Incomplete Multi-View Multi-Label Classification via Diffusion-Guided Redundancy Removal

Shilong Ou[1], Zhe Xue[1*], Lixiong Qin[1], Yawen Li[1], Meiyu Liang[1], Junjiang Wu[1], Xuyun Zhang[2], Amin Beheshti[2], Yuankai Qi[2*]

[1]Beijing University of Posts and Telecommunications, China
[2]Macquarie University, Australia
{osl, xuezhe, lxqin, meiyu1210, wujunjiang}@bupt.edu.cn, {xuyun.zhang, amin.beheshti, yuankai.qi}@mq.edu.au, warmly0716@126.com

Abstract

Incomplete multi-view multi-label classification aims to accurately predict labels for each sample in the face of some missing views. Due to its widespread presence in real-world scenarios, it has become an extensively researched topic. In addition to the challenges brought by missing views, it also encounters issues caused by redundant views, whose inclusion fails to make a positive contribution to performance. In this paper, we make the first attempt to take advantage of diffusion models to address the missing view problem and design a strategy to identify and remove redundant views. Specifically, we train a diffusion model conditioned on the pseudo-labels to recover information of missing views. The learned diffusion model can carry data distribution knowledge in training split to the data. Regarding redundant identification strategy, it is designed by considering both the additional information of views and the classification difficulty level of samples, thereby adaptively identifying and removing redundant views. We conduct extensive experiments on five datasets, and the proposed method achieves favorable performance against several state-of-the-art methods on the multi-view multi-label classification task.

Introduction

Multi-view data, which involves the collection of data from different media, is ubiquitously present in the real world (Zhang et al. 2013b; Wang et al. 2021). Multi-label data (Sun et al. 2024), capable of representing complex relationships in the real world, is also becoming prevalent. Based on the characteristics of the data, multi-view multi-label learning is proposed (Fang and Zhang 2012), where the data under this scenario includes information from multiple views while concurrently belonging to several categories. Multi-view multi-label learning has garnered considerable research interest (Liu et al. 2015; Tan et al. 2018; Wang et al. 2022). Among this, the problem of multi-view multi-label classification (MVMLC) stands as a primary focus of investigation. It necessitates the extraction of features from the views and the fusion of these features, followed by mapping them onto the label space.

In real-world scenarios, observations from different views of an object or event are not entirely independent. For instance, the RGB and HSV views of the same scene possess a large amount of similar structural information. At the same time, each view inevitably contains noise, therefore, more views do not mean better performance. We define this observation as the view redundancy phenomenon. Thus, identifying and removing redundant views is important for achieving better final performance. RMVC (Tao et al. 2018) employs the squared χ^2 distance for clustering, aiming to ensure that the performance of multiple views is not inferior to that of a single view; DSMVC (Tang and Liu 2022) proposes a method for safely adding views, by discarding newly added views that contain more noise than supplementary information, thus reducing the performance degradation brought by increasing the number of views. These methods alleviate the issue of redundant views, but they overlook the difference in each individual data.

On the other hand, MVMLC often encounters the issue of incomplete views or incomplete labels in real-world datasets. For example, a particular view of an image may be missing due to damage to the sensor, and multimedia content on social platforms can be under-annotated by users. To tackle the challenges of incomplete views and incomplete labels, recent innovations include the development of deep contrastive networks (Trosten et al. 2021; Liu et al. 2023a), transformer (Liu et al. 2023b) and deep graph-based networks (Li et al. 2024). These methods proficiently handle the incomplete multi-view multi-label data within datasets; however, they still fail to make full use of the data distribution knowledge inherent in the views and labels present in the datasets, rendering them unable to overcome the performance limitations introduced by incompleteness.

To address these issues, we introduce *DiffSumm*, a redundancy-driven diffusion model for multi-view multi-label classification. Specifically, we design a view encoding module to transform the original views into latent space vectors for multi-label classification, referred to as *summary vectors*. Subsequently, we design a label-guided *diffusion* module to recover summary vectors for data containing missing views. Thereafter, we design a redundant view identification strategy that utilizes the synthetic summary vectors and their pseudo-labels to identify and remove redundant views, and then regenerates new synthetic summary vectors and pseudo-labels with the removed versions to manifest the data distribution knowledge of the synthetic summary

*Corresponding authors.

vectors. Finally, in the final prediction module, we fuse the original summary vectors with the synthetic summary vectors to obtain the final predictions. Comprehensive experiments validate the superior performance of our proposed DiffSumm framework. To summarize, this work makes the following contributions:

- To the best of our knowledge, this is the first work to employ a diffusion model to handle the incomplete multi-view multi-label classification problems, which recovers the lost information of missing views.
- We introduce a redundant view identification strategy with assistance of diffusion model, which identifies and removes redundant views at the sample level, reducing the data noise and enhancing model performance.
- Extensive experiments show the favorable performance on five datasets compared to state-of-the-art methods.

Related Work

Incomplete Multi-View Multi-Label Classification A variety of strategies have been proposed for addressing the multi-view multi-label classification challenge. LrMMC (Liu et al. 2015) maintains a low-rank common subspace, aligning with matrix completion strategies and determines the weights for each view to harness their unique advantages. To address the difficulties with datasets lacking complete views or labels, MVL-IV (Xu, Tao, and Xu 2015) explores the realm of incomplete multi-view learning by exploiting inter-view relationships. Similarly, the iMSF approaches the challenge of incomplete multi-view single-label learning by dividing it into multiple complete tasks (Yuan et al. 2012). On the other hand, MvEL (Zhang et al. 2013a) focuses on mining context and neighborhood consistency for incomplete labels, but it is limited to only addressing label incompleteness. In the realm of tasks involving incomplete views and labels, iMvWL (Tan et al. 2018) integrates multi-view and multi-label elements into a unified subspace. Conversely, NAIML (Li and Chen 2022) tackles the dual incompleteness by employing a low-rank framework for the sub-label matrix and sub-classes. Deep learning approaches have also been increasingly applied to these challenges. DDINet (Wen et al. 2023) introduces a deep learning architecture with a view-specific decoder network that adeptly retains essential view information. DICNet (Liu et al. 2023a) pioneers the use of instance-level contrastive learning in this problem. LMVCAT (Liu et al. 2023b), a transformer-based architecture, effectively addresses this problem by incorporating two transformers designed for views and labels. VIST (Ou et al. 2024) addresses this issue by introducing a view-label interaction transformer, effectively handles data incompleteness and improves classification performance.

Redundant Views in Multi-View Learning RMVC (Tao et al. 2018) marks a pioneering effort in ensuring safety within the domain of multi-view clustering. In a similar vein, DSMTL (Yue et al. 2021) introduces a safe multi-task model, designed to perform at least as well as its single-task counterparts for each respective task. DSMVC (Tang and Liu 2022) diverges from these approaches by aiming to ensure that the introduction of an additional view does not compromise the clustering efficacy of the pre-existing dataset and the data associated with the newly added view.

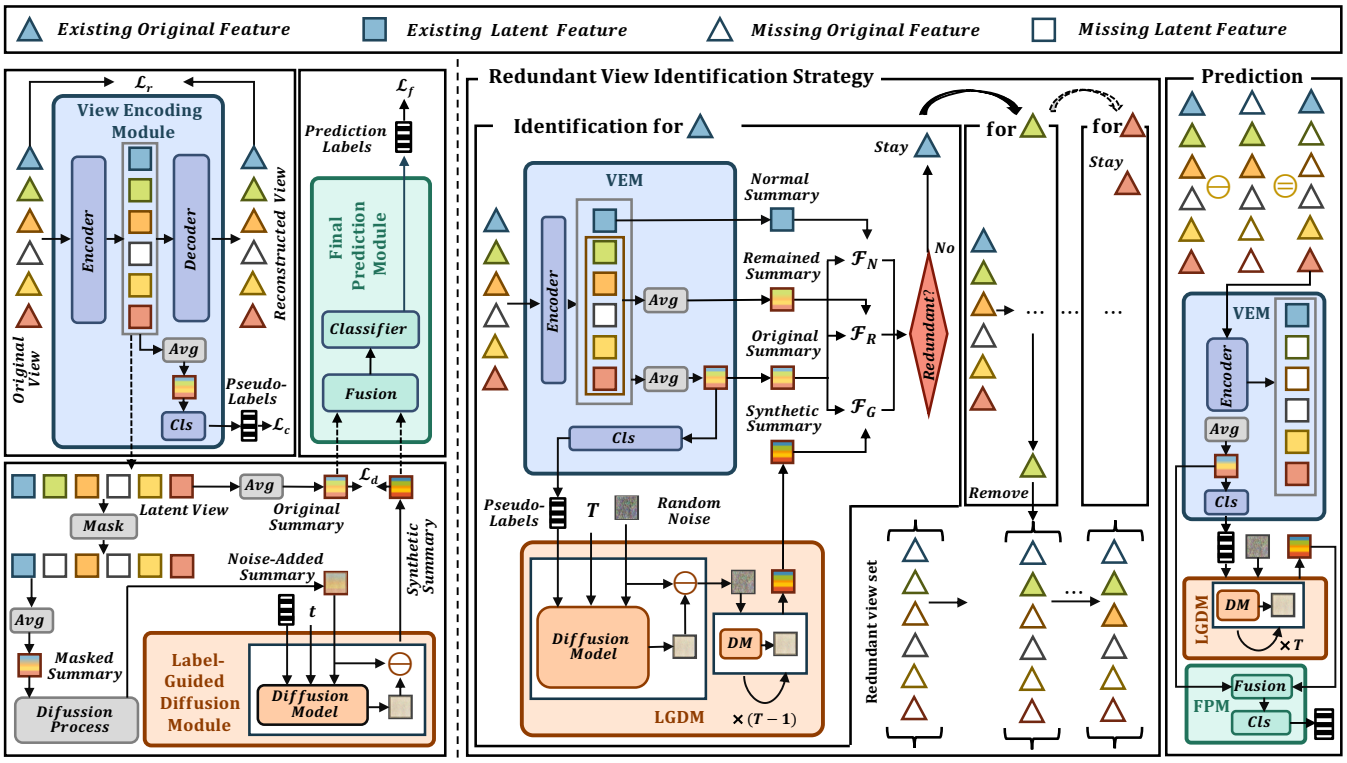
Diffusion Models Diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) are a class of generative models that have gained popularity in recent years. To accelerate the denoising process, DDIM (Song, Meng, and Ermon 2021) introduce a strategy to reduce the number of denoising steps. LDM (Rombach et al. 2022) seeks to perform diffusion in latent space rather than pixel space, reducing training costs and speeding up inference. Although these methods can generate high-quality samples, they do not specify the content of the generation. GLIDE (Nichol et al. 2022) and Imagen (Saharia et al. 2022) achieve classifier-free guidance (Ho and Salimans 2022) by directly incorporating conditional information into the training of the diffusion model.

Methodology

We primarily present our method DiffSumm as three modules: view encoding module, label-guided diffusion module and final prediction module. The training process is illustrated in Figure 1a. We train the view encoding module to obtain original summary vectors and pseudo-labels. Concurrently, we train a label-guided diffusion module to generate synthetic summary vectors. Finally, we train the final prediction module using both the original and synthetic summary vectors. Subsequently, during the inference phase in Figure 1b, data first passes through view encoding module to obtain original summary vectors and pseudo-labels. Label-guided diffusion module then uses these pseudo-labels to generate synthetic summary vectors. These, along with normal and remained summary vectors generated from other view combinations, assist the model in identifying and removing redundant views, thereby getting new view combinations. These new combinations are then reintroduced into view encoding module, completing a step in the loop of the strategy. After the loop ends, the original and synthetic summary vectors are input into the final prediction module for the final prediction. We will provide a detailed explanation of this method.

Preliminaries

Diffusion Models Diffusion models constitute a type of generative model that primarily encompasses a forward diffusion process and a reverse denoising process. The forward diffusion process samples a data point from the distribution of a dataset and subsequently adds noise with variance $\beta_t \in (0, 1)$ to the sample through each step $t \in \{1, 2, \ldots, T\}$, until it converges to the standard normal distribution $\mathbf{s}_T \sim \mathcal{N}(0, \mathbf{I})$, i.e., $q(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t; \sqrt{1-\beta_t}\mathbf{s}_{t-1}, \beta_t\mathbf{I})$. Conversely, the reverse denoising process employs a diffusion model parameterized by θ to restore the sampled $\mathbf{s}_t$ through T time step back to $\mathbf{s}_0$, which can be represented as $p_\theta(\mathbf{s}_{t-1}|\mathbf{s}_t) = \mathcal{N}(\mathbf{s}_{t-1}; \mu_\theta(\mathbf{s}_t, t), \Sigma_\theta(\mathbf{s}_t, t))$. The model is trained through the maximization of the variational lower bound of the negative log likelihood, which, according to



(a) Training Process (b) Inference Process

Figure 1: The overview of our method DiffSumm. For simplicity, we omit the term "vector" from the summary vector. During the inference process, the model first performs the redundant view identification strategy, repeatedly utilizing view encoding module and label-guided diffusion module to identify and remove redundant views in sequential order. After the strategy, the model inputs the original and synthetic summary vector based on the final view combination into final prediction module.

(Ho, Jain, and Abbeel 2020), can be simplified as the following loss function:

$$\mathcal{L}_{simple} = \mathbb{E}_{t,\mathbf{s}_0,\epsilon}[||\epsilon - \epsilon_\theta(\mathbf{s}_t, t)||^2] \quad (1)$$

where ϵ_θ is the prediction of ϵ. For $\mathbf{s}_t$, the denoising process can be iteratively conducted as follows:

$$\mathbf{s}_{t-1} = \mu_\theta(\mathbf{s}_t, t) + \sqrt{\Sigma_\theta(\mathbf{s}_t, t)}\mathbf{m}, \quad \mathbf{m} \sim \mathcal{N}(0, \mathbf{I}) \quad (2)$$

Problem Definition For n samples from V different views, they are represented by a set of matrices $\mathcal{X} = \{\mathbf{X}_v\}_{v=1}^V$, where $\mathbf{X}_v \in \mathbb{R}^{n \times d_v}$ and d_v represents the dimension of data from the v-th view. The matrix $\mathbf{Y} \in \{0,1\}^{n \times L}$, serves as the label matrix for all samples, with L indicating the total number of categories. $\mathbf{Y}_{i,j} = 1$ signifies that the i-th sample belongs to the j-th category, whereas $\mathbf{Y}_{i,j} = 0$ denotes the opposite. For handling multi-view data where certain features may be missing, we employ a missing-view indicator matrix $\mathbf{W} \in \{0,1\}^{n \times V}$. In this matrix, $\mathbf{W}_{i,j} = 1$ indicates the presence of the j-th view for the i-th sample, while $\mathbf{W}_{i,j} = 0$ implies its absence. Likewise, the missing-label indicator matrix $\mathbf{U} \in \{0,1\}^{n \times L}$ is defined, where $\mathbf{U}_{i,j} = 1$ suggests the availability of the j-th category label for the i-th sample, and $\mathbf{U}_{i,j} = 0$ indicates the contrary. During the initial data processing stage, any missing values in views $\mathcal{X}$ or labels $\mathbf{Y}$ are assigned a value of 0. The goal of this paper is to develop a model for the task of incomplete multi-view multi-label classification.

Model Training

View Encoding Module In the original data, there often exists an abundance of noise and redundant information. Therefore, it is necessary to first extract the essential information and latent features. Similar to prior work in deep learning (Wen et al. 2020), we employ autoencoders for this task. Taking the original data $\mathbf{X}_v$ from a view as an example, it is transformed into latent representations through an encoder E_v specific to that view, denoted as $\mathbf{Z}_v = E_v(\mathbf{X}_v)$, where $\mathbf{Z}_v \in \mathbb{R}^{n \times d_z}$ denotes the latent representation of that view, and d_z is the dimension of latent space. Correspondingly, the latent representations are then passed through a decoder D_v, also specific to that view, which is utilized to reconstruct the original view, represented as $\hat{\mathbf{X}}_v = D_v(\mathbf{Z}_v)$, where $\hat{\mathbf{X}}_v$ denotes the reconstructed view. Due to the incompleteness of the views, it is important to consider the impact of the missing-view indicator matrix $\mathbf{W}$ on training. Thus,

the autoencoder is trained using the following loss function:

$$\mathcal{L}_r = \frac{1}{V}\sum_{v=1}^{V}(\frac{1}{d_v}\sum_{i=1}^{n}||\hat{\mathbf{x}}_{i,v} - \mathbf{x}_{i,v}||_2^2\mathbf{W}_{i,v}) \quad (3)$$

where $x_{i,v}$ and $\hat{x}_{i,v}$ represent the original and reconstructed feature of the i-th sample under the v-th view. Noted that this loss function calculates the loss across all views, rather than for a single view.

At the sample level, to obtain the original summary vector, it is necessary to fuse the latent representations from different views into a unified representation. Drawing upon the previous work (Chen et al. 2022), we employ an average fusion method that accounts for missing views to facilitate the exchange of information across different views:

$$\mathbf{s}_i = F(\mathbf{z}_{i,1}, \mathbf{z}_{i,2}, \ldots, \mathbf{z}_{i,V}, \mathbf{W}) = \frac{\sum_{v=1}^{V}\mathbf{z}_{i,v}\mathbf{W}_{i,v}}{\sum_{v=1}^{V}\mathbf{W}_{i,v}} \quad (4)$$

where $\mathbf{z}_{i,v}$ denotes the latent representation of the i-th sample under the v-th view, and $\mathbf{s}_i$ is the unified representation of the i-th sample, which is the original summary vector we define. To ensure that the original summary vector can accurately guide the model to complete classification tasks, it is necessary to train a classifier C_p, which should be capable of recognizing the category information contained within the summary vector, the process is delineated as $\mathbf{P} = C_p(\mathbf{S})$. Herein, $\mathbf{P} \in \mathbb{R}^{n\times L}$ is the prediction of the classifier, and a Sigmoid activate function is applied so that the value of the output is in $[0, 1]$. $\mathbf{S} = [\mathbf{s}_1^T\mathbf{s}_2^T\cdots\mathbf{s}_n^T]^T \in \mathbb{R}^{n\times d_z}$ denotes the collection matrix of summary vectors. Finally, we incorporate the missing label indicator matrix $\mathbf{U}$ into the binary cross-entropy loss which is widely utilized in multi-label classification tasks, enabling the model to compare predictions and labels in scenarios with incomplete labels:

$$\begin{aligned}\mathcal{L}_c = -\frac{1}{nL}\sum_{i=1}^{n}\sum_{j=1}^{L}(\mathbf{Y}_{i,j}log(\mathbf{p}_{i,j}) + \\ (1-\mathbf{Y}_{i,j})log(1-\mathbf{p}_{i,j}))\mathbf{U}_{i,j}\end{aligned} \quad (5)$$

where $\mathbf{p}_{i,j}$ represents the predict value of the j-th label for the i-th sample.

Label-Guided Diffusion Module Due to the incompleteness of views, the original summary vectors obtained from the view encoding module may not fully represent the results across all views to a certain extent. Therefore, we anticipate that diffusion models, as the current most popular generative models, could provide some information for missing views. Our goal is to generate a synthetic summary vector for full views based on pseudo-labels. For a given sample i, the set of the label is $\mathcal{Y} = \{j|\mathbf{Y}_{i,j} = 1\}$. And we can obtain its original summary vector $\mathbf{s} = F(E_1(\mathbf{x}_1), E_2(\mathbf{x}_2), \ldots, E_V(\mathbf{x}_V), \mathbf{W})$ through the trained autoencoder of the summary vector encoding module (for simplicity, i is omitted in the remaining part of this section). Additionally, we can also mask out certain views randomly to obtain a masked summary vector $\bar{\mathbf{s}} = F(E_1(\mathbf{x}_1), E_2(\mathbf{x}_2), \ldots, E_V(\mathbf{x}_V), Mask(\mathbf{W}, k_{mask}))$, where $Mask(\cdot,\cdot)$ is a function that randomly masks according to a masking rate k_{mask}. To simplify the multi-label task, we sequentially combine each label in $\mathcal{Y}$ with the mask summary vector $\bar{\mathbf{s}}$. Taking $y \in \mathcal{Y}$ as the condition, and with the summary vector $\bar{\mathbf{s}}$ regarded as $\bar{\mathbf{s}}_0$, we get the noise-added summary vector $\bar{\mathbf{s}}_t$ at each time step t using the reparameterization operation (Sohl-Dickstein et al. 2015):

$$q(\bar{\mathbf{s}}_t|\bar{\mathbf{s}}_0, y) = \mathcal{N}(\bar{\mathbf{s}}_t; \sqrt{\bar{\alpha}_t}\bar{\mathbf{s}}_0, (1-\bar{\alpha}_t)\mathbf{I}) \quad (6)$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$. During the training process, the diffusion model, parameterized by θ, needs to predict the denoised variant of the input $\bar{\mathbf{s}}_t$ conditioned on time step t and label y. Here, since $\bar{\mathbf{s}}$ is obtained by masking certain views, we consider $\bar{\mathbf{s}}$ as a noised variant of $\mathbf{s}$:

$$\tilde{\epsilon} = \epsilon + \bar{\mathbf{s}} - \mathbf{s} \quad (7)$$

where ϵ represents the noise sampled from the standard Gaussian distribution. Consequently, we can reformulate Equation 1 into the following form:

$$\mathcal{L}_d = \mathbb{E}_{t,\mathbf{s},y,\tilde{\epsilon}}[||\tilde{\epsilon} - \epsilon_\theta(\bar{\mathbf{s}}_t, t, y)||^2] \quad (8)$$

Final Prediction Module To combine the obtained original summary vectors and synthetic summary vectors into a single unified representation, we employ an adaptive weighted fusion method. The fused vectors are then input into a new classifier C_f to generate the final predictions, as outlined in the following process:

$$\mathbf{F} = C_f(l_{ori}\mathbf{s} + l_{syn}\mathbf{s}^d) \quad (9)$$

where l_{ori} and l_{syn} are learnable scalar weight, $\mathbf{s}$ and $\mathbf{s}^d$ are original and synthetic summary vectors, and $\mathbf{F} \in \mathbb{R}^{n\times L}$ represents the final predictions. The module employs a loss function of the same form as Equation 5, denoted as $\mathcal{L}_f$.

Model Inference

Redundant View Identification Strategy Based on the view redundancy phenomenon, removing redundant views would make the summary vector more discriminative. However, how to identify redundant views is our primary concern. To overcome this challenge, we design the redundant view identification strategy.

To identify a redundant view, an intuitive approach is to separate the currently processed view from the others, and observe the impact of this operation on the summary vector. Specifically, for a processed view v, we split its original view combination $\mathbf{W}$ into a single-view combination $\mathbf{W}_v^o$ and a remaining-view combination $\mathbf{W}_v^r$:

$$\mathbf{W}_{i,1}^o = 0, \mathbf{W}_{i,2}^o = 0, \ldots, \mathbf{W}_{i,v}^o = 1, \ldots, \mathbf{W}_{i,V}^o = 0 \quad (10)$$

$$\mathbf{W}_{i,1}^r = \mathbf{W}_{i,1}, \ldots, \mathbf{W}_{i,v}^r = 0, \ldots, \mathbf{W}_{i,V}^r = \mathbf{W}_{i,V} \quad (11)$$

Noted that we only process the existing view so $\mathbf{W}_{i,v} \neq 0$.

For i-th sample, the view encoding module can use these combinations to generate different summary vectors, which are the original summary vector $\mathbf{s}_i^a$, the normal summary vector $\mathbf{s}_{i,v}^o = F(E_1(\mathbf{x}_{i,1}), E_2(\mathbf{x}_{i,2}), \ldots, E_V(\mathbf{x}_{i,V}), \mathbf{W}_v^o)$ and the remained summary vector $\mathbf{s}_{i,v}^r = F(E_1(\mathbf{x}_{i,1}), E_2(\mathbf{x}_{i,2}), \ldots, E_V(\mathbf{x}_{i,V}), \mathbf{W}_v^r)$.

Now, we need to assess the impact of the separation. In the view encoding module, we acquire pseudo-labels $\hat{\mathbf{Y}} \in \mathbb{R}^{n\times L}$ and a classifier C_p. Given that classifier can recognize the category information of the summary vectors, it serves as one of the most intuitive tools for measuring the impact. Consequently, we design a new metric for evaluating it, termed the fidelity score:

$$\mathcal{F}(\mathbf{s}, \hat{\mathbf{y}}) = precision(C_p(\mathbf{s}), \hat{\mathbf{y}}) \tag{12}$$

where $precision(\cdot, \cdot)$ is the multi-label classification precision function. In a dataset, when $\mathcal{F}(\mathbf{s}^o_{i,v}, \hat{\mathbf{y}}^i)$ and $\mathcal{F}(\mathbf{s}^r_{i,v}, \hat{\mathbf{y}}^i)$ of v-th view of i-th sample are significantly greater than the average score within the dataset, it implies the combination does not introduce additional information, suggesting the original view combination can be disassembled.

However, this consideration overlooks the varying difficulty levels of each sample. Due to the long-tail effect in datasets, not all labels appear with equal frequency, facilitating easier predictions for the model on certain labels, allowing it to match most pseudo-labels even with the loss of view structure information. The synthetic summary vector generated by the label-guided diffusion module, containing full-view information of the label along with random noise, serves ideally as a reference for assessing the sample's difficulty level. For the pseudo-label of j-th category of i-th sample $\hat{y} \in \hat{\mathcal{Y}} = \{j|\hat{\mathbf{y}}^i_j = 1\}$, its reverse denoising process can be reformulated through Equation 2 as:

$$\mu_\theta(\mathbf{s}, t, \hat{y}) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{s}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{s}_t, t, \hat{y})) \tag{13}$$

$$\mathbf{s}_{t-1} = \mu_\theta(\mathbf{s}, t, \hat{y}) + \sqrt{\Sigma_\theta(\mathbf{s}_t, t)}\mathbf{m}, \;\; \mathbf{m} \sim \mathcal{N}(0, \mathbf{I}) \tag{14}$$

We define $\mathbf{s}_0$ generated under the guidance of $\hat{y}$ as $\mathbf{s}_{\hat{y}}$, then the synthetic summary vector of i-th sample can be calculated in the following:

$$\mathbf{s}^d_i = \frac{1}{|\hat{\mathcal{Y}}|}\sum_{\hat{y}\in\hat{\mathcal{Y}}} \mathbf{s}_{\hat{y}} \tag{15}$$

When $\mathcal{F}(\mathbf{s}^d_i, \hat{\mathbf{y}}^i)$ of i-th sample is significantly lower than the average score in the dataset, it indicates that the sample is particularly challenging for the model. Thus, we identify redundant views in the following:

$$\mathcal{R}_v = \{i|(\mathcal{F}(\mathbf{s}^o_{i,v}, \hat{\mathbf{y}}^i) > \mu_o + k_{sig}\sigma_o) \wedge (\mathcal{F}(\mathbf{s}^r_{i,v}, \hat{\mathbf{y}}^i) > \mu_r + k_{sig}\sigma_r) \wedge (\mathcal{F}(\mathbf{s}^d_i, \hat{\mathbf{y}}^i) < \mu_d - k_{sig}\sigma_d)\} \tag{16}$$

where (μ_o, σ_o), (μ_r, σ_r), (μ_d, σ_d) denotes the mean value and variance of fidelity score of the normal summary vectors, remained summary vectors and synthetic summary vectors, k_{sig} denotes the variance multiplier, $\mathcal{R}_v$ is the set of sample indices for which view v is considered redundant. The module will remove the views of the samples corresponding to the indices in the set $\mathcal{R}_v$ and replace $\mathbf{s}^a$ and $\mathbf{W}$ with $\mathbf{s}^r$ and $\mathbf{W}^r_v$. This process will continuously repeat in sequence from view 1 to view V, with each completion of a loop making k_{sig} increasing by a fixed value. The loop ends when $\mathcal{R}_v$ becomes an empty set.

Final Prediction At last, we input $\mathbf{s}^a_i$ and $\mathbf{s}^d_i$ into the final prediction module to obtain the final prediction $\mathbf{f}_i \in \mathbb{R}^L$:

$$\mathbf{f}_i = C_f(l_{ori}\mathbf{s}^a_i + l_{syn}\mathbf{s}^d_i) \tag{17}$$

Experiment

Experimental Setup

Datasets Our research employs the methodology suggested by (Tan et al. 2018) to assess the effectiveness of DiffSumm. The datasets included in this evaluation are as follows: **Corel 5k** (Duygulu et al. 2002)-Recognized as a benchmark in the field of image annotation and search, it comprises 5000 images labeled with 260 tags. **Pascal07** (Everingham et al. 2010)-Known as PASCAL VOC 2007, Pascal07 is a prominent dataset in the domain of visual object categorization, containing 9963 images and 20 tags. **Espgame** (von Ahn and Dabbish 2004)-The Espgame dataset originates from an online game where participants tag images, and includes 20770 images annotated with 268 tags. **IAPRTC12** (Grubinger et al. 2006)-The IAPRTC12 dataset, designed for global image annotation challenges, includes around 20,000 images across 290 categories. **Mirflickr** (Huiskes and Lew 2008)-Comprising 25,000 images and 38 tags, featuring a diverse range of themes and subjects. We convert images into six different views, namely GIST, HSV, HUE, LAB, RGB, and SIFT, each providing a unique insight into the visual content by highlighting different visual characteristics.

Evaluation Metrics We utilize metrics Average Precision (AP), Ranking Loss (RL), Hamming Loss (HL) and the Area Under the Curve (AUC) for our evaluations. To simplify comparisons, we use the inverse of HL (1-HL) and RL (1-RL). A higher score signifies better performance.

Implementation Details In terms of model architecture, we employ 2 multilayer perceptrons with four fully connected layers as the encoder and decoder for the autoencoder. For the diffusion model, we utilize a UNet (Ronneberger, Fischer, and Brox 2015) with a cross-attention mechanism, akin to that described in (Rink et al. 2021), comprising five convolutional layers. To simulate a scenario of incompleteness, we randomly remove 50% of the views from the data in each dataset. Within every category, we randomly mark 50% of both the positive and negative tags as unavailable. We set the masking rate k_{mask} to 0.25 across all experiments. To validate the reliability of our findings, we conduct each experiment 10 times and calculate the average and variance of the results.

Compared Methods We compare our approach against ten benchmark methods. Among these, detailed discussions of eight methods - lrMMC, MVL-IV, iMSF, iMvWL, NAIML, DDINet, DICNet, and LMVCAT - can be found in Related Work Section. Notably, iMvWL and NAIML are designed for scenarios involving incomplete views and labels, necessitating adjustments to the other methods to ensure consistency. Inspired by (Li and Chen 2022), we compensate for missing views in lrMMC by using the mean values

Dataset	Metric	lrMMC	MVL-IV	iMSF	iMvWL	NAIML	DDINet	DICNet	LMVCAT	DiffOnly	DiffSumm
Corel5k	AP	24.0±0.2	24.0±0.1	18.9±0.2	28.3±0.7	30.9±0.4	36.4±0.1	38.1±0.4	*38.4±0.4*	38.2±0.2	**45.7±0.1**
	1-RL	76.2±0.2	75.6±0.1	70.9±0.5	86.5±0.3	87.8±0.2	87.1±0.0	*88.2±0.4*	88.0±0.2	87.8±0.1	**89.3±0.1**
	1-HL	95.4±0.0	95.4±0.0	94.3±0.0	97.8±0.0	*98.7±0.0*	*98.7±0.0*	**98.8±0.0**	98.6±0.0	**98.8±0.0**	**98.8±0.0**
	AUC	76.3±0.2	76.2±0.1	66.3±0.5	86.8±0.3	88.1±0.2	87.5±0.1	*88.4±0.4*	88.3±0.2	88.2±0.2	**90.2±0.1**
Pascal07	AP	42.5±0.3	43.3±0.2	32.5±0.0	44.1±1.7	48.8±0.3	*53.6±0.2*	50.5±1.2	51.9±0.5	50.0±0.2	**56.1±0.1**
	1-RL	69.8±0.3	70.2±0.1	56.8±0.0	73.7±0.9	78.3±0.1	80.7±0.1	78.3±0.8	*81.1±0.4*	78.9±0.5	**84.5±0.1**
	1-HL	88.2±0.0	88.3±0.0	83.6±0.0	88.2±0.0	92.8±0.1	93.2±0.0	92.9±0.1	*99.7±0.0*	*99.7±0.0*	**99.8±0.0**
	AUC	72.8±0.2	73.0±0.1	68.6±0.5	78.6±0.3	76.7±1.2	82.7±0.0	*87.6±0.2*	83.4±0.4	87.4±0.2	**88.6±0.1**
Espgame	AP	18.8±0.0	18.9±0.0	10.8±0.0	24.2±0.3	24.6±0.2	28.3±0.1	29.7±0.2	29.4±0.4	*30.6±0.2*	**36.3±0.2**
	1-RL	77.7±0.1	77.8±0.0	72.2±0.2	80.7±0.1	81.8±0.2	81.5±0.0	83.2±0.1	82.8±0.2	*83.4±0.1*	**87.8±0.1**
	1-HL	97.0±0.0	97.0±0.0	96.4±0.0	97.2±0.0	*98.3±0.0*	*98.3±0.0*	*98.3±0.0*	98.1±0.0	98.2±0.0	**98.6±0.0**
	AUC	78.3±0.1	78.4±0.1	67.4±0.3	81.3±0.2	82.4±0.2	82.0±0.1	*83.2±0.1*	82.8±0.2	82.8±0.1	**84.5±0.1**
IAPRTC12	AP	19.7±0.0	19.8±0.0	10.1±0.0	23.5±0.4	26.1±0.1	30.3±0.2	32.3±0.1	31.7±0.3	*35.2±0.2*	**37.9±0.1**
	1-RL	80.1±0.0	79.9±0.1	63.1±0.0	83.3±0.3	84.8±0.1	85.3±0.1	87.3±0.1	87.0±0.1	*88.2±0.1*	**89.0±0.2**
	1-HL	96.7±0.0	96.7±0.0	96.0±0.0	96.9±0.0	*98.1±0.0*	98.0±0.0	*98.1±0.0*	97.9±0.0	98.0±0.0	**98.2±0.0**
	AUC	80.5±0.0	80.4±0.1	66.5±0.1	83.6±0.2	85.0±0.1	85.4±0.0	*87.4±0.1*	87.2±0.1	87.3±0.1	**88.4±0.1**
Mirflickr	AP	44.1±0.1	44.9±0.1	32.3±0.0	49.5±1.2	55.1±0.2	59.8±0.2	58.9±0.5	59.4±0.5	*60.5±0.4*	**64.6±0.2**
	1-RL	80.5±0.0	80.4±0.1	66.5±0.1	83.6±0.2	85.0±0.1	86.3±0.0	86.3±0.4	*86.5±0.3*	86.4±0.2	**89.2±0.3**
	1-HL	83.9±0.0	83.9±0.0	77.5±0.0	84.0±0.3	88.2±0.1	*88.8±0.0*	88.8±0.2	87.6±0.2	*88.8±0.0*	**89.0±0.1**
	AUC	80.6±0.1	80.7±0.0	76.1±0.1	79.4±1.5	83.7±0.1	85.2±0.1	84.9±0.4	*85.3±0.3*	84.6±0.2	**86.9±0.2**

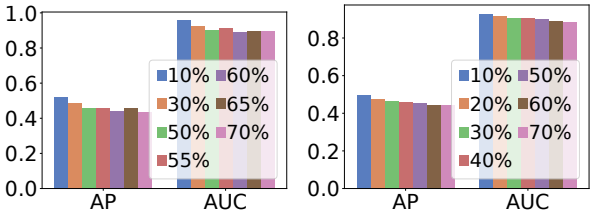
Table 1: The results of various methods on five datasets with 50% missing instances and missing labels. The best results are highlighted in bold. The second-best result is indicated in italics. For simplicity, percent sign is omitted in the following section.

of available views. In the case of MVL-IV and iMSF, missing labels are assumed to be negative. Moreover, we design a method that solely employs pseudo-labels as condition to guide the diffusion model in generating synthetic summary vectors for prediction, named DiffOnly. We follow the recommendations for parameter settings from their published works or available codebases to ensure a fair comparison.

Experimental Results

In Table 1, we present the experimental results of these methods across all datasets with 70% training samples, 50% missing views and missing labels. From the table, it can be observed that DiffSumm surpasses any other method on all metrics across all datasets. Additionally, deep learning models, including DDINet, DICNet, and LMVCAT, outperform the first five methods, it suggests that deep learning holds considerable potential for addressing this issue. Furthermore, DiffOnly achieves the second-best on some datasets, demonstrating that synthetic summary vectors indeed provide additional useful information for classifiers.

Figure 2 illustrates the model's performance under various view and label missing ratios. As seen from Figure 2a, within a specific range of missing view ratios, an increase in the ratio does not lead to a decline in model performance. This result can be elucidated by the redundant view phenomenon we discover. This revelation also suggests that masking some views during training could potentially augment the model's performance. From Figure 2b, it is observed that as the missing label ratio increases, the model's performance consistently declines, indicating that more label information aids the model's learning process. This is possibly because labels have defined categories, and noise is less likely to cause significant disruption to this information.



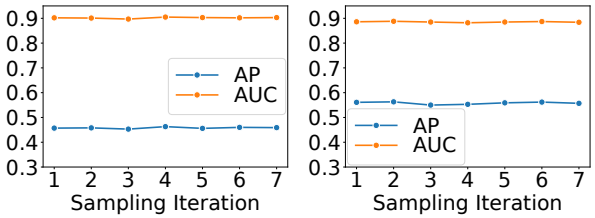
(a) Missing View Ratios (b) Missing Label Ratios

Figure 2: The results for the Corel5k dataset under two scenarios: (a) different missing view ratios while maintaining 50% missing labels ratio, and (b) a fixed 50% missing views ratio combined with different missing label ratios.

Parameter Analysis

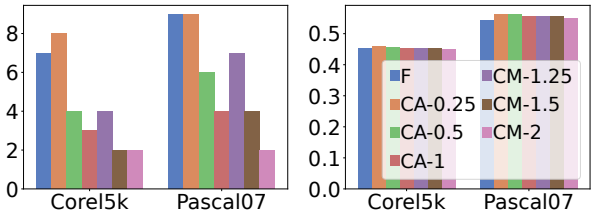
We evaluate two hyperparameters in our model: the number of diffusion model sampling iterations and the variance multiplier k_{sig}, with results depicted in the figures. When we adopt the strategy of multiple sampling iterations of the diffusion model and average the outcomes of all samples, the results can be obtained as illustrated in Figure 3. It can be observed that there appears to be no direct correlation between the number of sampling iterations and the model's performance. This may be attributed to the fact that with a limited number of sampling iterations, the random noise inherent in the diffusion model cannot be effectively eliminated on such a small scale. Moreover, our framework also incorporates an averaging operation, which can, to a certain extent, substitute for this process.

For k_{sig}, we design various methods for evaluation, namely fixed value, cumulative addition, and cumulative



(a) Corel5k (b) Pascal07

Figure 3: The results of sampling with varying numbers of sampling iteration across different datasets.



(a) Loop Ending Rounds (b) AP

Figure 4: The results of employing different methods for determining k_{sig} values across various datasets.

multiplication. The fixed value (F) method entails maintaining the same value of k_{sig} across different rounds; cumulative addition (CA-x) method means that k_{sig} is increased by a fixed amount x each round; cumulative multiplication (CM-x) involves k_{sig} being multiplied by a fixed value x each round. Experimentally, the fixed value method sets k_{sig} to 1 for rounds 1-3, 2 for rounds 4-6, and 3 for rounds beyond 6; cumulative addition is tested with k_{sig} starting at 1 and increasing by 0.25, 0.5, 1, in each round respectively; cumulative multiplication started with k_{sig} at 1, with each round multiplying by 1.25, 1.5, 2, respectively. In the evaluation, given the slow generation process of the label-guided diffusion module, the number of loop ending rounds should be considered, with the comprehensive results illustrated in Figure 4. It can be observed that the fixed value method achieves an average accuracy, but its practicality is limited due to the need for adjusting the fixed value according to the number of rounds across different datasets. The cumulative addition method exhibits relatively high accuracy, but requires a larger number of iterations to end the loop, leading to greater computational resource consumption. The cumulative multiplication method increases values rapidly, resulting in fewer rounds needed to end the loop, but it achieves lower accuracy. This could be due to the larger values in each round, potentially missing some redundant views that could have been identified. Considering both accuracy and computation cost, we ultimately selected the method of adding 0.5 per round as the approach for setting k_{sig}.

Ablation Study

For the identification of redundant views, we design the following variants: **based on statistical regularities** (SR) — by recording the performance of different view combinations during training, the probability of certain combinations forming redundant views can be derived. Then, during inference, if such a combination appears, it is identified as redundant and removed according to the probability; **based on the similarity of summary vectors** (SoSV) — calculating the cosine similarity between summary vectors as an alternative to fidelity scores, then identifying redundant views; **not using synthetic summary vectors as a reference** (NSSV) — omitting the condition of synthetic summary vector and pseudo-label fidelity scores significantly lower than a certain value. We record the performance of these variants and our method in Table 2. In the table, NSSV result in the worst performance, validating our view that assessing sample difficulty is necessary. SoSV is the second best, proving that fidelity scores to some extent equate to similarity. However, fidelity scores, with their clearer semantics and more explicit information, perform better than using similarity.

Method	Corel5k		Espgame		IAPRTC12	
	AP	AUC	AP	AUC	AP	AUC
SR	40.7	89.8	31.9	83.5	34.5	87.1
SoSV	45.0	90.0	35.9	84.4	37.5	88.2
NSSV	37.5	87.6	29.8	83.6	32.2	87.5
DiffSumm	45.7	90.2	36.3	84.5	37.9	88.4

Table 2: Results of different redundant view identification variants.

Method	Corel5k		Espgame		IAPRTC12	
	AP	AUC	AP	AUC	AP	AUC
V	36.7	87.6	28.6	82.1	30.8	85.2
V+L (DiffOnly)	38.2	88.2	30.6	82.8	35.2	87.3
V+L+R+F (DiffSumm)	45.7	90.2	36.3	84.5	37.9	88.4

Table 3: Results of the effectiveness of each module.

Subsequently, we test the effectiveness of each module, including the view encoding module (V), the label-guided diffusion module (L), the redundant view identification strategy (R) and final prediction module (F). We conduct tests on three datasets, with results shown in Table 3. It can be seen that every module within the framework plays a crucial role.

Conclusion

In this paper, we explore the utilization of diffusion models to assist in addressing the incomplete multi-view multi-label classification problem and the view redundancy phenomenon. We propose the DiffSumm framework, which encompasses the view encoding module for obtaining a summary vector and making pseudo-labels based on this vector, a label-guided diffusion module for generating synthetic summary vectors, and a strategy for identifying and removing redundant views. Experimental evidence demonstrates that our proposed DiffSumm is effective in resolving this problem, and that the diffusion model is capable of efficiently generating discriminative summary vectors.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62422202, 62272058, U23A20319, 62192784), and Beijing Natural Science Foundation (4242027). Xuyun Zhang, Amin Beheshti, and Yuankai Qi are not supported by the above mentioned funds.

References

Chen, M.; Liu, T.; Wang, C.; Huang, D.; and Lai, J. 2022. Adaptively-weighted Integral Space for Fast Multiview Clustering. In *MM '22*, 3774–3782.

Duygulu, P.; Barnard, K.; de Freitas, J. F. G.; and Forsyth, D. A. 2002. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In *ECCV 2002*, volume 2353, 97–112.

Everingham, M.; Gool, L. V.; Williams, C. K. I.; Winn, J. M.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.*, 88: 303–338.

Fang, Z.; and Zhang, Z. 2012. Simultaneously combining multi-view multi-label learning with maximum margin classification. In *ICDM 2012*, 864–869.

Grubinger, M.; Clough, P.; Müller, H.; and Deselaers, T. 2006. The IAPR TC12 Benchmark: A New Evaluation Resource for Visual Information Systems. *Workshop Ontoimage*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *NeurIPS 2020*.

Ho, J.; and Salimans, T. 2022. Classifier-Free Diffusion Guidance. *CoRR*, abs/2207.12598.

Huiskes, M. J.; and Lew, M. S. 2008. The MIR flickr retrieval evaluation. In *ACM SIGMM MIR 2008*, 39–43.

Li, Q.; Luo, T.; Jiang, M.; Liao, J.; and Jiang, Z. 2024. Deep Incomplete Multi-View Network Semi-Supervised Multi-Label Learning with Unbiased Loss. In *MM 2024*, 9048–9056.

Li, X.; and Chen, S. 2022. A Concise Yet Effective Model for Non-Aligned Incomplete Multi-View and Missing Multi-Label Learning. *IEEE TPAMI*, 44: 5918–5932.

Liu, C.; Wen, J.; Luo, X.; Huang, C.; Wu, Z.; and Xu, Y. 2023a. DICNet: Deep Instance-Level Contrastive Network for Double Incomplete Multi-View Multi-Label Classification. In *AAAI 2023*, 8807–8815.

Liu, C.; Wen, J.; Luo, X.; and Xu, Y. 2023b. Incomplete Multi-View Multi-Label Learning via Label-Guided Masked View- and Category-Aware Transformers. In *AAAI 2023*, 8816–8824.

Liu, M.; Luo, Y.; Tao, D.; Xu, C.; and Wen, Y. 2015. Low-Rank Multi-View Learning in Matrix Completion for Multi-Label Image Classification. In *AAAI 2015*, 2778–2784.

Nichol, A. Q.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *ICML 2022*, volume 162, 16784–16804.

Ou, S.; Xue, Z.; Li, Y.; Liang, M.; Cai, Y.; and Wu, J. 2024. View-Category Interactive Sharing Transformer for Incomplete Multi-View Multi-Label Learning. In *CVPR 2024*, 27457–27466.

Rink, N. A.; Paszke, A.; Vytiniotis, D.; and Schmid, G. S. 2021. Memory-efficient array redistribution through portable collective communication. *CoRR*, abs/2112.01075.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR 2022*, 10674–10685.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI 2015*, volume 9351, 234–241.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, S. K. S.; Lopes, R. G.; Ayan, B. K.; Salimans, T.; Ho, J.; Fleet, D. J.; and Norouzi, M. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *NeurIPS 2022*.

Sohl-Dickstein, J.; Weiss, E. A.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *ICML 2015*, volume 37, 2256–2265.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *ICLR 2021*.

Sun, Y.; Xu, Q.; Wang, Z.; Yang, Z.; and He, J. 2024. EDGE: Unknown-aware Multi-label Learning by Energy Distribution Gap Expansion. arXiv:2412.07499.

Tan, Q.; Yu, G.; Domeniconi, C.; Wang, J.; and Zhang, Z. 2018. Incomplete Multi-View Weak-Label Learning. In *IJCAI 2018*, 2703–2709.

Tang, H.; and Liu, Y. 2022. Deep Safe Multi-view Clustering: Reducing the Risk of Clustering Performance Degradation Caused by View Increase. In *CVPR 2022*, 202–211.

Tao, H.; Hou, C.; Liu, X.; Liu, T.; Yi, D.; and Zhu, J. 2018. Reliable Multi-View Clustering. In *AAAI-18*, 4123–4130.

Trosten, D. J.; Løkse, S.; Jenssen, R.; and Kampffmeyer, M. 2021. Reconsidering Representation Alignment for Multi-View Clustering. In *CVPR 2021*, 1255–1265.

von Ahn, L.; and Dabbish, L. 2004. Labeling images with a computer game. In *CHI 2004*, 319–326.

Wang, Q.; Ding, Z.; Tao, Z.; Gao, Q.; and Fu, Y. 2021. Generative partial multi-view clustering with adaptive fusion and cycle consistency. *IEEE TIP*, 30: 1771–1783.

Wang, Q.; Tao, Z.; Gao, Q.; and Jiao, L. 2022. Multi-View Subspace Clustering via Structured Multi-Pathway Network. *IEEE TNNLS*.

Wen, J.; Liu, C.; Deng, S.; Liu, Y.; Fei, L.; Yan, K.; and Xu, Y. 2023. Deep Double Incomplete Multi-View Multi-Label Learning With Incomplete Labels and Missing Views. *IEEE TNNLS*.

Wen, J.; Zhang, Z.; Zhang, Z.; Wu, Z.; Fei, L.; Xu, Y.; and Zhang, B. 2020. DIMC-net: Deep Incomplete Multi-view Clustering Network. In *MM '20*, 3753–3761.

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view learning with incomplete views. *IEEE TIP*, 24: 5812–5825.

Yuan, L.; Wang, Y.; Thompson, P. M.; Narayan, V. A.; and Ye, J. 2012. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *NeuroImage*, 61: 622–632.

Yue, Z.; Ye, F.; Zhang, Y.; Liang, C.; and Tsang, I. W. 2021. Deep Safe Multi-Task Learning. *arXiv preprint arXiv:2111.10601*.

Zhang, W.; Zhang, K.; Gu, P.; and Xue, X. 2013a. Multi-View Embedding Learning for Incompletely Labeled Data. In *IJCAI 2013*, 1910–1916.

Zhang, Y.; Zhang, H.; Nasrabadi, N. M.; and Huang, T. S. 2013b. Multi-metric learning for multi-sensor fusion based classification. *Inf. Fusion*, 14: 431–440.