CFDM: Contrastive Fusion and Disambiguation for Multi-View Partial-Label Learning

Qiuru Hai[1], Yongjian Deng[1], Yuena Lin[1,2], Zheng Li[1], Zhen Yang[1*], Gengyu Lyu[1*]

[1]College of Computer Science, Beijing University of Technology
[2]Idealism Beijing Technology Co., Ltd.
haiqiuru@emails.bjut.edu.cn, yjdeng@bjut.edu.cn, yuenalin@126.com, lizhengcn@bjut.edu.cn, yangzhen@bjut.edu.cn, lyugengyu@gmail.com

Abstract

When dealing with multi-view data, the heterogeneity of data attributes across different views often leads to label ambiguity. To effectively address this challenge, this paper designs a Multi-View Partial-Label Learning (MVPLL) framework, where each training instance is described by multiple view features and associated with a set of candidate labels, among which only one is correct. The key to deal with such problem lies in how to effectively fuse multi-view information and accurately disambiguate these ambiguous labels. In this paper, we propose a novel approach named CFDM, which explores the consistency and complementarity of multi-view data by multi-view contrastive fusion and reduces label ambiguity by multi-class contrastive prototype disambiguation. Specifically, we first extract view-specific representations using multiple view-specific autoencoders, and then integrate multi-view information through both inter-view and intra-view contrastive fusion to enhance the distinctiveness of these representations. Afterwards, we utilize these distinctive representations to establish and update prototype vectors for each class within each view. Based on these, we apply contrastive prototype disambiguation to learn global class prototypes and accordingly reduce label ambiguity. In our model, multi-view contrastive fusion and multi-class contrastive prototype disambiguation are conducted mutually to enhance each other within a coherent framework, leading to a more ideal classification performance. Experimental results on multiple datasets have demonstrated that our proposed method is superior to other state-of-the-art methods.

Introduction

With the rapid development of internet technologies, network platforms generate vast amounts of multi-view data. For example, in the task of news webpage classification (Figure 1), the news webpage can be represented by *text*, *images*, and *video*, and may be associated with multiple topic labels (such as *natural disasters*, *weather*, and *environment*). However, only one of these labels can accurately reflect the news webpage. Obviously, it is a significant challenge to accurately identify the correct label due to heterogeneous data from multiple views and ambiguous labels. To this end, we design a new learning framework named Multi-View Partial-Label Learning (MVPLL), where each training instance is described by multiple heterogeneous views and associated with several candidate labels, among which only one is correct. MVPLL provides an effective way to learn from such complex data and predict correct labels for unseen instances.

Figure 1: An Example of multi-view webpage classification.

The key to deal with multi-view partial-label data lies in how to effectively integrate these heterogeneous features while accurately identify the correct labels from ambiguous label sets. An intuitive strategy to formulate MVPLL problem is to utilize either Multi-View Learning (MVL) (Lyu et al. 2024; Liu et al. 2023b; Zhang et al. 2023) or Partial-Label Learning (PLL) (Dong et al. 2023; Xu et al. 2023; Feng et al. 2020; Li et al. 2023). MVL provides an effective framework to integrate multi-view heterogeneous information and directly induce the final prediction model. However, such a unique feature-fusion operation cannot precisely eliminate the ambiguity in the candidate labels, inevitably leading to inaccurate identification of the true labels. PLL focuses on identifying the unique correct label from the candidate label set, but it ignores the exploration of potential multi-view consistency and complementarity information, which also significantly damages the classification performance of the learning model. Recently, some studies (Zhao et al. 2022; Sun, Yu, and Tian 2023) attempt to handle these two challenges simultaneously, however, they can only handle dual-view scenarios and hardly be extended to more

*Corresponding author.

views, which restricts their practicalities in real-world applications. Meanwhile, they don't establish direct and compact relationships between multi-view features and ambiguous labels, which naturally leads the label disambiguation performance to be sub-optimal and finally decreases the reliability of the final prediction model.

To address the above issue, in this paper, we design a general MVPLL framework and accordingly propose a novel approach named CFDM, which explores the consistency and complementarity of multi-view data by multi-view contrastive fusion and addresses label ambiguity by multi-class contrastive prototype disambiguation. Specifically, we first utilize multiple view-specific autoencoders to project the raw multi-view data into low-level representations, which are further respectively processed through two view-shared MLP layers to obtain high-level representations. Then, we integrate these high-level representations using both inter-view and intra-view contrastive learning to enhance their distinctiveness in the embedding space, where the outputs of the classifier are employed to generate positive pairs for intra-view contrastive comparison. Afterward, we establish and dynamically update prototype vectors for each class within each view according to these distinctive representations. Finally, we employ a contrastive prototype disambiguation method to learn global class prototypes, thereby reducing label ambiguity. In our model, multi-view contrastive fusion and multi-class contrastive prototype disambiguation are conducted mutually to enhance each other within a coherent framework, which leads to a more ideal classification performance. In summary, the contributions of our paper lie in the following aspects:

- We design a general MVPLL framework and propose a novel CFDM method, which integrates multi-view heterogeneous information through a contrastive fusion mechanism and reduces label ambiguity via a contrastive prototype disambiguation module.
- CFDM not only explores the comprehensive consistency and complementarity across different views, but also establishes the compact relationships between multi-view features and ambiguous labels for facilitating label disambiguation, which significantly enhances the classification performance of the learning model.
- Extensive experimental results and comprehensive experimental analysis have demonstrated that our proposed CFDM method performs significant superiorities against other existing state-of-the-art approaches.

Related Work

Partial-Label Learning (PLL)

Partial-Label Learning focuses on learning from instances with multiple ambiguous candidate labels, where the key to solve the problem lies in how to effectively conduct label disambiguation. To this end, researchers propose various disambiguation strategies, primarily divided into average-based strategy and identification-based strategy. Average-based strategy (Zhang and Yu 2015; Lv et al. 2023) assigns equal weight to the candidate labels and makes predictions by averaging their modeling outputs; whereas identification-based strategy (Chai, Tsang, and Chen 2019; Wang and Zhang 2022) treats the true label as an implicit variable and achieves disambiguation by iteratively updating label confidence and model parameters. Recently, deep learning techniques are applied to PLL (Guo et al. 2023; Cao et al. 2023), leading to the development of deep partial label learning. This enables models to more accurately identify true labels through learning deeper feature representations, significantly enhancing the ability to handle complex data (Wang et al. 2021; Xia et al. 2023).

Multi-View Learning (MVL)

Multi-View Learning aims to learn from instances with heterogeneous features, where the key to solve the problem lies in how to effectively learn complementary and consistent information from multiple views. Existing MVL methods can be roughly divided into the following categories: Co-training methods (Du et al. 2021; Appice and Malerba 2015) alternate training across different views to promote consistency between them; Multi-kernel methods (Li et al. 2022) jointly optimize a set of preset kernels and try to produce a consistent optimal kernel; Multi-view matrix factorization methods (Zhang et al. 2021; Luong et al. 2022) reduce data dimensions and extract a common low-dimensional representation from different views; Graph-based methods (Gu et al. 2023; Zhong, Lyu, and Yang 2024) construct graph structures to integrate information from multi-view data. Besides, there are multi-view subspace clustering methods (Chen et al. 2024; Liu et al. 2022) and deep multi-view methods (Wen et al. 2024; Liu et al. 2024; Wang et al. 2023; Lin et al. 2022).

Multi-View Partial-Label Learning (MVPLL)

Multi-View Partial-Label Learning can be regarded as an integration of PLL and MVL, which aims to learn from training data with diverse representations and label ambiguity. To learn from such complicated data, (Zhao et al. 2022) proposes an MVPLL method that employs a large-margin-based learning strategy and integrates complementary and consensus information across different views to train the MVPLL classifier. (Sun, Yu, and Tian 2023) adopts deep neural networks for multi-view information fusion, and utilizes class prototypes to enhance the model's discriminative capabilities. However, these methods are primarily designed for dual-view scenarios and thus are difficult to extend to more view situations, which limits their applicability in real-world applications. Additionally, they don't establish a direct and compact connection between multi-view features and ambiguous labels, which results in sub-optimal label disambiguation performance and consequently affects the final classification performance of the prediction models.

Methodology

Preliminary

Notations. Formally, we denote $\mathcal{X} = \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \ldots \times \mathbb{R}^{d_V} = \{\mathbf{X}^v\}_{v=1}^{V}$ as the feature space with V views and $\mathcal{Y} = \{c_1, c_2, \ldots, c_q\}$ as the label space with q class labels, where

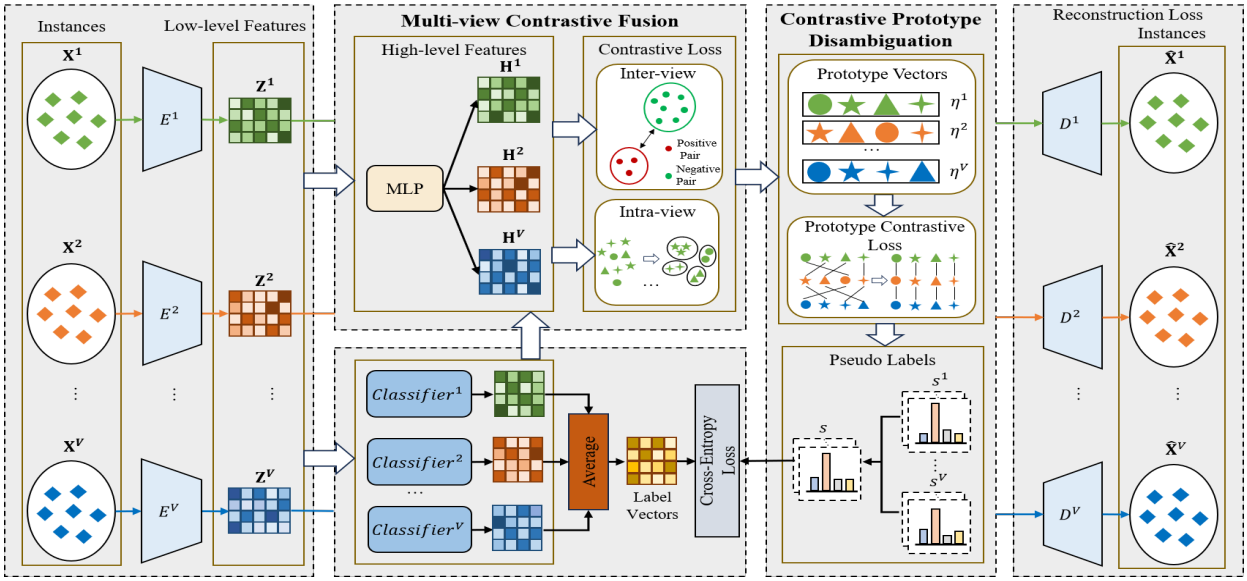

Figure 2: The framework of our proposed CFDM, which consists of two main components: (1) *Multi-view Contrastive Fusion*, we employ inter-view and intra-view contrastive learning to explore the consistency and complementarity of multi-view data, thereby enhancing the distinctiveness of high-level features, where the outputs of classifier are employed to generate positive pairs for intra-view contrastive comparison; (2) *Multi-class Contrastive Prototype Disambiguation*, we update prototype vectors for each class based on distinctive high-level features and employ contrastive prototype disambiguation to ensure consistency of class representations across different views and gradually reduce label ambiguity.

$d_v (1 \leq v \leq V)$ is the feature dimension of the v-th view. Given the MVPLL training dataset $\mathcal{D} = \{(\mathbf{x}_i, S_i) \mid 1 \leq i \leq N\}$ with N instances, where $\mathbf{x}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, ..., \mathbf{x}_i^V] \in \mathcal{X}$ denotes the i-th training instance and $S_i \subseteq \mathcal{Y}$ denotes the candidate label set associated with $\mathbf{x}_i$. The key assumption of MVPLL is that the ground-truth label $\mathbf{y}_i \in \{0,1\}^{q \times 1}$ is always concealed in its candidate label set and not accessible to the model during the whole training process. MVPLL aims to learn a desired multi-class classifier $f : \mathcal{X} \to \mathcal{Y}$ from $\mathcal{D}$ and predict the correct label for unseen instances.

Classifier Induction. The key of our model is to induce a desired multi-class classifier for prediction. Specifically, we assign each instance $\mathbf{x}_i$ with a pseudo label vector $\mathbf{s}_i \in [0,1]^q$, whose element s_{ij} indicates the probability of label j being the ground-truth label of $\mathbf{x}_i$, and the total probability of 1 is allocated among the candidate labels in S_i. Ideally, as the training progresses, $\mathbf{s}_i$ should allocate more probability to the (unknown) correct label. During the whole training process, s_{ij} is updated by the contrastive prototype disambiguation module in the following subsection, and performs as a supervised signal to guide the training of the objective classifier. In our model, we use the cross-entropy loss ℓ_{CE} to compute the classification loss:

$$\begin{aligned} \mathcal{L}_{cls} &= \sum_{j=1}^{q} -s_{ij} \log\left(p_j(\mathbf{x}_i)\right), \\ \text{s.t.} &\sum_{j \in S_i} s_{ij} = 1 \text{ and } s_{ij} = 0, \forall j \notin S_i, \end{aligned} \tag{1}$$

where $p_j(\mathbf{x}_i)$ is the prediction probability of the classifier, which is calculated by averaging the outputs of V different views, *i.e.*, $p_j(\mathbf{x}_i) = \frac{1}{V}\sum_{v=1}^{V} p_j(\mathbf{x}_i^v)$.

View-Specific Autoencoder Network. Multi-view data often contains feature redundancy and random noise, which significantly affects the reliability of the learned classifiers. To address this issue, motivated by (Lin et al. 2022), we apply multiple view-specific autoencoder networks to purify the view features and extract the corresponding view-specific latent representations. Specifically, for v-th view, we denote $E^v(\mathbf{X}^v; \theta^v)$ and $D^v(\mathbf{Z}^v; \rho^v)$ as the encoder and the decoder respectively, where θ^v and ρ^v are network parameters. Meanwhile, denote $\mathbf{z}_i^v = E^v(\mathbf{x}_i^v; \theta^v) \in \mathbb{R}^{l_v}$ as l_v-dimensional latent representation of i-th instance, and $\hat{\mathbf{x}}_i^v = D^v(\mathbf{z}_i^v; \rho^v)$ as the reconstruction feature representation corresponding to $\mathbf{z}_i^v$. To ensure that these latent representations can accurately recover view-specific information for each view, we define $\mathcal{L}_{rec}$ as the reconstruction loss between the original input features and their reconstructed features:

$$\mathcal{L}_{rec} = \frac{1}{V}\sum_{v=1}^{V} \|\mathbf{X}^v - D^v\big(E^v(\mathbf{X}^v)\big)\|_F^2. \tag{2}$$

After obtaining view-specific latent representations $\mathbf{Z}^v$, we adopt a view-shared project head $F(\cdot)$ on $\mathbf{Z}^v$ to obtain high-level representations $\mathbf{H}^v = F(\mathbf{Z}^v; \delta^v) \in \mathbb{R}^{N \times k_v}$, where δ^v denotes network parameter and k_v is the dimension of the high-level representations for the v-th view. Our proposed CFDM method aims to integrate these high-level representations from different views and reduce label ambiguity, which is separately achieved through two main modules in Figure

2: *Multi-view Contrastive Fusion Module* and *Multi-class Contrastive Prototype Disambiguation Module*.

Multi-view Contrastive Fusion

Consistency and complementarity are two fundamental principles for boosting multi-view data fusion. Inspired by the success of contrastive learning on multi-view clustering (Wang et al. 2023), we design two contrastive learning components, inter-view contrastive learning and intra-view contrastive learning, to explore the consistency and complementarity of MVPLL data and learn distinctive high-level representations for improving subsequent label disambiguation.

Inter-view Contrastive Learning. Inter-view contrastive learning focuses on learning consistent representations across different views to explore common semantics, which encourages the representations of the same instance to cluster together while pushing apart those of different instances. Therefore, when conducting inter-view contrastive learning, the representations of the same instance from different views are regarded as positive samples, while others are regarded as negative samples. Specifically, given a mini-batch $\{(\mathbf{x}_i, S_i)\}_{i=1}^{M}$ includes M instances with V views, each high-level representation $\mathbf{h}_i^v$ has $(MV-1)$ feature pairs, *i.e.*, $\{\mathbf{h}_i^v, \mathbf{h}_j^m\}_{j=1,..,M}^{m=1,...,V}$,where $\{(\mathbf{h}_i^v, \mathbf{h}_i^m), v \neq m\}$ are $(V-1)$ positive pairs and the remaining $V(M-1)$ representation pairs are negative pairs. Then, the inter-view contrastive loss function between each pair of views v and m is formulated as:

$$\ell_{cont1}^{(vm)} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{d(\mathbf{h}_i^v,\mathbf{h}_i^m)/\tau}}{\sum\limits_{j=1}^{M}\sum\limits_{n=v,m} e^{d\left(\mathbf{h}_i^v,\mathbf{h}_j^n\right)/\tau} - e^{1/\tau}}, \quad (3)$$

where τ is the temperature hyper-parameter, and $d(\cdot,\cdot)$ is the cosine distance to measure the similarity between two high-level representations. After calculating the loss within each pair of views, we compute the total inter-view contrastive loss by averaging the losses across all view pairs:

$$\mathcal{L}_{cont1} = \frac{1}{2V}\sum_{v=1}^{V}\sum_{m=1,v\neq m}^{V}\ell_{cont1}^{(vm)}. \quad (4)$$

Intra-view Contrastive Learning. Intra-view contrastive learning focuses on learning complementary representations within each view to capture specific semantics, making instances of the same class more cohesive and that of different classes more easily distinguishable. Therefore, within each view, positive samples consist of representations from instances belonging to the same class, while representations from different classes are regarded as negative samples. Specifically, given a mini-batch $\{(\mathbf{x}_i, S_i)\}_{i=1}^{M}$ includes M instances with V views, for each $(\mathbf{x}_i^v, S_i)$, we use the output of the classifier to predict its label $\tilde{y} = argmax_{j\in S_i} p_j(\mathbf{x}_i^v)$. Then, we select the positive samples for its high-level representation $\mathbf{h}_i^v$ as follows:

$$\mathbb{P}(\mathbf{x}_i^v) = \{\mathbf{h}_k^v | \mathbf{h}_k^v \in \mathbb{N}(\mathbf{x}_i^v), \tilde{y}' = \tilde{y}\}, \quad (5)$$

where $\mathbb{P}(\mathbf{x}_i^v)$ is the positive set, $\mathbb{N}(\mathbf{x}_i^v) = \mathbf{h}^v \setminus \{\mathbf{h}_i^v\}, \mathbf{h}^v = \{\mathbf{h}_i^v\}_{i=1}^{M} \in \mathbb{R}^{M\times k_v}$, and $\tilde{y}'$ is the predicted label for the corresponding training instance of $\mathbf{h}_k^v$. For view v, we calculate the intra-view contrastive loss as follows:

$$\ell_{cont2}^{(v)} = -\frac{1}{M|A|}\sum_{i=1}^{M}\sum_{\mathbf{h}_n^v\in A}\log\frac{e^{d(\mathbf{h}_i^v,\mathbf{h}_n^v)/\tau}}{\sum\limits_{k=1,k\neq i}^{M} e^{d\left(\mathbf{h}_i^v,\mathbf{h}_k^v\right)/\tau}}, \quad (6)$$

where $A = \mathbb{P}(\mathbf{x}_i^v)$, and $|\cdot|$ denotes the number of elements in a set. After calculating the loss within each view, we establish the overall intra-view contrastive loss by averaging the losses across all views as follows:

$$\mathcal{L}_{cont2} = \frac{1}{V}\sum_{v=1}^{V}\ell_{cont2}^{(v)}. \quad (7)$$

Multi-class Contrastive Prototype Disambiguation

As described in the previous subsections, intra-view contrastive learning relies heavily on accurate classifier predictions for positive set selection. However, in MVPLL, the learned classifier struggles to make correct predictions since the label supervision signals are always ambiguous. To address this issue, we further design a multi-class contrastive prototype disambiguation module to reduce label ambiguity, which consists of two key components: prototype contrastive learning and pseudo label updating.

Prototype Contrastive Learning. Prototype contrastive learning aims to obtain global class prototypes that characterize the attributes of each class accurately, which are further applied to update the pseudo labels. Specifically, we first establish a set of class prototype vectors for each view, where these prototype vectors derive from multi-view contrastive fusion module. In this paper, we update the prototype vector η_c^v by the following moving-average mechanism:

$$\begin{aligned}\eta_c^v = \text{Normalize}\big(\gamma\eta_c^v + (1-\gamma)\mathbf{h}_i^v\big),\\ \text{if } c = argmax_{j\in S_i} p_j(\mathbf{x}_i^v),\end{aligned} \quad (8)$$

where the momentum prototype η_c^v of class c is defined by the moving-average of the normalized high-level representations $\mathbf{h}_i^v$ from instances predicted to be class c. γ is a tunable hyperparameter that controls the fusion rate between the old prototype and the new information. To maintain cross-view consistency of the prototypes, we should encourage the prototype vectors of the same class from different views to be as close as possible. Specifically, for the v-th view, each class prototype η_j^v has $(qV-1)$ feature pairs, *i.e.*, $\{\eta_j^v, \eta_c^m\}_{c=1,..,q}^{m=1,...,V}$, where $\{(\eta_j^v, \eta_j^m), v \neq m\}$ are $(V-1)$ positive pairs and the remaining $V(q-1)$ representation pairs are negative pairs. The formula for the prototype contrastive loss between η^v and η^m is formulated as:

$$\ell_{cont3}^{(vm)} = -\frac{1}{q}\sum_{j=1}^{q}\log\frac{e^{d\left(\eta_j^v,\eta_j^m\right)/\tau}}{\sum\limits_{c=1}^{q}\sum\limits_{n=v,m} e^{d\left(\eta_j^v,\eta_c^n\right)/\tau} - e^{1/\tau}}. \quad (9)$$

Algorithm 1: Pseudo-code of CFDM (one epoch)

Input: MVPLL training dataset: $\mathcal{D} = \{(\mathbf{x}_i, S_i)\}_{i=1}^{N}$; network model: $E^v(\cdot), D^v(\cdot)$ and $F(\cdot)$; trade-off coefficients: $\lambda_1, \lambda_2, \lambda_3$ and λ_4.

Process:

for $iter = 1, 2, ...,$ **do**
  Sample a mini-batch $\{(\mathbf{x}_i, S_i)\}_{i=1}^{M}$ from $\mathcal{D}$.
  Initialize $\{\theta^v, \rho^v\}_{v=1}^{V}$ by minimizing Eq.(2).
  Obtain high-level representations by $F(\cdot)$.
  for $\mathbf{x}_i \in \{(\mathbf{x}_i, S_i)\}_{i=1}^{M}$ **do**
    Compute classifier prediction by $\tilde{y} = argmax_{j \in S_i} p_j(\mathbf{x}_i^v)$.
    Update class prototype by Eq. (8) − Eq. (10).
    Updata pseudo label by Eq. (11).
  end for
  Compute the reconstruction loss $\mathcal{L}_{rec}$, inter-view contrastive loss $\mathcal{L}_{cont1}$, intra-view contrastive loss $\mathcal{L}_{cont2}$, prototype contrastive loss $\mathcal{L}_{cont3}$ and classification loss $\mathcal{L}_{cls}$ respectively using Eq. (2), Eq. (4), Eq. (7), Eq. (10) and Eq. (1).
  Update network parameters by minimizing overall loss $\mathcal{L}$ in Eq. (12).
end for

And correspondingly, the total multi-view prototype contrastive loss across all views is calculated by:

$$\mathcal{L}_{cont3} = \frac{1}{2V} \sum_{v=1}^{V} \sum_{m=1, v \neq m}^{V} \ell_{cont3}^{(vm)}. \tag{10}$$

Pseudo Label Updating. Pseudo label updating focuses on finding the nearest class prototype for the high-level representation of each instance, thereby reducing label ambiguity. Specifically, for each $(\mathbf{x}_i^v, S_i)$, we assign a pseudo label vector $\mathbf{s}_i^v \in [0, 1]^q$, where each element s_{ij}^v indicates the probability of label j being the ground-truth label of $\mathbf{x}_i^v$. Before model training, the pseudo label $\mathbf{s}_i^v$ is initialized with a uniform distribution, $s_{ij}^v = \frac{1}{|S_i|}\mathbb{I}(j \in S_i)$. Then, it is updated according to the distance between the high-level representation of $\mathbf{x}_i^v$ and the class prototype of the v-th view, with the closest class prototype being assigned as the class of the current $\mathbf{x}_i^v$. Finally, we use a moving average strategy to iteratively update the pseudo label $\mathbf{s}_i^v$ as follows:

$$\begin{aligned} \mathbf{s}_i^v &= \beta \mathbf{s}_i^v + (1 - \beta) I_i^v, \\ I_{i,c}^v &= \begin{cases} 1, & \text{if } c = argmax_{j \in S_i} \mathbf{h}_i^{v\top} \eta_j^v, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{11}$$

where $\beta \in (0, 1)$ is a positive constant. For each $\mathbf{x}_i^v$, the closest prototype indicates its ground-truth class label. During the training process, the pseudo label $\mathbf{s}_i^v$ gradually moves towards the one-hot distribution defined by Eq. (11). If $\mathbf{x}_i^v$ consistently points to the same class prototype, the pseudo label $\mathbf{s}_i^v$ will almost converge to the correct label of that class. For multi-view data, the pseudo label $\mathbf{s}_i$ of instance $\mathbf{x}_i$ is computed by averaging the pseudo labels $\mathbf{s}_i^v$ across all views, *i.e.*, $\mathbf{s}_i = \frac{1}{V} \sum_{v=1}^{V} \mathbf{s}_i^v$.

Datasets	Instances	Labels	Views dimensions
MSRCv1	210	5	24/576/512/256/245
Caltech101-7	1474	7	48/40/254/1984/512/928
Mfeat	2000	10	216/76/64/6/240/47
Scene15	4485	15	20/59/40
CCV	6773	20	20/20/20
Caltech101-all	9144	102	48/40/254/512/928

Table 1: Characteristics of our employed datasets.

The Overall Loss Function

By integrating classification loss in Eq. (1), reconstruction loss in Eq. (2), inter-view contrastive loss Eq. (4), intra-view contrastive loss Eq. (7) and prototype contrastive loss Eq. (10), the overall loss function of our model is formulated as:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{cont1} + \lambda_3 \mathcal{L}_{cont2} + \lambda_4 \mathcal{L}_{cont3}, \tag{12}$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are hyper-parameters to balance the weight of different losses.

Optimization

During the training stage, multi-view contrastive fusion and multi-class contrastive prototype disambiguation mutually reinforce each other within a coherent framework. Firstly, the discriminative high-level representations obtained through multi-view contrastive fusion can generate more precise class prototypes, which are beneficial for label disambiguation. Conversely, more accurate label disambiguation results also facilitate the construction of positive sets in multi-view contrastive fusion, leading to more discriminative high-level representations. The collaborative iteration between these two components ultimately enhances the classification performance of the learning model. The pseudo-code of our CFDM is shown in Algorithm 1.

Experiments

Experimental Settings

Datasets. To evaluate the performance of our proposed CFDM, we conducted experiments on six synthetic MVPLL datasets, which are generated from the widely-used multi-view datasets, including *MSRCv1* (Xu, Han, and Nie 2016), *Caltech101-7* (Fei-Fei, Fergus, and Perona 2004), *Mfeat* (Wang, Yang, and Liu 2019), *Scence15* (Fei-Fei and Perona 2005), *CCV* (Jiang et al. 2011), *Caltech101-all* (Fei-Fei, Fergus, and Perona 2004), by randomly adding labeling noise under different configurations of the controlling parameters p and r. Here, $p \in \{0.3, 0.5\}$ controls the proportion of partially labeled instances and $r \in \{1, 2, 3\}$ controls the number of noisy labels in the candidate label set. Table 1 summarizes the characteristics of these employed datasets.

The Comparing Methods. We employ six state-of-the-art methods from three categories for comparative studies: (1) Multi-view partial multi-label methods, including **GRADIS** (Chen et al. 2020) and **GLADE** (Xu et al. 2022), which focus on both multi-view fusion and partial-label disambiguation simultaneously; (2) Multi-view multi-label methods, including **ML-BVAE** (Fu et al. 2024) and **LMVCAT** (Liu

Dataset	Controlling Parameters	Comparing Approach						
		GRADIS	GLADE	ML-BVAE	LMVCAT	UCL	TERIAL	Ours
MSRCv1	$r=1,p=0.3$	0.482±0.074	0.914±0.027	0.891±0.027	0.966±0.036	0.961±0.039	0.880±0.016	**0.971±0.021**
	$r=1,p=0.5$	0.439±0.076	0.857±0.044	0.885±0.068	0.953±0.035	0.952±0.058	0.880±0.016	**0.957±0.031**
	$r=2,p=0.3$	0.481±0.084	0.885±0.026	0.890±0.027	0.957±0.039	0.952±0.047	0.880±0.016	**0.962±0.039**
	$r=2,p=0.5$	0.460±0.092	0.776±0.052	0.870±0.036	0.952±0.037	0.942±0.062	0.876±0.035	**0.953±0.046**
	$r=3,p=0.3$	0.472±0.058	0.852±0.026	0.876±0.199	0.952±0.056	0.942±0.052	0.881±0.168	**0.957±0.019**
	$r=3,p=0.5$	0.457±0.103	0.685±0.051	0.865±0.039	0.933±0.052	0.934±0.079	0.876±0.035	**0.938±0.027**
Caltech101-7	$r=1,p=0.3$	0.853±0.013	0.936±0.011	0.857±0.026	0.972±0.053	0.981±0.013	0.976±0.007	**0.982±0.002**
	$r=1,p=0.5$	0.853±0.021	0.889±0.016	0.858±0.027	0.959±0.017	0.978±0.019	0.973±0.007	**0.979±0.004**
	$r=2,p=0.3$	0.853±0.008	0.879±0.012	0.859±0.022	0.964±0.011	0.976±0.020	0.974±0.004	**0.980±0.011**
	$r=2,p=0.5$	0.853±0.022	0.774±0.021	0.858±0.027	0.945±0.039	0.972±0.019	0.971±0.006	**0.975±0.009**
	$r=3,p=0.3$	0.853±0.029	0.821±0.020	0.860±0.030	0.959±0.019	0.973±0.017	0.975±0.005	**0.978±0.012**
	$r=3,p=0.5$	0.841±0.036	0.679±0.031	0.858±0.024	0.953±0.016	0.971±0.019	0.966±0.006	**0.973±0.011**
Mfeat	$r=1,p=0.3$	0.953±0.012	0.979±0.012	0.974±0.004	0.931±0.037	0.968±0.009	0.969±0.008	**0.987±0.005**
	$r=1,p=0.5$	0.952±0.012	0.970±0.010	0.970±0.005	0.930±0.042	0.968±0.007	0.968±0.008	**0.972±0.006**
	$r=2,p=0.3$	0.953±0.006	0.972±0.009	0.971±0.003	0.927±0.031	0.967±0.009	0.968±0.008	**0.978±0.005**
	$r=2,p=0.5$	0.951±0.015	0.943±0.009	0.968±0.006	0.910±0.039	0.968±0.009	0.968±0.008	**0.974±0.004**
	$r=3,p=0.3$	0.953±0.008	0.961±0.012	0.970±0.005	0.918±0.023	0.968±0.009	0.968±0.009	**0.975±0.006**
	$r=3,p=0.5$	0.951±0.014	0.914±0.010	0.965±0.004	0.908±0.058	0.967±0.009	0.965±0.008	**0.970±0.006**
Scene15	$r=1,p=0.3$	0.542±0.017	0.729±0.017	0.667±0.012	0.684±0.041	0.762±0.012	0.691±0.017	**0.778±0.010**
	$r=1,p=0.5$	0.531±0.014	0.731±0.015	0.663±0.024	0.674±0.020	0.761±0.011	0.681±0.016	**0.767±0.011**
	$r=2,p=0.3$	0.541±0.021	0.723±0.021	0.657±0.018	0.675±0.023	0.761±0.013	0.683±0.017	**0.768±0.004**
	$r=2,p=0.5$	0.531±0.021	0.715±0.017	0.652±0.011	0.670±0.015	**0.761±0.008**	0.664±0.016	0.753±0.008
	$r=3,p=0.3$	0.537±0.021	0.714±0.024	0.654±0.017	0.665±0.028	0.760±0.012	0.679±0.019	**0.762±0.009**
	$r=3,p=0.5$	0.535±0.019	0.703±0.020	0.653±0.017	0.662±0.020	**0.757±0.006**	0.654±0.020	0.748±0.005
CCV	$r=1,p=0.3$	0.167±0.010	0.455±0.020	0.346±0.016	0.512±0.020	0.534±0.012	0.467±0.020	**0.537±0.011**
	$r=1,p=0.5$	0.159±0.013	0.452±0.020	0.341±0.021	0.498±0.011	0.530±0.027	0.455±0.021	**0.534±0.019**
	$r=2,p=0.3$	0.156±0.008	0.454±0.020	0.341±0.019	0.512±0.016	0.532±0.013	0.461±0.019	**0.533±0.013**
	$r=2,p=0.5$	0.147±0.009	0.448±0.020	0.341±0.013	0.480±0.016	0.522±0.014	0.442±0.020	**0.532±0.018**
	$r=3,p=0.3$	0.158±0.018	0.450±0.023	0.340±0.021	0.499±0.014	0.531±0.018	0.455±0.021	**0.532±0.013**
	$r=3,p=0.5$	0.149±0.023	0.442±0.024	0.340±0.103	0.461±0.013	0.521±0.018	0.434±0.020	**0.527±0.009**
Caltech101-all	$r=1,p=0.3$	0.294±0.013	0.614±0.013	0.293±0.011	0.641±0.011	0.653±0.008	0.544±0.003	**0.668±0.011**
	$r=1,p=0.5$	0.287±0.005	0.604±0.011	0.288±0.006	0.615±0.011	0.642±0.003	0.531±0.003	**0.662±0.025**
	$r=2,p=0.3$	0.292±0.011	0.603±0.011	0.289±0.010	0.629±0.018	0.631±0.006	0.534±0.004	**0.661±0.026**
	$r=2,p=0.5$	0.289±0.010	0.586±0.007	0.286±0.006	0.590±0.013	0.629±0.003	0.513±0.003	**0.658±0.017**
	$r=3,p=0.3$	0.288±0.009	0.589±0.009	0.287±0.011	0.613±0.013	0.626±0.008	0.529±0.003	**0.660±0.010**
	$r=3,p=0.3$	0.287±0.010	0.568±0.008	0.286±0.005	0.564±0.006	0.621±0.004	0.500±0.004	**0.642±0.010**

Table 2: Performance comparisons between our proposed CFDM and other comparing methods on six synthetic datasets, where the best performance is shown in bold.

et al. 2023a), which focus on multi-view fusion and treat all candidate labels as true labels; (3) Partial label learning methods, including **UCL** (Dong et al. 2023) and **TERIAL** (Bao, Rui, and Zhang 2024), which focus on label disambiguation and concatenate all view features as the inputs of the learning model. The configuration parameters for each comparing method are set according to the recommendations from their respective literature.

Implementation Details. We implement our experiments based on Pytorch (Paszke et al. 2019). The encoder $E^v(\cdot)$ and decoder $D^v(\cdot)$ are formed by four-layer fully connected networks, where the dimensions are respectively set as $\{d_v, 500, 500, 2000, 512\}$ and $\{512, 2000, 500, 500, d_v\}$. The multi-view shared feature project head $F(\cdot)$ is composed of a two-layer MLP with dimensions $\{512, 256\}$, and utilizes ReLU as its activation function. The training process uses the Adam optimizer, and the learning rate is chosen from $\{1e-4, 3e-4, 5e-4\}$. The hyperparameter γ is set to 0.99, τ is set to 0.07, while β linearly decreases from 0.95 to 0.8. The number of training epochs is set to 130. All experiments are conducted on a machine equipped with an Intel(R) Xeon(R) Gold 6148 2.40GHz CPU, GeForce RTX 3090 GPU, and 512GB RAM. For all experiments, we utilize 5-fold cross-validation, and record the mean and standard deviation (mean ± std) as the final results.

Comparison Results

Table 2 presents the average test accuracy and standard deviation of our proposed CFDM method compared with six sota methods on six datasets. According to 216 (6 comparing methods × 6 datasets × 6 configurations) statistical compar-

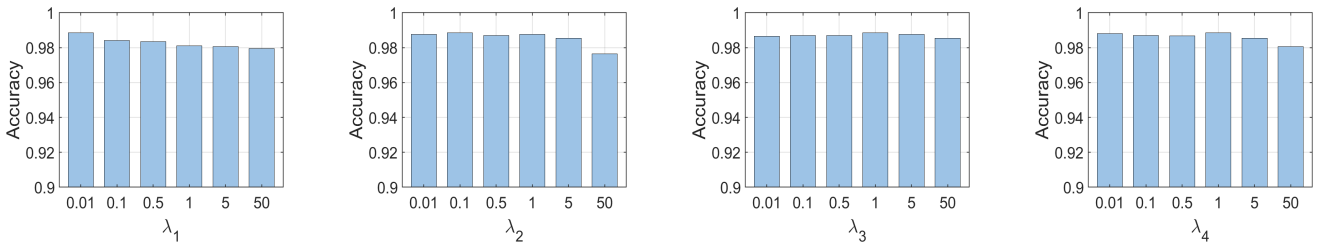

Figure 3: Accuracy with different parameters $\lambda_1, \lambda_2, \lambda_3$ and λ_4 on *Mfeat* dataset with $r = 1$ and $p = 0.3$.

	$\mathcal{L}_{cls}$	$\mathcal{L}_{rec}$	$\mathcal{L}_{cont1}$	$\mathcal{L}_{cont2}$	$\mathcal{L}_{cont3}$	Accuracy
(A)	✓		✓	✓	✓	0.656±0.021
(B)	✓	✓		✓	✓	0.641±0.010
(C)	✓	✓	✓		✓	0.642±0.015
(D)	✓	✓	✓	✓		0.655±0.013
Ours	✓	✓	✓	✓	✓	0.668±0.011

Table 3: Ablation study on *Caltech101-all* dataset with $r = 1$ and $p = 0.3$.

isons, the following observations can be drawn:

- Among all six comparing methods, our proposed CFDM method is superior to **GRADIS**, **GLADE**, **ML-BVAE**, **LMVCAT**, and **TERIAL** in all cases, and it also outperforms **UCL** in 94.4% cases.
- Among all employed datasets, CFDM outperforms all comparing methods on *MSRCv1*, *Caltech101-7*, *Mfeat*, *CCV*, *Caltech101-all* datasets. And on *Scene15*, it is also superior to other comparing methods over 94.4% cases.
- The improvements of CFDM against other methods are quite significant, especially in *Caltech101-all* dataset, which demonstrates the robustness and effectiveness of our proposed method to address the MVPLL datasets.

Further Analysis

Ablation Study

To evaluate the effect of each component within CFDM, we conduct the ablation study between CFDM and its four degenerated algorithms, where each degenerated algorithm separately ignores the reconstruction loss $\mathcal{L}_{rec}$, inter-view contrastive loss $\mathcal{L}_{cont1}$, intra-view contrastive loss $\mathcal{L}_{cont2}$, and contrastive prototype loss $\mathcal{L}_{cont3}$. Table 3 records the experimental results of accuracy on *Caltech101-all* dataset. According to Table 3, CFDM is siginificantly superior to the degenerated algorithms (B) and (C), which indicates inter-view and intra-view contrastive learning can effectively explore the multi-view consistency and complementarity to learn discriminative representations. Meanwhile, CFDM also outperforms the degenerated algorithms (A) and (D), which indicates view-specific autoencoder network can eliminate the negative effects of multi-view feature redundancy and random noise, and prototype contrast learning can learn more precise class prototypes to promote label disambiguation. Overall, CFDM significantly outperforms all

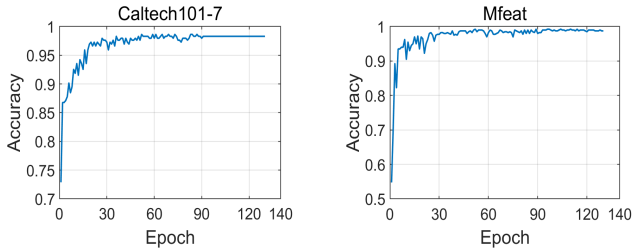

Figure 4: The convergence analysis on *Caltech101-7* and *Mfeat* dataset with $r = 1$ and $p = 0.3$.

of its four degenerated algorithms, which demonstrates that CFDM can effectively integrate the above components and improve the performance of the final classification model.

Sensitivity Analysis

We conduct the sensitivity analysis of CFDM with regard to its four employed parameters ($i.e.$, $\lambda_1, \lambda_2, \lambda_3$ and λ_4) and select each parameter from $\{0.01, 0.1, 0.5, 1, 5, 50\}$. Figure 3 shows the experimental results on *Mfeat* dataset. As illustrated in Figure 3, our proposed CFDM achieves good performance in a wide range of four parameters, which indicates it is insensitive to all parameters $\lambda_1, \lambda_2, \lambda_3$ and λ_4.

Convergence Analysis

We conduct the convergence analysis of our CFDM method on both *Caltech101-7* [left] and *Mfeat* [right] datasets, where experimental results of accuracy are illustrated in Figure 4. According to Figure 4, it is clearly observed that as the number of epochs increases, the accuracy gradually increases and soon reaches stability. Such phenomenon empirically confirms the convergence of our proposed CFDM.

Conclusion

In this paper, we design a general MVPLL framework and propose a novel learning approach named CFDM, which explores the consistency and complementarity of multi-view data by multi-view contrastive fusion and reduces label ambiguity by multi-class contrastive prototype disambiguation. Compared with previous methods, CFDM integrates multi-view heterogeneous information, and establishes the compact relationships between multi-view features and ambiguous labels to facilitate label disambiguation, which signifi-

cantly improves the model's performance. Extensive experimental results demonstrate that our proposed method exhibits significant superiority against the state-of-the-art approaches when learning from MVPLL data.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2023YFB3107100), the National Natural Science Foundation of China (No. 62306020, 62203024, 62173286), the Young Elite Scientist Sponsorship Program by BAST (No. BYESS2024199), the R&D Program of Beijing Municipal Education Commission (No. KM202310005027), the Major Research Plan of National Natural Science Foundation of China (No. 92167102), and the Beijing Natural Science Foundation (No. L244009).

References

Appice, A.; and Malerba, D. 2015. A co-training strategy for multiple view clustering in process mining. *IEEE Transactions on Services Computing*, 9(6): 832–845.

Bao, W.; Rui, Y.; and Zhang, M. 2024. Disentangled partial label learning. In *AAAI Conference on Artificial Intelligence*, 11007–11015.

Cao, X.; Guo, Y.; Yang, W.; Luo, X.; and Xie, S. 2023. Intrinsic feature extraction for unsupervised domain adaptation. *International Journal of Web Information Systems*, 19(5/6): 173–189.

Chai, J.; Tsang, I. W.; and Chen, W. 2019. Large margin partial label machine. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7): 2594–2608.

Chen, Y.; Wang, S.; Zhao, Y.; and Chen, C. P. 2024. Double discrete cosine transform-oriented multi-view subspace clustering. *IEEE Transactions on Image Processing*, 33: 2491–2501.

Chen, Z.; Wu, X.; Chen, Q.; Hu, Y.; and Zhang, M. 2020. Multi-view partial multi-label learning with graph-based disambiguation. In *AAAI Conference on Artificial Intelligence*, 3553–3560.

Dong, R.; Hang, J.; Wei, T.; and Zhang, M. 2023. Can label-specific features help partial-label learning? In *AAAI Conference on Artificial Intelligence*, 7432–7440.

Du, S.; Liu, Z.; Chen, Z.; Yang, W.; and Wang, S. 2021. Differentiable bi-sparse multi-view co-clustering. *IEEE Transactions on Signal Processing*, 69: 4623–4636.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 178–178.

Fei-Fei, L.; and Perona, P. 2005. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 524–531.

Feng, L.; Lv, J.; Han, B.; Xu, M.; Niu, G.; Geng, X.; An, B.; and Sugiyama, M. 2020. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems*, 10948–10960.

Fu, K.; Du, C.; Wang, S.; and He, H. 2024. Multi-view multi-label fine-grained emotion decoding from human brain activity. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7): 9026–9040.

Gu, Z.; Feng, S.; Hu, R.; and Lyu, G. 2023. ONION: Joint unsupervised feature selection and robust subspace extraction for graph-based multi-view clustering. *ACM Transactions on Knowledge Discovery from Data*, 17(5): 1–23.

Guo, Y.; Yu, H.; Ma, L.; Zeng, L.; and Luo, X. 2023. THFE: A triple-hierarchy feature enhancement method for tiny boat detection. *Engineering Applications of Artificial Intelligence*, 123: 106271.

Jiang, Y.; 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 *ACM International Conference on Multimedia Retrieval*, 1–8.

Li, H.; Wei, T.; Yang, H.; Hu, K.; Peng, C.; Sun, L.; Cai, X.; and Zhang, M. 2023. Stochastic feature averaging for learning with long-tailed noisy labels. In *International Joint Conference on Artificial Intelligence*, 3902–3910.

Li, X.; Sun, Y.; Sun, Q.; and Ren, Z. 2022. Consensus cluster center guided latent multi-kernel clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6): 2864–2876.

Lin, F.; Bai, B.; Bai, K.; Ren, Y.; Zhao, P.; and Xu, Z. 2022. Contrastive multi-view hyperbolic hierarchical clustering. In *International Joint Conference on Artificial Intelligence*, 3250–3256.

Liu, C.; Wen, J.; Luo, X.; and Xu, Y. 2023a. Incomplete multi-view multi-label learning via label-guided masked view-and category-aware transformers. In *AAAI Conference on Artificial Intelligence*, 8816–8824.

Liu, C.; Wen, J.; Wu, Z.; Luo, X.; Huang, C.; and Xu, Y. 2024. Information recovery-driven deep incomplete multi-view clustering network. *IEEE Transactions on Neural Networks and Learning Systems*, 35(11): 15442–15452.

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 *AAAI Conference on Artificial Intelligence*, 7576–7584.

Liu, W.; Yuan, J.; Lyu, G.; and Feng, S. 2023b. Label driven latent subspace learning for multi-view multi-label classification. *Applied Intelligence*, 53(4): 3850–3863.

Luong, K.; Nayak, R.; Balasubramaniam, T.; and Bashar, M. A. 2022. Multi-layer manifold learning for deep non-negative matrix factorization-based multi-view clustering. *Pattern Recognition*, 131: 108815.

Lv, J.; Liu, B.; Feng, L.; Xu, N.; Xu, M.; An, B.; Niu, G.; Geng, X.; and Sugiyama, M. 2023. On the robustness of average losses for partial-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5): 2569–2583.

Lyu, G.; Kang, W.; Wang, H.; Li, Z.; Yang, Z.; and Feng, S. 2024. Common-individual semantic fusion for multi-view multi-label learning. In *International Joint Conference on Artificial Intelligence*, 4715–4723.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 8024–8035.

Sun, S.; Yu, X.; and Tian, Y. 2023. Multi-view prototype-based disambiguation for partial label learning. *Pattern Recognition*, 141: 109625.

Wang, H.; Xiao, R.; Li, Y.; Feng, L.; Niu, G.; Chen, G.; and Zhao, J. 2021. PiCO: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations*, 1–18.

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

Wang, J.; Feng, S.; Lyu, G.; and Gu, Z. 2023. Triple-granularity contrastive learning for deep multi-view subspace clustering. In *ACM International Conference on Multimedia*, 2994–3002.

Wang, W.; and Zhang, M. 2022. Partial label learning with discrimination augmentation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1920–1928.

Wen, J.; Liu, C.; Deng, S.; Liu, Y.; Fei, L.; Yan, K.; and Xu, Y. 2024. Deep double incomplete multi-view multi-label learning with incomplete labels and missing views. *IEEE transactions on neural networks and learning systems*, 35(8): 11396–11408.

Xia, S.; Lv, J.; Xu, N.; Niu, G.; and Geng, X. 2023. Towards effective visual representations for partial-label learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 15589–15598.

Xu, J.; Han, J.; and Nie, F. 2016. Discriminatively embedded k-means for multi-view clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 5356–5364.

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

Xu, N.; Liu, B.; Lv, J.; Qiao, C.; and Geng, X. 2023. Progressive purification for instance-dependent partial label learning. In *International Conference on Machine Learning*, 38551–38565.

Zhang, C.; Jiang, B.; Wang, Z.; Yang, J.; Lu, Y.; Wu, X.; and Sheng, W. 2023. Efficient multi-view semi-supervised feature selection. *Information Sciences*, 649: 119675.

Zhang, C.; Wang, S.; Liu, J.; Zhou, S.; Zhang, P.; Liu, X.; Zhu, E.; and Zhang, C. 2021. Multi-view clustering via deep matrix factorization and partition alignment. In *ACM International Conference on Multimedia*, 4156–4164.

Zhang, M.; and Yu, F. 2015. Solving the partial label learning problem: An instance-based approach. In *International Joint Conference on Artificial Intelligence*, 4048–4054.

Zhao, L.; Xiao, Y.; Liu, B.; and Hao, Z. 2022. Multi-view partial label machine. *Information Sciences*, 586: 310–325.

Zhong, Q.; Lyu, G.; and Yang, Z. 2024. Align while fusion: A generalized nonaligned multiview multilabel classification method. *IEEE Transactions on Neural Networks and Learning Systems*, 1–10.