# Multi-View Multi-Label Classification via View-Label Matching Selection

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

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

## Abstract

In multi-view multi-label classification (MVML), each object is described by several heterogeneous views while annotated with multiple related labels. The key to learn from such complicate data lies in how to fuse cross-view features and explore multi-label correlations, while accordingly obtain correct assignments between each object and its corresponding labels. In this paper, we proposed an advanced MVML method named VAMS, which treats each object as a bag of views and reformulates the task of MVML as a "view-label" matching selection problem. Specifically, we first construct an object graph and a label graph respectively. In the object graph, nodes represent the multi-view representation of an object, and each view node is connected to its K-nearest neighbor within its own view. In the label graph, nodes represent the semantic representation of a label. Then, we connect each view node with all labels to generate the unified "view-label" matching graph. Afterwards, a graph network block is introduced to aggregate and update all nodes and edges on the matching graph, and further generating a structural representation that fuses multi-view heterogeneity and multi-label correlations for each view and label. Finally, we derive a prediction score for each view-label matching and select the optimal matching via optimizing a weighted cross-entropy loss. Extensive results on various datasets have verified that our proposed VAMS can achieve superior or comparable performance against state-of-the-art methods.

## Introduction

Multi-view multi-label classification (MVML) is a crucial task in the field of machine learning and data mining, which aims to extract both consensus and complementary information from multiple high-dimensional heterogeneous views and assign multiple semantically relevant labels to the given samples (Liu et al. 2023b; Li et al. 2024). For example, in the task of news classification (Figure 1), multiple sources of information (image, video, text) are integrated to provide multiple semantically relevant labels such as disaster, wildfire, and rescue. MVML provides an effective framework to learn a desired multi-label classifier for bridging these various information (features) to the diverse topics (labels) and further provides support for subsequent tasks, such as public opinion analysis and monitoring on social media platforms.


Figure 1: An application of MVML in news classification.

The key to learn from MVML data lies in how to effectively fuse these heterogeneous features while comprehensively characterizing all relevant labels. Based on different multi-view fusion strategies, existing MVML methods can be roughly divided into two categories: Feature-fusion strategy and Decision-fusion strategy. Feature-fusion strategy based methods (Liu et al. 2015; Zhang, Jia, and Li 2020) usually conduct multi-view feature fusion first to obtain a common feature representation and then they employ the common representation to learn final multi-label classifier. Decision-fusion strategy (Luo et al. 2015) based methods directly learn multiple multi-label base classifiers for different views, and they make the final prediction by averaging the results of these base classifiers. Intuitively, Feature-fusion strategy pays more attention to cross-view consensus information, which tends to correspond to significant labels, while some rare or insignificant labels may be overwhelmed. Decision-fusion strategy focuses more on the individual-view specificity information, which tends to correspond to view-specific labels, where some labels with difficult feature representations can not be easily detected. Recently, some MVML methods (Tan et al. 2021; Wu et al. 2019; Lyu et al. 2022) attempt to simultaneously take both consensus and specific information into consideration, which intend to comprehensively characterize all relevant labels. However, most of these methods extract cross-view consensus features and individual-view specificity features in a sep-

*indicates corresponding author.

arate manner and they independently learn a common classifier and multiple view-specific classifier to make the final prediction. Basically, these methods still split up the connection of exploiting consistency and specificity in multi-view data, and meanwhile, they also have not well described the inherent view-label correspondence relationship, which inevitably leads the final prediction model to be sub-optimal.

In order to address these issues, in this paper, we propose a matching-based MVML method named VAMS, which integrates cross-view consensus information and individual-view specificity information into a unified framework and directly constructs the explicit matching correspondences between each view and label. Specifically, we first construct an object graph by connecting different view representations for fusing cross-view consistency and individual-view specificity, where the neighbor information of each view node is incorporated as supplementary information to enhance its feature expressiveness. Next, a label graph is established to capture multi-label semantic correlations, and all label nodes are connected to each view node to form the unified "view-label" matching graph. Afterwards, a graph network block (GN Block) is employed to perform node interactions among all views and labels, where cross-view feature consistency, view-specific feature specificity, and multi-label semantic correlations are simultaneously incorporated by a graph convolution propagation mechanism, thereby obtaining structural representation for each view and label. Finally, we derive prediction confidence of each view-label matching by optimizing the output of our model with a weighted cross-entropy loss. In summary, the contributions of our paper lie in the following aspects:

- We propose an advanced MVML method named VAMS, which leverages "view-label" matching to jointly fuse cross-view consensus, individual-view specificity, and multi-label correlation into the whole learning process, finally achieving accurate MVML prediction.
- To the best of our knowledge, it is the first time to incorporate matching selection mechanism into MVML task, which avoids the split of consistency and specificity exploration in previous MVML methods, and accordingly improving the performance of the prediction model.
- Enormous experimental results as well as comprehensive experimental analysis on various datasets have demonstrated that our proposed VAMS can achieve superior performance against state-of-the-art methods.

## Related Work

### Multi-View Learning (MVL)

Multi-View Learning aims to learn from different feature spaces to enhance model performance. Existing MVL methods can be roughly categorized into the following types: (Zhang et al. 2018) proposed a type of co-training method, which considers both view-specific and shared representations. (Liu et al. 2023a) proposed a multi-kernel learning, where contrastive learning is employed to leverage complementary information from data for high-quality kernel computation and combines kernels to enhance learning performance. (Zhang et al. 2023a; Luo et al. 2018) proposed subspace learning methods, which combine view consistency and specificity for effective subspace representation learning in multi-view clustering problems. Besides, there are also many other MVL methods for different tasks, such as clustering (Wang et al. 2023; Gu et al. 2023; Zhang et al. 2023b), retrieval (Dong et al. 2024) and classification (Jiang et al. 2021; Wen et al. 2024; Tan et al. 2024), etc.

### Multi-Label Learning (MLL)

Multi-Label Learning focuses on learning from data with multiple labels, and existing MLL methods can be broadly categorized into traditional methods and deep learning-based methods. 1) Traditional methods for handling MLL problems include problem transformation-based methods, which transfer multi-label problems into single-label learning, such as BR (Tsoumakas and Ioannis Katakis 2007), Classifier Chains (Liu, Tsang, and Müller 2017) and algorithm adaption-based methods, which convert the task of multi-label classification to some well-established learning scenarios, including CAMEL (Feng, An, and He 2019), RMFL (Feng et al. 2022), Metric Learning (Zhang et al. 2024; Liu et al. 2019) etc. 2) deep learning-based MLL methods tend to utilize deep neural networks to automatically learn complex relationships among labels. The most frequently used deep learning based methods for MLL include deep neural networks (Zhao et al. 2024, 2022; Wei et al. 2023; Yang et al. 2023), convolution (Wu et al. 2021; Feng et al. 2020), and transformer (Lyu et al. 2024b), etc.

### Multi-View Multi-Label Learning (MVML)

Multi-View Multi-Label Learning combines the characteristics of MVL and MLL, making it more complex when dealing with multi-view data (Zhong, Lyu, and Yang 2024; Lyu et al. 2022). To learn from such complicated data, the main methods include Feature-fusion strategy and Decision-fusion strategy. Feature-fusion strategy typically obtains a unified feature representation by fusing multi-view features, which is then used to train the final multi-label classifier (Liu et al. 2015). Decision-fusion strategy trains multiple multi-label base classifiers for different views and makes the final prediction by averaging the outputs of these base classifiers (Tan et al. 2021). Recently, some methods attempt to simultaneously take both consensus and specific information into consideration, which intend to comprehensively characterize all relevant labels, such as (Lyu et al. 2024a).

## The Proposed Method

Formally speaking, we define $\mathcal{X} = R^{d_1} \times R^{d_2} \times \cdots \times R^{d_V}$ as the feature space with $V$ views, where each view has $d_m$ dimension. For a given dataset $D = \{(\boldsymbol{X}_i, \boldsymbol{y}_i) | 1 \leq i \leq N\}$, we denote each object $\boldsymbol{X}_i$ consists of $V$ feature vectors $[\boldsymbol{x}_i^1; \boldsymbol{x}_i^2; ...; \boldsymbol{x}_i^V]$, $\boldsymbol{y}_i = [y_i^1, y_i^2, ..., y_i^Q]$ as the ground-truth label for $\boldsymbol{X}_i$, where $Q$ is the total number of labels in the dataset. $y_i^c = 1 (1 \leq c \leq Q)$ indicates that object $\boldsymbol{X}_i$ is annotated with label $c$, $y_i^c = 0$ otherwise. Our proposed VAMS method aims to integrate these diverse representations from different views to construct a robust multi-

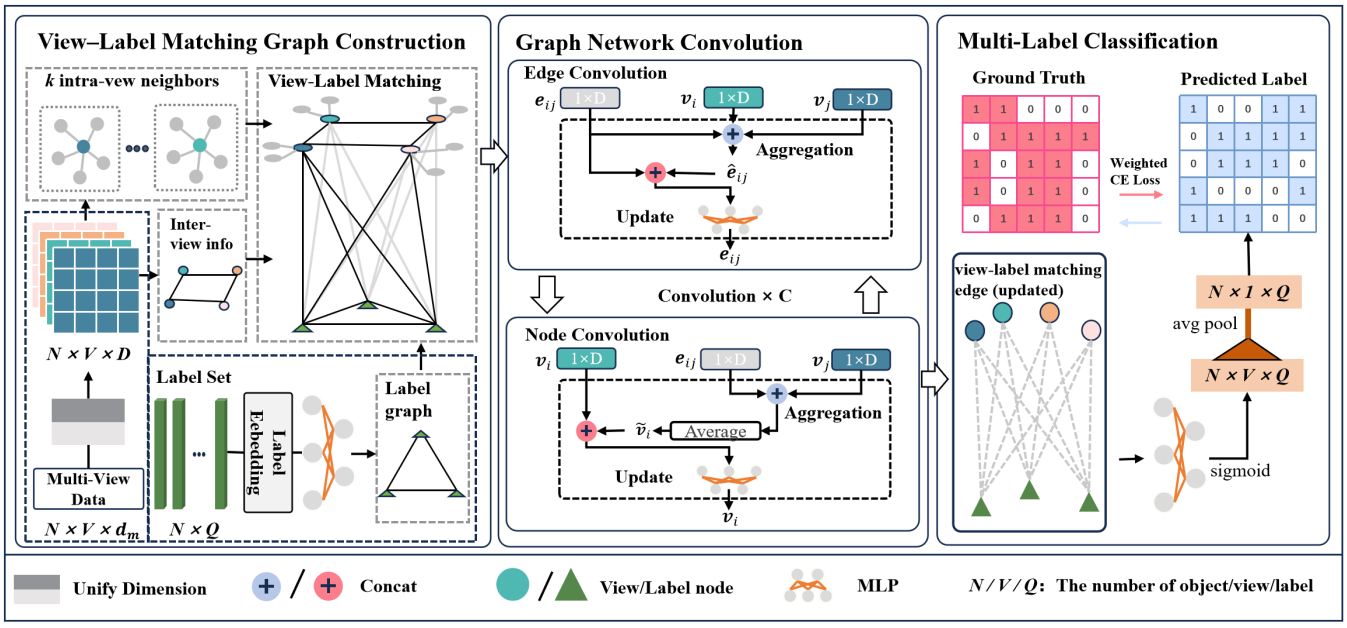



Figure 2: The framework of our proposed VAMS, which consists of three components: (1) View-Label Matching Graph Construction, which connects the inter-view features of an object and links each view node to $k$ intra-view neighbors to form an object graph, while also constructing a fully connected label graph, ultimately connecting each view node to all labels to form a view-label matching; (2) Graph Network Convolution, using a Graph Network Block (GN Block) containing edge convolution and node convolution to perform aggregating and updating; (3) Multi-Label Classification, where a decoder derives view-label matching scores from the updated graph state, and averages them to obtain the final label prediction.

label classifier and further assign the predictive labels for test samples. Figure 2 illustrates the overview architecture of VAMS, which consists of three key components: *View-Label Matching Graph Construction*, *Graph Network Convolution*, and *Multi-Label Classification*.

## View-Label Matching Graph Construction

In order to explicitly characterize the direct view-label correspondence relationship, we construct a unified view-label matching graph $\mathbb{G}^m = (\mathbb{G}^o, \mathbb{G}^l, \mathbb{E}^m)$, which consists of an object graph $\mathbb{G}^o = (\mathbb{V}^o, \mathbb{E}^o)$, a label graph $\mathbb{G}^l = (\mathbb{V}^l, \mathbb{E}^l)$, and their matching edges $\mathbb{E}^m$. Specifically, we first connect $V$ feature representations of an object to construct the object graph $\mathbb{G}^o = (\mathbb{V}^o, \mathbb{E}^o)$, where cross-view feature nodes $\{\boldsymbol{v}_i^o|_{i=1}^V\} \in \mathbb{V}^o$ are integrated into a unified graph to explore the cross-view consensuses. Meanwhile, in order to enhance the individual-view specificities, we introduce the $K$ nearest neighbors $\{\boldsymbol{v}_{i_k}^o|_{k=1}^K\}$ of each view node as complementary information and connect them to their corresponding view node $\boldsymbol{v}_i^o$. Note that $\mathbb{G}^o$ is an undirected graph, where each node $\boldsymbol{v}_i^o \in \mathbb{V}^o$ is represented by the feature vector $\boldsymbol{x}^i$, and each edge $\boldsymbol{e}_{ij}^o \in \mathbb{E}^o$ is described by concatenating the feature vectors of the connected nodes $\boldsymbol{v}_i^o$ and $\boldsymbol{v}_j^o$:

$$\boldsymbol{v}_i^o = \boldsymbol{x}^i, \quad \boldsymbol{e}_{ij}^o = [\boldsymbol{x}^i, \boldsymbol{x}^j], \tag{1}$$

where $[\cdot, \cdot]$ represents the vector concatenation operation.

After obtaining the object graph $\mathbb{G}^o$, we further construct a fully connected label graph $\mathbb{G}^l = (\mathbb{V}^l, \mathbb{E}^l)$ to capture the label correlations, where each label node $\{\boldsymbol{v}_i^l|_{i=1}^Q\} \in \mathbb{V}^l$ is represented by one-hot embedding $\boldsymbol{c}^i$ of the $i$-th class. For the edge $\boldsymbol{e}_{ij}^l \in E^l$, we employ the same strategy in object graph to concatenate the embedding vectors of the two connected label nodes $\boldsymbol{v}_i^l$ and $\boldsymbol{v}_j^l$, forming its edges attributes:

$$\boldsymbol{v}_i^l = \boldsymbol{c}^i, \quad \boldsymbol{e}_{ij}^l = [\boldsymbol{c}^i, \boldsymbol{c}^j], \tag{2}$$

Finally, to establish the full matching correspondence relationship between views and labels, we connect each view node $\boldsymbol{v}_i^o$ in object graph $\mathbb{G}^o$ with each label node $\boldsymbol{v}_i^l$ in label graph $\mathbb{G}^l$ to generate the unified View-Label Matching Graph $\mathbb{G}^m = (\mathbb{G}^o, \mathbb{G}^l, \mathbb{E}^m)$, where the edges $\boldsymbol{e}_{ij}^m \in \mathbb{E}^m$ connecting view nodes and labels remain undirected, and their attributes are generated by the feature concatenation of the connected nodes:

$$\boldsymbol{e}_{ij}^m = [\boldsymbol{v}_i^o, \boldsymbol{v}_j^l]. \tag{3}$$

According to the above operations, we explicitly construct the "view-label" matching correspondence between object and labels, which jointly integrates cross-view consensuses, individual-view specificities, and multi-label correlations into a unified framework. Meanwhile, such unified framework avoids the separation of previous MVML methods in exploiting multi-view consistency and specificity and significantly enhances the comprehensive semantic characterization capability of the final model.

## Graph Network Convolution

To better fuse the above multi-granularity relationships in MVML data and generate more distinctive graph represen-

tations, motivated by (Wang et al. 2020), we introduce graph network block (GN Block) to aggregate and update nodes and edges in the "view-label" matching graph. This module consists of a node convolution layer, which collects the attributes of all the nodes and edges adjacent to each node to compute per-node updates, and an edge convolution layer, which assembles the attributes of the two nodes associated with each edge to generate a new attribute of this edge.

**Node Convolution.** Each node convolution layer consists of a group of aggregation functions, which gather the information from its adjacent nodes and associated edges, and an update function, which updates node attributes according to these gathered information. In our model, there are two different types of nodes that need to be aggregated and updated, including view nodes and label nodes.

Specifically, for a view node $\boldsymbol{v}_i^o$ in $\mathbb{G}^o$, its connected view nodes $\{\boldsymbol{v}_j^o|_{j=1}^{V-1}\}$ via view edges $\boldsymbol{e}_{ij}^o$, neighbor view nodes $\{\boldsymbol{v}_{i_k}^o|_{k=1}^{K}\}$ via neighbor edges $\boldsymbol{e}_{i_k}^o$ and label nodes $\{\boldsymbol{v}_j^l|_{j=1}^{Q}\}$ via matching edges $\boldsymbol{e}_{ij}^m$ are gathered to update its attribute representations, i.e.,

$$\begin{aligned} \bar{\boldsymbol{v}}_i^o =& \frac{1}{V-1}\bar{\rho}_n^o([\boldsymbol{e}_{ij}^o, \boldsymbol{v}_j^o]), \quad \hat{\boldsymbol{v}}_i^o = \frac{1}{K}\hat{\rho}_n^o([\boldsymbol{e}_{i_k}^o, \boldsymbol{v}_{i_k}^o]), \\ \tilde{\boldsymbol{v}}_i^o =& \tilde{\rho}_n^o([\boldsymbol{e}_{ij}^m, \boldsymbol{v}_j^l]), \qquad \boldsymbol{v}_i^o \leftarrow \phi_n^o([\boldsymbol{v}_i^o, \bar{\boldsymbol{v}}_i^o, \hat{\boldsymbol{v}}_i^o, \tilde{\boldsymbol{v}}_i^o]), \end{aligned} \tag{4}$$

where $\bar{\rho}_n^o$, $\hat{\rho}_n^o$, $\tilde{\rho}_n^o$ are aggregation functions, which gather the information from view nodes in $\mathbb{G}^o$ and label nodes in $\mathbb{G}^l$ respectively. $\phi_n^o$ concatenates the current attributes of $\boldsymbol{v}_i^o$ with the gathered information $\bar{\boldsymbol{v}}_i^o$, $\hat{\boldsymbol{v}}_i^o$ and $\tilde{\boldsymbol{v}}_i^o$ to derive the updated attributes for $\boldsymbol{v}_i^o$.

For a label node $\boldsymbol{v}_i^l$ in $\mathbb{G}^l$, its connected label nodes $\{\boldsymbol{v}_j^l|_{j=1}^{Q-1}\}$ via label edges $\boldsymbol{e}_{ij}^l$ and view nodes $\{\boldsymbol{v}_j^o|_{j=1}^{V}\}$ via matching edges $\boldsymbol{e}_{ij}^m$ are gathered to update its attribute representation, i.e.,

$$\bar{\boldsymbol{v}}_i^l = \frac{1}{Q-1}\bar{\rho}_n^l([\boldsymbol{e}_{ij}^l, \boldsymbol{v}_j^l]), \quad \tilde{\boldsymbol{v}}_i^l = \frac{1}{V}\tilde{\rho}_n^l([\boldsymbol{e}_{ij}^m, \boldsymbol{v}_j^o]), \tag{5}$$

$$\boldsymbol{v}_i^l \leftarrow \phi_n^l([\boldsymbol{v}_i^l, \bar{\boldsymbol{v}}_i^l, \tilde{\boldsymbol{v}}_i^l]). \tag{6}$$

where $\bar{\rho}_n^l$ and $\tilde{\rho}_n^l$ are aggregation functions, which gather the information from $\mathbb{G}^o$ and $\mathbb{G}^l$ respectively. $\phi_n^l$ concatenates the current attributes of $\boldsymbol{v}_i^l$ with the gathered information $\bar{\boldsymbol{v}}_i^l$ and $\tilde{\boldsymbol{v}}_i^l$ to derive the updated attributes for $\boldsymbol{v}_i^l$.

**Edge Convolution.** Each edge convolution layer consists of an aggregation function that gathers the information from its associated nodes and an update function that updates node attributes via these gathered information. In our model, three different types of edges are aggregated and updated, including view edges, label edges, and matching edges.

Specifically, for a view edge $\boldsymbol{e}_{ij}^o$, $\boldsymbol{e}_{i_k}^o$ in $\mathbb{G}^o$, its attribute representation is updated by:

$$\begin{aligned} \hat{\boldsymbol{e}}_{ij}^o &= \rho_{\boldsymbol{e}}^o([\boldsymbol{v}_i^o, \boldsymbol{v}_j^o]), \qquad \boldsymbol{e}_{ij}^o \leftarrow \phi_{\boldsymbol{e}}^o([\boldsymbol{e}_{ij}^o, \hat{\boldsymbol{e}}_{ij}^o]), \\ \hat{\boldsymbol{e}}_{i_k}^o &= \rho_{\boldsymbol{e}}^o([\boldsymbol{v}_i^o, \boldsymbol{v}_{i_k}^o]), \qquad \boldsymbol{e}_{i_k}^o \leftarrow \phi_{\boldsymbol{e}}^o([\boldsymbol{e}_{i_k}^o, \hat{\boldsymbol{e}}_{i_k}^o]). \end{aligned} \tag{7}$$

Similarly, for a label edge $\boldsymbol{e}_{ij}^l$ in $\mathbb{G}^l$, its attribute representation is aggregated and updated by:

$$\hat{\boldsymbol{e}}_{ij}^L = \rho_{\boldsymbol{e}}^l([\boldsymbol{v}_i^l, \boldsymbol{v}_j^l]), \quad \boldsymbol{e}_{ij}^l \leftarrow \phi_{\boldsymbol{e}}^l([\boldsymbol{e}_{ij}^l, \hat{\boldsymbol{e}}_{ij}^l]). \tag{8}$$

Algorithm 1: The training process of VAMS

**Input**: Multi-view data: $D = \{(\boldsymbol{X}_i, \boldsymbol{y}_i)|1 \le i \le N\}$, The number of convolutions: $C$, The number of epochs: $I_m$.
**Output**: Prediction Model.
**Process**:

```
Construct view-label matching graph G^m by Eq.(1)-(3);
Intialized the attributes of nodes and edges;
for epoch = 1 to I_m do
    // Forward Propagation
    for conv = 1 to C do
        Conduct node convolution via Eq. (4)-(6);
        Conduct edge convolution via Eq. (7)-(9);
    end for
    Calculate matching score r_vc by Eq. (10);
    Calculate prediction score r_c^i of X_i;
    // Backward Propagation
    Update the model parameters by optimizing Eq. (11);
end for
```

As for the matching edges $\boldsymbol{e}_{ij}^m$ in $\mathbb{E}^m$, its structural attributes can be represented by:

$$\hat{\boldsymbol{e}}_{ij}^m = \rho_{\boldsymbol{e}}^m([\boldsymbol{v}_i^o, \boldsymbol{v}_j^l]), \quad \boldsymbol{e}_{ij}^m \leftarrow \phi_{\boldsymbol{e}}^m([\boldsymbol{e}_{ij}^m, \hat{\boldsymbol{e}}_{ij}^m]). \tag{9}$$

All the aforementioned aggregation and update functions in node convolution layer and edge convolution layer are implemented as MLPs, with different structures and parameters. Additionally, in order to better integrate the consistency and specificity information from different views and further enhance the feature expression of each graph node, we repeat the above convolution operation to fuse more information into each node, then obtain desired structural representations for subsequent multi-label classification.

## Multi-Label Classification

In our model, the task of MVML classification is transferred as "view-label" matching selection problem. Thus, we directly map the attributes of view-label matching edges to the view-label prediction scores for subsequent evaluation, i.e.,

$$r_{vc} = \Phi_e^{dec}(\boldsymbol{e}_{vc}^m). \tag{10}$$

Here, $r_{vc} \in [0,1]^{V\times Q}$ represents the prediction score on $c$-th label in $v$-th view, and $\Phi_e^{dec}$ is an MLP.

Considering the contributions of different views, for each object $\boldsymbol{X}_i$, we calculate its label prediction scores $r_c^i$ by averaging the outputs from all $V$ views, i.e., $r_c^i = \frac{1}{V}\sum_{v=1}^{V} r_{vc}^i$, where $r_{vc}^i$ is the prediction score of $\boldsymbol{X}_i$ on $c$-th label in $v$-th view. Accordingly, we can train the model by optimizing:

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{c=1}^{Q} w_c \left[y_i^c \log\left(r_c^i\right) + \left(1-y_i^c\right)\log\left(1-r_c^i\right)\right], \tag{11}$$

where $w_c = y_i^c \cdot e^{\beta(1-p^c)} + (1-y_i^c)\cdot e^{\beta p^c}$ is employed to alleviate the class imbalance, $\beta$ is hyperparameter and $p^c$ is the ratio of label $c$ in the whole data set. Algorithm 1 illustrates the whole training process of our proposed method.

| H-L | LrMMC | LSPC | SIMM | FIMAN | IMvMLC | ML-BVAE | VAMS |
|---|---|---|---|---|---|---|---|
| Emotions | 0.196±0.011 | 0.251±0.014 | 0.307±0.004 | 0.231±0.013 | 0.330±0.021 | 0.317±0.012 | **0.193±0.019** |
| Plant | 0.115±0.002 | 0.168±0.007 | **0.090±0.001** | 0.238±0.011 | 0.202±0.038 | **0.090±0.001** | **0.090±0.001** |
| scene | **0.082±0.006** | 0.221±0.008 | 0.179±0.002 | 0.195±0.005 | 0.197±0.007 | 0.179±0.001 | 0.086±0.007 |
| Yeast | 0.255±0.020 | 0.297±0.008 | 0.241±0.011 | 0.216±0.004 | 0.313±0.007 | 0.232±0.004 | **0.204±0.007** |
| human | 0.096±0.002 | 0.176±0.003 | 0.085±0.001 | 0.151±0.002 | 0.112±0.006 | 0.085±0.001 | **0.083±0.001** |
| Corel5k | 0.013±0.000 | 0.020±0.000 | 0.013±0.018 | 0.018±0.000 | 0.158±0.008 | 0.013±0.000 | **0.012±0.008** |
| Pascal | 0.073±0.000 | 0.219±0.003 | **0.060±0.001** | 0.116±0.002 | 0.074±0.000 | 0.062±0.001 | 0.110±0.013 |
| R-L | LrMMC | LSPC | SIMM | FIMAN | IMvMLC | ML-BVAE | VAMS |
| Emotions | **0.133±0.016** | 0.185±0.022 | 0.344±0.047 | 0.161±0.026 | 0.183±0.019 | 0.423±0.035 | 0.144±0.013 |
| Plant | 0.371±0.014 | 0.576±0.037 | 0.378±0.025 | 0.277±0.028 | 0.210±0.015 | 0.238±0.012 | **0.184±0.024** |
| scene | 0.115±0.011 | 0.233±0.023 | 0.280±0.026 | 0.107±0.006 | 0.086±0.004 | 0.171±0.017 | **0.070±0.010** |
| Yeast | 0.275±0.011 | 0.530±0.018 | 0.218±0.018 | 0.187±0.005 | 0.180±0.005 | 0.204±0.007 | **0.168±0.004** |
| human | 0.358±0.006 | 0.618±0.018 | 0.261±0.046 | 0.186±0.011 | 0.149±0.007 | 0.181±0.008 | **0.148±0.009** |
| Corel5k | 0.173±0.004 | 0.860±0.005 | 0.160±0.005 | 0.085±0.000 | 0.114±0.003 | 0.188±0.008 | **0.082±0.004** |
| Pascal | 0.336±0.005 | 0.868±0.003 | 0.097±0.006 | 0.118±0.003 | **0.063±0.002** | 0.106±0.001 | 0.101±0.002 |
| Cov | LrMMC | LSPC | SIMM | FIMAN | IMvMLC | ML-BVAE | VAMS |
| Emotions | 2.198±0.094 | 1.905±0.138 | **0.457±0.051** | 7.796±0.189 | 1.873±0.037 | 3.163±0.242 | 1.659±0.055 |
| Plant | 4.256±0.147 | 6.606±0.369 | 4.325±0.270 | 3.216±0.350 | 2.461±0.171 | 2.749±0.141 | **2.184±0.256** |
| scene | 0.677±0.057 | 1.252±0.109 | **0.248±0.020** | 0.628±0.020 | 0.516±0.024 | 0.942±0.090 | 0.434±0.058 |
| Yeast | 10.32±0.195 | 11.27±0.171 | 7.368±0.293 | 6.673±0.074 | 6.538±0.094 | 6.706±0.094 | **6.376±0.096** |
| human | 5.281±0.072 | 8.848±0.276 | 3.797±0.640 | 2.817±0.095 | 2.271±0.099 | 2.666±0.112 | **2.245±0.102** |
| Corel5k | 96.72±1.300 | 257.3±0.499 | 95.99±3.146 | 53.94±0.790 | 70.80±2.256 | 108.7±4.792 | **53.20±2.218** |
| Pascal | 7.900±0.060 | 17.08±0.065 | 2.772±0.140 | 3.486±0.081 | **1.908±0.065** | 2.966±0.028 | 2.231±0.038 |
| A-P | LrMMC | LSPC | SIMM | FIMAN | IMvMLC | ML-BVAE | VAMS |
| Emotions | 0.763±0.020 | 0.773±0.025 | 0.634±0.043 | 0.806±0.027 | 0.782±0.021 | 0.572±0.022 | **0.826±0.012** |
| Plant | 0.464±0.016 | 0.376±0.023 | 0.369±0.029 | 0.492±0.030 | 0.544±0.017 | 0.505±0.020 | **0.585±0.027** |
| scene | 0.852±0.012 | 0.647±0.020 | 0.608±0.027 | 0.827±0.010 | 0.844±0.004 | 0.717±0.023 | **0.878±0.014** |
| Yeast | 0.610±0.013 | 0.554±0.012 | 0.712±0.014 | 0.740±0.007 | 0.738±0.006 | 0.712±0.007 | **0.763±0.009** |
| human | 0.480±0.006 | 0.304±0.021 | 0.495±0.042 | 0.583±0.015 | 0.600±0.010 | 0.536±0.010 | **0.609±0.014** |
| Corel5k | 0.215±0.010 | 0.075±0.004 | 0.292±0.004 | 0.430±0.007 | 0.333±0.008 | 0.286±0.000 | **0.452±0.009** |
| Pascal | 0.422±0.004 | 0.116±0.003 | 0.685±0.010 | 0.721±0.003 | 0.695±0.008 | 0.659±0.002 | **0.759±0.008** |
| Micro-F1 | LrMMC | LSPC | SIMM | FIMAN | IMvMLC | ML-BVAE | VAMS |
| Emotions | 0.685±0.018 | 0.653±0.022 | 0.034±0.051 | 0.671±0.014 | 0.482±0.010 | 0.107±0.091 | **0.692±0.015** |
| Plant | 0.339±0.016 | 0.230±0.025 | 0.004±0.005 | 0.299±0.014 | 0.181±0.006 | 0.146±0.036 | **0.370±0.027** |
| scene | **0.772±0.017** | 0.544±0.019 | 0.001±0.002 | 0.616±0.008 | 0.308±0.002 | 0.304±0.095 | 0.748±0.016 |
| Yeast | 0.516±0.012 | 0.385±0.010 | 0.428±0.042 | 0.111±0.008 | 0.468±0.003 | 0.479±0.008 | **0.656±0.012** |
| human | 0.391±0.009 | 0.212±0.014 | 0.001±0.002 | 0.177±0.012 | 0.159±0.000 | 0.389±0.011 | **0.438±0.017** |
| Corel5k | 0.273±0.009 | 0.153±0.004 | 0.038±0.010 | 0.361±0.009 | 0.029±0.000 | 0.025±0.001 | **0.372±0.005** |
| Pascal | 0.283±0.015 | 0.084±0.003 | 0.343±0.044 | 0.008±0.002 | 0.136±0.001 | 0.385±0.031 | **0.426±0.026** |
| Macro-F1 | LrMMC | LSPC | SIMM | FIMAN | IMvMLC | ML-BVAE | VAMS |
| Emotions | **0.684±0.017** | 0.647±0.023 | 0.032±0.051 | 0.657±0.016 | 0.478±0.012 | 0.052±0.031 | 0.679±0.017 |
| Plant | 0.111±0.007 | 0.105±0.014 | 0.002±0.003 | **0.202±0.018** | 0.160±0.003 | 0.024±0.008 | 0.196±0.026 |
| scene | **0.772±0.017** | 0.522±0.019 | 0.001±0.002 | 0.622±0.008 | 0.307±0.002 | 0.304±0.078 | 0.758±0.014 |
| Yeast | 0.398±0.012 | 0.243±0.005 | 0.133±0.007 | 0.356±0.004 | 0.425±0.003 | 0.122±0.001 | **0.478±0.015** |
| human | 0.157±0.010 | 0.089±0.012 | 0.00±0.001 | 0.166±0.005 | 0.143±0.001 | 0.111±0.009 | **0.211±0.017** |
| Corel5k | **0.174±0.018** | 0.005±0.000 | 0.003±0.001 | 0.078±0.002 | 0.025±0.000 | 0.003±0.000 | 0.094±0.002 |
| Pascal | 0.231±0.017 | 0.034±0.001 | 0.164±0.012 | 0.416±0.008 | 0.127±0.001 | 0.169±0.007 | **0.420±0.007** |

Table 1: Experimental comparisons of VAMS with other comparing methods on six evaluation metrics, where the best performances on each metric are shown in bold face. For H-L, R-L and Cov, the lower value indicates the better performance. For A-P, Micro-F1 and Macro-F1, the higher value indicates the better performance.

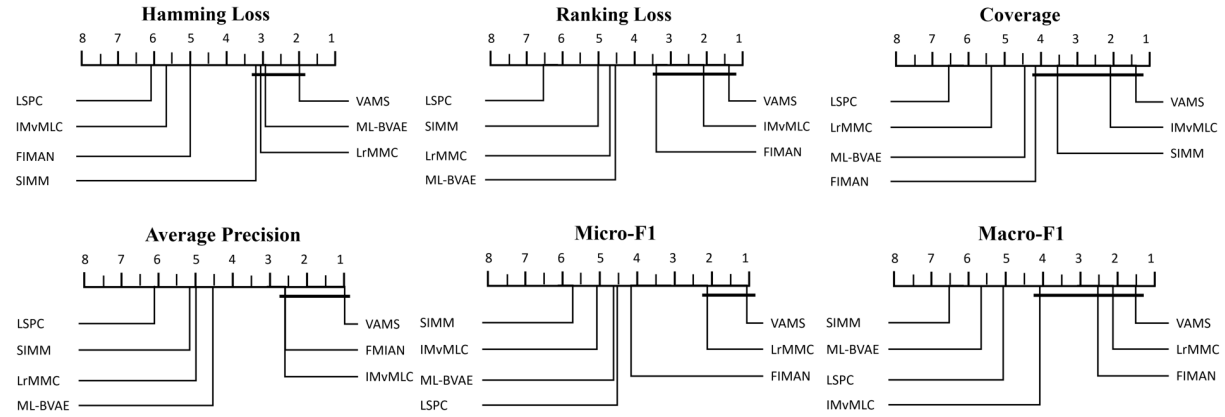



Figure 3: Experimental comparisons of our proposed VAMS against other comparing algorithms with the Bonferroni-Dunn test. Algorithms not connected with VAMS are significantly inferior to VAMS (CD = 3.046 at 0.05 significance level).

# Experiments

## Experimental Setup

To evaluate our proposed VAMS method, we conducted experiments on seven widely-used MVML datasets, including *Emotions*, *Scene*, *Yeast*, *Plant*, *Human*, *Corel5k* and *Pascal*, which can be downloaded from Mulan website[1]. Meanwhile, we compare it with several state-of-the-art methods, including **LrMMC** (Liu et al. 2015), **LSPC** (Szymański, Kajdanowicz, and Kersting 2016), **SIMM** (Wu et al. 2019), **FIMAN** (Wu et al. 2020), **IMvMLC** (Wen et al. 2024) and **ML-BVAE** (Fu et al. 2024). The configured parameters of all comparing methods are set according to the suggestions in their corresponding literature. Additionally, we adopt six widely used multi-label metrics to evaluate each method, including Hamming Loss (H-L), Ranking Loss (R-L), Coverage (Cov), Average Precision (A-P), Micro-F1 and Macro-F1. The detailed definitions of these metrics are available in (Sun and Zong 2021). Finally, we conduct experimental comparison between VAMS and all other methods, where five-fold cross-validation is performed on each data set.

## Experimental Results

Table 1 illustrates the experimental comparison between our proposed VAMS and other six comparing methods across all evaluation metrics, where the mean results and standard deviations are recorded. According to the 252 statistical comparisons, some key observations are clearly revealed:

- Among all comparing methods, our proposed VAMS is superior to LSPC in all cases. Meanwhile, it also outperforms FIMAN in 97.6% cases, ML-BVAE in 95.2% cases, IMvMLC in 92.9% cases, SIMM in 88% cases, and LrMMC in 85% cases, respectively.
- Among all evaluation metrics, our proposed VAMS achieves the best performance on Average Precision metric, and it also outperforms other methods over 98% cases on Micro-F1, 93.9% on Ranking Loss and Coverage, 91.8% on Macro-F1, and 89.8% on Hamming Loss.

[1]http://mulan.sourceforge.net/datasets-mlc.html

| Evaluation Metric | $\tau_F$ | critical value |
|---|---|---|
| Hamming Loss | 7.203 | |
| Ranking Loss | 11.367 | |
| Coverage | 11.974 | 2.365 |
| Average Precision | 7.235 | Methods:7, Data Set:7 |
| Micro-F1 | 8.967 | |
| Macro-F1 | 20.902 | |

Table 2: Friedman statics $\tau_F$ in terms of each evaluation metric (at 0.05 significance level).

- Additionally, the improvements of our proposed VAMS against other methods are quite significant, especially compared with the second-best method, our proposed VAMS shows a significant improvement of 2% to 4% on Average Precision and 2% to 14% on Micro-F1.

In order to comprehensively evaluate the superiority of the proposed VAMS, we employ the Friedman test (Demšar 2006) as the statistical method to analyze relative performance among the comparing algorithms. As shown in Table 2, the null hypothesis of distinguishable performance among the comparing algorithms is rejected at 0.05 significance level. Consequently, we utilize the post-hoc Bonferroni-Dunn test (Demšar 2006) to further compare the relative performance among the algorithms. Figure 3 illustrates the Critical Difference (CD) diagrams on each evaluation metric, with the average rank of each algorithm marked along the axis. According to Figure 3, it is observed that VAMS always ranks 1st on all evaluation metrics and it performs significant superiority against most comparing methods.

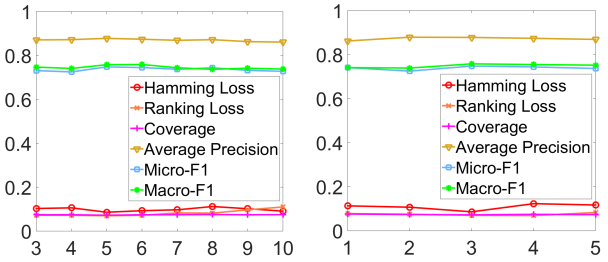
# Further Analysis

## Ablation Study

To evaluate the effectiveness of each component in our proposed framework, we perform an ablation study comparing VAMS with its three degenerated versions: VAMS-w/o IntraKNN, VAMS-w/o InterConn, VAMS-w/o SemCor, where

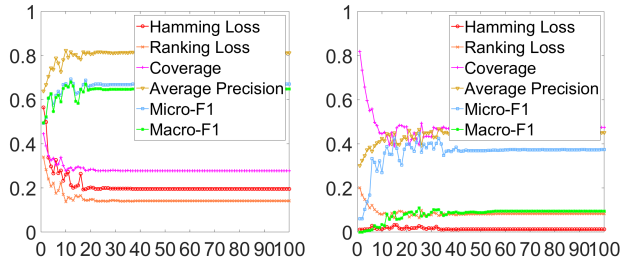
| Emotions | Hamming Loss | Ranking Loss | Coverage | Average precision | Micro-F1 | Macro-F1 |
|---|---|---|---|---|---|---|
| VAMS-w/o IntraKNN | 0.224±0.025 | 0.149±0.016 | 1.707±0.079 | 0.818±0.018 | 0.689±0.019 | 0.676±0.020 |
| VAMS-w/o InterConn | 0.223±0.020 | 0.151±0.024 | 1.727±0.155 | 0.812±0.031 | 0.614±0.058 | 0.601±0.062 |
| VAMS-w/o SemCor | 0.218±0.018 | 0.145±0.016 | 1.692±0.087 | 0.826±0.019 | 0.682±0.016 | 0.665±0.024 |
| VAMS | **0.193±0.019** | **0.144±0.013** | **1.659±0.055** | **0.826±0.012** | **0.692±0.015** | **0.679±0.017** |
| Scene | Hamming Loss | Ranking Loss | Coverage | Average precision | Micro-F1 | Macro-F1 |
| VAMS-w/o IntraKNN | 0.089±0.009 | 0.071±0.007 | 0.442±0.037 | 0.872±0.011 | 0.746±0.018 | 0.757±0.013 |
| VAMS-w/o InterConn | 0.094±0.008 | 0.075±0.008 | 0.463±0.054 | 0.865±0.013 | 0.713±0.022 | 0.722±0.019 |
| VAMS-w/o SemCor | 0.094±0.010 | **0.070±0.003** | 0.434±0.018 | 0.876±0.009 | 0.740±0.016 | 0.749±0.016 |
| VAMS | **0.086±0.007** | **0.070±0.010** | **0.434±0.008** | **0.878±0.014** | **0.748±0.016** | **0.758±0.014** |
| Corel5k | Hamming Loss | Ranking Loss | Coverage | Average precision | Micro-F1 | Macro-F1 |
| VAMS-w/o IntraKNN | 0.014±0.003 | 0.146±0.005 | 55.787±2.253 | 0.410±0.005 | 0.331±0.021 | 0.056±0.015 |
| VAMS-w/o InterConn | 0.016±0.007 | 0.151±0.006 | 54.088±2.427 | 0.428±0.006 | 0.315±0.047 | 0.047±0.043 |
| VAMS-w/o SemCor | 0.014±0.002 | 0.149±0.003 | 54.009±2.352 | 0.408±0.004 | 0.340±0.035 | 0.071±0.026 |
| VAMS | **0.013±0.008** | **0.082±0.004** | **53.200±2.218** | **0.452±0.009** | **0.372±0.005** | **0.094±0.002** |

Table 3: The experimental results of our proposed VAMS and its three degenerated methods over all employed evaluation metrics on *Emotions*, *Scene* and *Pascal* data sets, where VAMS-w/o IntraKNN, VAMS-w/o InterConn and VAMS-w/o SemCor do not consider the intra-view correlations, inter-view alignments, and label semantic correlations, respectively.




(a) The number of neighbors (b) The number of conv layers

Figure 4: Performance comparisons with different parameters configuration on *Scene* dataset.




(a) Emotions (b) Corel5k

Figure 5: The convergence analysis of VAMS on *Emotions* and *Corel5k* data sets.

each degenerated algorithm ignores the intra-view K-nearest neighbors, inter-view connections, and label semantic correlations, respectively. Table 3 records the experimental results on *Emotions*, *Scene* and *Corel5k* data sets. Specifically, compared with removing label semantic correlations, the model performance drops more significantly when removing intra-view K-nearest neighbors or inter-view connections, which indicates that multi-view data fusion has more contribution to the performance of model than label semantic correlations. Meanwhile, VAMS significantly surpasses its three degenerated algorithms, which also strongly demonstrates the superiority of utilizing such three multi-granularity relationships simultaneously when learning from MVML data.

## Sensitivity Analysis

We study the sensitivity analysis of VAMS with regard to its two parameters: the number of intra-view neighbors $k$ and convolution layers $C$. Figure 4(a) shows the performance changes as $k$ increases from 3 to 10, where the performance gradually improves and then slightly declines. In our experiments, we set $k$ to 5. Figure 4(b) illustrates the performance of VAMS with different numbers of convolution layers, where the optimal result is reached when $C$ is set to 3.

## Convergence Analysis

We conduct convergence analysis on *Emotions* and *Corel5k* data sets. Figure 5 illustrates the performance of VAMS as the number of epochs increases, where the Coverage results are normalized to make all metric results be characterized in a unified figure. According to Figure 5, the performance of VAMS gradually improves and becomes stable. Therefore, the convergence of VAMS is empirically demonstrated.

## Conclusion

In this paper, we proposed a new matching selection based MVML method, which integrates cross-view consensus information and individual-view specificity information into a unified framework. To the best of our knowledge, it is the first time to incorporate matching selection mechanism into MVML task, which avoids the separation of multi-view consistency and specificity, and significantly enhances the semantic characterization capability of the model.

## 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

Demšar, J. 2006. Statistical Comparisons of Classifiers Over Multiple Data Sets. *The Journal of Machine Learning Research*, 7(1): 1–30.

Dong, K.; Deik, D.-G.-X.; Lee, Y.; Zhang, H.; Li, X.; Zhang, C.; and Liu, Y. 2024. Multi-view Content-Aware Indexing for Long Document Retrieval. *arXiv preprint arXiv:2404.15103*.

Feng, L.; An, B.; and He, S. 2019. Collaboration Based Multi-Label Learning. In *AAAI Conference on Artificial Intelligence*, 3550–3557.

Feng, L.; Huang, J.; Shu, S.; and An, B. 2022. Regularized Matrix Factorization for Multilabel Learning With Missing Labels. *IEEE Transactions on Cybernetics*, 52(5): 3710–3721.

Feng, L.; Kaneko, T.; Han, B.; Niu, G.; An, B.; and Sugiyama, M. 2020. Learning with Multiple Complementary Labels. *International Conference on Machine Learning*, 3072–3081.

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.

Jiang, B.; Xiang, J.; Wu, X.; He, W.; Hong, L.; and Sheng, W. 2021. Robust Adaptive-Weighting Multi-view Classification. In *ACM International Conference on Information & Knowledge Management*, 3117–3121.

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 *ACM International Conference on Multimedia*, 9048–9056.

Liu, J.; Liu, X.; Yang, Y.; Liao, Q.; and Xia, Y. 2023a. Contrastive Multi-View Kernel Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 9552–9566.

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 Conference on Artificial Intelligence*, 2778–2784.

Liu, W.; Tsang, I. W.-H.; and Müller, K.-R. 2017. An Easy-to-hard Learning Paradigm for Multiple Classes and Multiple Labels. *Journal of Machine Learning Research*, 18(94): 1–38.

Liu, W.; Xu, D.; Tsang, I. W.-H.; and Zhang, W. 2019. Metric Learning for Multi-Output Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2): 408–422.

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.

Luo, S.; Zhang, C.; Zhang, W.; and Cao, X. 2018. Consistent and Specific Multi-View Subspace Clustering. In *AAAI Conference on Artificial Intelligence*, 3730–3737.

Luo, Y.; Liu, T.; Tao, D.; and Xu, C. 2015. Multiview Matrix Completion for Multilabel Image Classification. *IEEE Transactions on Image Processing*, 24(8): 2355–2368.

Lyu, G.; Deng, X.; Wu, Y.; and Feng, S. 2022. Beyond Shared Subspace: A View-Specific Fusion for Multi-View Multi-Label Learning. In *AAAI Conference on Artificial Intelligence*, 7647–7654.

Lyu, G.; Kang, W.; Wang, H.; Li, Z.; Yang, Z.; and Feng, S. 2024a. Common-Individual Semantic Fusion for Multi-View Multi-Label Learning. In *International Joint Conference on Artificial Intelligence*, 4715–4723.

Lyu, G.; Yang, Z.; Deng, X.; and Feng, S. 2024b. L-VSM: Label-Driven View-Specific Fusion for Multiview Multilabel Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Sun, S.; and Zong, D. 2021. LCBM: A Multi-View Probabilistic Model for Multi-Label Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8): 2682–2696.

Szymański, P.; Kajdanowicz, T.; and Kersting, K. 2016. How Is a Data-Driven Approach Better than Random Choice in Label Space Division for Multi-Label Classification? *Entropy*, 18(8): 282.

Tan, Q.; Yu, G.; Wang, J.; Domeniconi, C.; and Zhang, X. 2021. Individuality- and Commonality-Based Multiview Multilabel Learning. *IEEE Transactions on Cybernetics*, 51(3): 1716–1727.

Tan, X.; Zhao, C.; Liu, C.; Wen, J.; and Tang, Z. 2024. A Two-Stage Information Extraction Network for Incomplete Multi-View Multi-Label Classification. In *AAAI Conference on Artificial Intelligence*, 15249–15257.

Tsoumakas, G.; and Ioannis Katakis. 2007. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3(3): 1–13.

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, T.; Liu, H.; Li, Y.; Jin, Y.; Hou, X.; and Ling, H. 2020. Learning Combinatorial Solver for Graph Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 7568–7577.

Wei, H.; Xie, R.; Feng, L.; Han, B.; and An, B. 2023. Deep Learning From Multiple Noisy Annotators as A Union. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12): 10552–10562.

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.

Wu, J.-H.; Wu, X.; Chen, Q.; Hu, Y.; and Zhang, M.-L. 2020. Feature-Induced Manifold Disambiguation for Multi-View Partial Multi-label Learning. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 557–565.

Wu, X.; Chen, Q.; Hu, Y.; Wang, D.-B.; Chang, X.; Wang, X.; and Zhang, M.-L. 2019. Multi-View Multi-Label Learning with View-Specific Information Extraction. In *International Joint Conference on Artificial Intelligence*, 3884–3890.

Wu, Y.; Liu, H.; Feng, S.; Jin, Y.; Lyu, G.; and Wu, Z. 2021. GM-MLIC: Graph Matching based Multi-Label Image Classification. In *International Joint Conference on Artificial Intelligence*, 1179–1185.

Yang, P.; Xie, M.; Zong, C.-C.; Feng, L.; Niu, G.; Sugiyama, M.; and Huang, S.-J. 2023. Multi-Label Knowledge Distillation. In *IEEE/CVF International Conference on Computer Vision*, 17271–17280.

Zhang, C.; Jiang, B.; Wang, Z.; Yang, J.; Lu, Y.; Wu, X.; and Sheng, W. 2023a. Efficient Multi-View Semi-Supervised Feature Selection. *Information Sciences*, 649: 119675.

Zhang, C.; Li, H.; Lv, W.; Huang, Z.; Gao, Y.; and Chen, C. 2023b. Enhanced Tensor Low-Rank and Sparse Representation Recovery for Incomplete Multi-View Clustering. In *AAAI Conference on Artificial Intelligence*, 11174–11182.

Zhang, C.; Yu, Z.; Hu, Q.; Zhu, P.; Liu, X.; and Wang, X. 2018. Latent Semantic Aware Multi-View Multi-Label Classification. In *AAAI Conference on Artificial Intelligence*, 4414–4421.

Zhang, F.; Jia, X.; and Li, W. 2020. Tensor Based Multi-View Label Enhancement for Multi-Label Learning. In *International Joint Conference on Artificial Intelligence*, 2369–2375.

Zhang, X.; Luo, T.; Liu, Y.; and Hou, C. 2024. Imbalanced Multi-Instance Multi-Label Learning via Coding Ensemble and Adaptive Thresholds. In *ACM International Conference on Multimedia*, 5413–5422.

Zhao, X.; An, Y.; Qi, L.; and Geng, X. 2024. Scalable Label Distribution Learning for Multi-Label Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Zhao, X.; An, Y.; Xu, N.; and Geng, X. 2022. Fusion Label Enhancement for Multi-Label Learning. In *International Joint Conference on Artificial Intelligence*, 3773–3779.

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.