDMvC to achieve the highest clustering performance consistently.

3) Parameters Sensitivity Analysis: We conduct experiments on two representative datasets, i.e., Prokaryotic and Handwritten-2V, to investigate the performance sensitivity of BDMvC over the mixing coefficient β_r and the total number of synthetic hard samples n_{hard}. Figure 3 shows the accuracy scores of BDMvC as β_r ranges from 0.1 to 0.5 and n_{hard} varies between 20 and 80, respectively. The results indicate that the performance of our method remains relatively stable to β_r and n_{hard} within the recommended ranges.

IV. Conclusion

In this study, we present BDMvC, a simple yet effective framework for multi-view clustering that integrates a Bayesian debiasing strategy to mitigate false positives and negatives, and to identify boundary hard samples by using probabilistic pseudo-labels, along with two mixing strategies that ensure smooth cluster boundaries. This unified framework consistently handles both hard positives and negatives, achieving state-of-the-art performance on multiple benchmarks. Future work will extend BDMvC to address incomplete multi-view scenarios.

References

[1] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Big Data Min. Anal.*, vol. 1, no. 2, pp. 83–107, 2018.

[2] D. Hu, S. Liu, *et al.*, "Reliable attribute-missing multi-view clustering with high-level and feature-level cooperative imputation," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, pp. 1456–1466, 2024.

[3] L. Fu, P. Lin, *et al.*, "An overview of recent multi-view clustering," *Neurocomputing*, vol. 402, pp. 148–161, 2020.

[4] Y. Khalafaoui, B. Matei, N. Grozavu, and M. Lovisetto, "Joint multi-view collaborative clustering," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, pp. 1–7, 2023.

[5] L. Li, Y. He, *et al.*, "An adaptive framework for multi-view clustering leveraging conditional entropy optimization," in *Proc. ICASSP*, pp. 1–5, 2025.

[6] X. Yang, J. Jin, *et al.*, "DealMVC: Dual contrastive calibration for multi-view clustering," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, pp. 337–346, 2023.

[7] C.-Y. Chuang, J. Robinson, *et al.*, "Debiased contrastive learning," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 8765–8775, 2020.

[8] T. Huynh, S. Kornblith, *et al.*, "Boosting contrastive self-supervised learning with false negative cancellation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2022.

[9] T.-S. Chen, W.-C. Hung, *et al.*, "Incremental false negative detection for contrastive learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[10] J. Xu, H. Tang, *et al.*, "Multi-level feature learning for contrastive multi-view clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 16051–16060, 2022.

[11] H. Zhao, X. Yang, *et al.*, "Graph debiased contrastive learning with joint representation clustering," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2021.

[12] J. Wang and S. Feng, "Contrastive and view-interaction structure learning for multi-view clustering," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, pp. 5055–5063, 2024.

[13] Y. Liu, X. Yang, *et al.*, "Hard sample aware network for contrastive deep graph clustering," in *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, vol. 37, no. 7, pp. 8194–8202, 2023.

[14] Y. Cai, Z. Chen, *et al.*, "Hard sample aware robust contrastive learning for multi-view clustering," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pp. 1–5, 2025.

[15] V. Verma, A. Lamb, *et al.*, "Manifold mixup: Better representations by interpolating hidden states," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 6438–6447, 2019.

[16] Y. Kalantidis, M. B. Sariyildiz, *et al.*, "Hard negative mixing for contrastive learning," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 21798–21809, 2020.

[17] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 4412–4419, 2020.

[18] M. Sun, P. Zhang, *et al.*, "Scalable multi-view subspace clustering with unified anchors," in *Proc. ACM Int. Conf. Multimedia (ACM MM)*, pp. 3528–3536, 2021.

[19] D. Wang, S. Han, *et al.*, "Pseudo-label guided collective matrix factorization for multiview clustering," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8681–8691, 2022.

[20] D. Huang, C.-D. Wang, *et al.*, "Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 11388–11402, 2023.

[21] X. Wan, J. Liu, *et al.*, "One-step multi-view clustering with diverse representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, pp. 5774–5786, 2024.

[22] D. J. Trosten, S. Lokse, *et al.*, "Reconsidering representation alignment for multi-view clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1255–1265, 2021.

[23] J. Chen, H. Mao, *et al.*, "Deep Multiview Clustering by Contrasting Cluster Assignments," in *Proc. ICCV*, pp. 16752–16761, 2023.

[24] W. Yan, Y. Zhang, C. Lv, C. Tang, G. Yue, L. Liao, and W. Lin, "GCFAGG: Global and cross-view feature aggregation for multi-view clustering," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19863–19872, 2023.

[25] C. Luo, J. Xu, *et al.*, "Simple contrastive multi-view clustering with data-level fusion," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2024.

[26] Y. Zhou, Q. Zheng, *et al.*, "MCoCo: Multi-level Consistency Collaborative multi-view clustering," *Expert Syst. Appl.*, vol. 238, p. 121976, 2024.