VADIS: Investigating Inter-View Representation Biases for Multi-View Partial Multi-Label Learning

Jie Wang[1] **Ning Xu**[1] **Xin Geng**[1]

[1]School of Computer Science and Engineering, Southeast University, Nanjing, China

Abstract

Multi-view partial multi-label learning (MVPML) deals with training data where each example is represented by multiple feature vectors and associated with a set of candidate labels, only a subset of which are correct. The diverse representation biases present in different views complicate the annotation process in MVPML, leading to the inclusion of incorrect labels in the candidate label set. Existing methods typically merge features from different views to identify the correct labels in the training data without addressing the representation biases inherent in different views. In this paper, we propose a novel MVPML method called VADIS, which investigates view-aware representations for disambiguation and predictive model learning. Specifically, we exploit the global common representation shared by all views, aligning it with a local semantic similarity matrix to estimate ground-truth labels via a low-rank mapping matrix. Additionally, to identify incorrect labels, the view-specific inconsistent representation is recovered by leveraging the sparsity assumption. Experiments on real-world datasets validate the superiority of our approach over other state-of-the-art methods.

1 INTRODUCTION

Partial multi-label learning has gained significant research attention as a means of modeling objects with imprecise semantics Xie and Huang [2018]. In this paradigm, each example is represented by a single feature vector associated with a candidate label set, of which only a subset is deemed valid. In recent years, this framework has been widely employed in many real-world scenarios with inaccurate supervision Xie and Huang [2018], Yu et al. [2018], Sun et al. [2019], Zhang and Fang [2020].

Figure 1: An example multi-view partial multi-label scenario. The news webpage can be represented from different views such as text, audio, and video. Among the candidate label set given by the crowdsourced annotators, only 'Argentina', 'FIFA World Cup', and 'France' are correct.

However, due to the intricate nature of real-world scenarios, objects often encompass descriptions from multiple perspectives, resulting in complex properties. Furthermore, the intricate representations stemming from these diverse viewpoints intensify the challenge of annotation, thereby elevating the likelihood of incorrect labeling. As illustrated in Figure 1, a news webpage can contain multiple views, such as text, audio, and video, each accompanied by numerous candidate labels contributed by crowdsourced annotators. Among these labels, only *Argentina*, *FIFA World Cup*, and *France* are correct.

To deal with the task under these circumstances, multi-view partial multi-label learning has emerged Chen et al. [2020], where each example is represented by multiple feature vectors associated with a candidate label set, of which only a subset is correct. Several works have been proposed to address the MVPML problem. One previous attempt Chen et al. [2020] induces a predictive model by simply fusing the similarity matrices over each view, followed by label propagation to disambiguate the candidate label set. Another method Wu et al. [2020] leverages the aggregated manifold structure of each view, and then maps the manifold struc-

ture to the label space for disambiguation. Furthermore, the latent label distribution is also extracted from the candidate labels by incorporating the graph-fusion-based topological structure of the feature space to obtain a predictive model Xu et al. [2022].

However, the aforementioned works simply fuse the features from separate views to facilitate disambiguation, overlooking the consideration of representation biases existing across these views. Actually, the diverse representation biases from different views increase the annotation challenge in MVPML, resulting in the fact that the incorrect labels are selected into the candidate label set in the training data. As the underlying cause of the generation of incorrect labels in the candidate label set is the foundation of inferring the true labels to train a reasonable classifier on MVPML data, the representation biases across different views should be considered for solving the MVPML problem.

To alleviate this issue, a novel approach named VADIS, i.e., *View-Aware DISambiguation for multi-view partial multi-label learning* is proposed to explore the feature representation biases via leveraging the properties of different views to identify the true labels, and induce the predictive model. Specifically, we utilize a global common feature representation shared across all views. This representation, which corresponds to the local similarity matrix in the semantic space, is employed to estimate the ground-truth labels by introducing a low-rank mapping matrix. Moreover, we recover the view-specific feature representation influenced by inconsistencies, to identify incorrect labels using the sparsity assumption. Extensive experiments on real-world datasets validate the superiority of VADIS over other state-of-the-art methods for solving MVPML problem.

2 RELATED WORKS

In this section, two learning frameworks, namely *partial multi-label learning* (PML) Xie and Huang [2018] and *multi-view multi-label learning* (MVML) Luo et al. [2013] are introduced, which are closely related to the multi-view partial multi-label learning (MVPML).

MVPML can be seen as a specialized case of the well-known PML problem when each instance is represented by a single feature vector from the same view in the input space. The PML problem aims to learn from the data where each instance is associated with a set of candidate labels, among which only a subset is considered correct. Several approaches have been proposed in this domain, involving the use of the confidence scores associated with each candidate label to determine the correct labels Xie and Huang [2018], Xu et al. [2020]. Additionally, the low-rank assumption is employed to identify noise labels for disambiguation Sun et al. [2019], Yu et al. [2018]. The credible label elicitation method is used to construct the final prediction, with correct labels being detected from each candidate label set Zhang and Fang [2020]. Furthermore, noisy label identification Xie and Huang [2021] is proposed to tackle the noise labels and ground-truth labels simultaneously.

On the other hand, MVPML can degenerate into the MVML problem Luo et al. [2013] when the interference of false-positive labels is absent in the label space. MVML is aimed to learn a multi-label classifier from the training data where each example is represented with multiple feature vectors and associated with multiple correct labels simultaneously. Previous works detect the informative subspaces over different views to learn a predictive model. These low-dimensional shared subspaces are designed to handle multi-label image classification, using constraints like consistency regularization Luo et al. [2013], Zhu et al. [2015], Liu et al. [2015]. Additionally, some papers propose the co-training framework Blum and Mitchell [1998], Zhou and Li [2010] to explore the reliable labeling information communication over different views by Confidence-rated filtering Xing et al. [2018] and diversity maximization Zhan and Zhang [2017]. The Hilbert-Schmidt Independence Criterion Zhang et al. [2018] and matrix factorization Zhu et al. [2018] based on the measurement of multi-view correlations are used to discover the shared subspaces. Furthermore, the view-specific information is also utilized to learn the classification model Wu et al. [2019].

Both MVML and PML can be viewed as degenerated versions of MVPML, which makes the task of learning from MVPML data more challenging to solve. One previous attempt towards MVPML Chen et al. [2020] induces a predictive model by simply fusing the similarity matrices from each view and employing label propagation to disambiguate the candidate label set. Another approach Wu et al. [2020] deploys the aggregated manifold structure of each view to disambiguate the candidate label set by mapping the manifold structure to the label space. The latent label distribution is also learned from the candidate labels by graph-fusion-based incorporation of the topological structure within the feature space, to induce a predictive model Xu et al. [2022]. Nonetheless, these methods neglect the impact of the feature representation biases in different views on the label space.

In the next section, a novel MVPML approach named VADIS with strong generalization performance is proposed, where the representation biases in different views are considered to learn a predictive model.

3 THE PROPOSED APPROACH

3.1 PROBLEM FORMULATION

Formally, let $\mathcal{X} = R^{d_1} \times R^{d_2} \ldots \times R^{d_V}$ denote the input space consisting of V views, where each view v has a dimensionality of d_v $(1 \leq v \leq V)$. Furthermore, let $\mathcal{Y} =$

$\{y_1, \ldots, y_c\}$ denote the label space consisting of c possible class labels. Let $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq n\}$ denote the MVPML training set, where $\boldsymbol{x}_i = \left[\boldsymbol{x}_i^1; \boldsymbol{x}_i^2; \ldots; \boldsymbol{x}_i^V\right] \in \mathcal{X}$ is the $\left(d = \sum_{v=1}^{V} d_v\right)$-dimensional multi-view instance and $Y_i \subseteq \mathcal{Y}$ is the candidate label set associated with x_i. Here, the ground-truth label set $\widetilde{Y}_i \subseteq \mathcal{Y}$ for x_i is concealed in its candidate label set (i.e. $\widetilde{Y}_i \subseteq Y_i$) and thus not directly accessible. Accordingly, the task of MVPML is to learn a multi-label classification model $h : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ from $\mathcal{D}$ which is capable of predicting the proper labels for unseen instances.

In this paper, let $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in R^{d \times n}$ denote the feature matrix, where $\mathbf{X}^v = [\boldsymbol{x}_1^v, \boldsymbol{x}_2^v, \ldots, \boldsymbol{x}_n^v] \in R^{d_v \times n}$ $(1 \leq v \leq V)$ is the feature matrix of the v-th view. Furthermore, let $\mathbf{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_n]^\top$ denote the partial multi-label matrix, where $\boldsymbol{l}_i = [l_i^{y_1}; l_i^{y_2}; \ldots; l_i^{y_c}] \in \{0, 1\}^c$ is the observed label vector of $\boldsymbol{x}_i$, i.e., $l_i^{y_j} = 1$ if $y_j \in Y_i$, otherwise $l_i^{y_j} = 0$.

3.2 THE VADIS FRAMEWORK

In order to capture the view-aware feature representations from diverse views, we employ the self-representation approach where an instance can be represented as a linear combination of other instances. This enables us to learn the following self-representation $\mathbf{Z}^v$ of v-th $(1 \leq v \leq V)$ view:

$$\mathbf{X}^v = \mathbf{X}^v \mathbf{Z}^v + \mathbf{E}^v, \tag{1}$$

where $\mathbf{Z}^v \in R^{n \times n}$ is the learned self-representation matrix and $\mathbf{E}^v \in R^{d_v \times n}$ is the error term. In Eq. (1), the feature representation of each view is reconstructed to the same dimension, which facilitates subsequent learning. Additionally, the reconstruction objective effectively alleviates the potential loss of representation associated with each view.

Due to the diverse properties inherent in different views, biases arise in the representation of views, alongside the shared common representation. These biases are caused by inconsistencies from different views, prompting us to recover the view-specific-inconsistent representation, denoted as $\mathbf{O}^v$. Subsequently, the view-aware self-representation $\mathbf{Z}^v$ can be decomposed into the global common representation $\mathbf{C}$ shared by all views, and the view-specific-inconsistent representation $\mathbf{O}^v$.

$$\mathbf{Z}^v = \mathbf{C} + \mathbf{O}^v. \tag{2}$$

In Eq. (2), the common feature representation matrix $\mathbf{C}$ serves as a global representation, assumed to be related to the local similarity among instances within each view in the semantic space. The local similarity of the v-th view denoted as $\mathbf{S}^v$, is defined. Notably, as the feature representations of distinct instances grow more alike, the common representation becomes richer in the specific view. Intuitively, the common feature representation should exhibit greater similarity to $S_{i,j}^v$ when the i-th and j-th instances within the v-th view share stronger correlations. Each element $S_{i,j}^v$ can be calculated by $S_{i,j}^v = \exp(-\frac{\|\boldsymbol{x}_i^v - \boldsymbol{x}_j^v\|^2}{2})$ if $\boldsymbol{x}_i$ is among K-nearest neighbors of $\boldsymbol{x}_j$, otherwise $S_{i,j}^v = 0$. Here, K represents a preset parameter.

Furthermore, it is essential to note that inconsistent representations are inherently not universal, making it reasonable to assume sparsity in the view-specific-inconsistent representation matrix $\mathbf{O}^v$. Therefore, we can formulate the following objective function:

$$\begin{aligned} \min_{\mathbf{C}, \mathbf{O}^v} \sum_{v=1}^{V} \|\mathbf{X}^v - \mathbf{X}^v(\mathbf{C} + \mathbf{O}^v)\|_F^2 + \|\mathbf{O}^v\|_1 + \|\mathbf{C} - \mathbf{S}^v\|_F^2 \\ s.t.\ \ \mathbf{X}^v = \mathbf{X}^v \mathbf{Z}^v + \mathbf{E}^v, \mathbf{Z}^v = \mathbf{C} + \mathbf{O}^v. \end{aligned} \tag{3}$$

Here, the first part represents the self-representation loss, the second part enforces a sparse constraint on the view-specific-inconsistent representation, and the final term adds a constraint based on local similarity.

The observed label matrix $\mathbf{L}$ can be divided into two distinct parts: the ground-truth label matrix $\widetilde{\mathbf{L}}$ and the remaining incorrect label matrix $\mathbf{N}$.

$$\mathbf{L} = \widetilde{\mathbf{L}} + \mathbf{N}. \tag{4}$$

To estimate the ground-truth labels from the observed labels, a matrix $\mathbf{P} \in R^{n \times c}$ is constructed to map the common feature representation $\mathbf{C}$ to the label space. Furthermore, the incorrect labels are influenced by representation biases from different views, suggesting a certain connection between the view-specific-inconsistent feature representation and the incorrect labels. Therefore, we introduce a mapping matrix $\mathbf{Q} \in R^{n \times c}$ to identify the incorrect labels. As a result, we obtain the following formula:

$$\widetilde{\mathbf{L}} \approx \mathbf{CP},\ \mathbf{N} \approx \mathbf{OQ}. \tag{5}$$

Here, $\mathbf{O}$ denotes the comprehensive inconsistent representation acquired through the fusion of view-specific representations from various views.

It is important to acknowledge that there are well-established label correlations among different labels in multi-label learning Zhang and Zhou [2013], which leads to the assumption that $\mathbf{P}$ is linearly dependent to effectively capture such label correlations, implying $\mathbf{P}$ is low-rank. Since the rank function poses optimization challenges due to its discrete nature, we opt for the nuclear norm as a replacement Sun et al. [2019]. Furthermore, the incorrect labels in partial multi-label learning tend to be sparse within the candidate label set. To simplify the optimization, we impose a sparse constraint on the mapping matrix $\mathbf{Q}$. Consequently, we can

formulate the following objective function:

$$\begin{aligned}\min_{\mathbf{C},\mathbf{O}^v,\mathbf{P},\mathbf{Q}} & \|\mathbf{L}-\mathbf{CP}-\mathbf{OQ}\|_F^2+\gamma_1\|\mathbf{P}\|_*+\gamma_2\|\mathbf{Q}\|_1 \\ s.t.\ & \mathbf{X}^v=\mathbf{X}^v\mathbf{Z}^v+\mathbf{E}^v \\ & \mathbf{Z}^v=\mathbf{C}+\mathbf{O}^v, \mathbf{O}=\frac{1}{V}\sum\nolimits_{v=1}^{V}\mathbf{O}^v.\end{aligned} \tag{6}$$

Here, the first part ensures that the learned labels match the observed labels. The last two components represent low-rank and sparse constraints applied to different mapping matrices, where γ_1 and γ_2 are the trade-off parameters.

Subsequently, the candidate label set is disambiguated by identifying the correct labels, and a linear predictive model $\mathbf{W}$ is induced. The overall optimization problem can be achieved as follows:

$$\begin{aligned}\min_{\substack{\mathbf{C},\mathbf{P},\mathbf{W},\\ \mathbf{O}^v,\mathbf{Q}}} & \sum_{v=1}^{V}\|\mathbf{X}^v-\mathbf{X}^v(\mathbf{C}+\mathbf{O}^v)\|_F^2+\|\mathbf{C}-\mathbf{S}^v\|_F^2 \\ & +\|\mathbf{O}^v\|_1+\|\mathbf{L}-\mathbf{CP}-\mathbf{OQ}\|_F^2+\gamma_1\|\mathbf{P}\|_* \\ & +\gamma_2\|\mathbf{Q}\|_1+\|\mathbf{L}-\mathbf{OQ}-\mathbf{X}^\top\mathbf{W}\|_F^2+\|\mathbf{W}\|_F^2 \\ s.t.\ & \mathbf{X}^v=\mathbf{X}^v\mathbf{Z}^v+\mathbf{E}^v \\ & \mathbf{Z}^v=\mathbf{C}+\mathbf{O}^v, \mathbf{O}=\frac{1}{V}\sum\nolimits_{v=1}^{V}\mathbf{O}^v.\end{aligned} \tag{7}$$

Here, $\|\mathbf{W}\|_F^2$ is a regularization term to control the model complexity.

3.3 OPTIMIZATION

In this section, an iterative strategy is utilized to address the final optimization problem in Eq. (7). When $\mathbf{O}^v, \mathbf{P}, \mathbf{Q}$ and $\mathbf{W}$ are fixed, $\mathbf{C}$ could be updated by the following ordinary least squares problem:

$$\begin{aligned}\min_{\mathbf{C}} & \sum_{v=1}^{V}\|\mathbf{X}^v-\mathbf{X}^v(\mathbf{C}+\mathbf{O}^v)\|_F^2+\|\mathbf{C}-\mathbf{S}^v\|_F^2 \\ & +\|\mathbf{L}-\mathbf{CP}-\mathbf{OQ}\|_F^2.\end{aligned} \tag{8}$$

When $\mathbf{C}, \mathbf{O}^v, \mathbf{P}$ and $\mathbf{Q}$ are fixed, $\mathbf{W}$ could be updated by the following ridge regression problem:

$$\min_{\mathbf{W}}\|\mathbf{Y}-\mathbf{OQ}-\mathbf{X}^\top\mathbf{W}\|_F^2+\|\mathbf{W}\|_F^2. \tag{9}$$

Note that the computational complexity would be demanding. Following Wang et al. [2019], we adopt an alternating optimization strategy BFGS for large-scale data sets.

When $\mathbf{C}, \mathbf{O}^v, \mathbf{Q}$ and $\mathbf{W}$ are fixed, $\mathbf{P}$ could be updated by the following problem:

$$\min_{\mathbf{P}}\|\mathbf{L}-\mathbf{CP}-\mathbf{OQ}\|_F^2+\gamma_1\|\mathbf{P}\|_*. \tag{10}$$

To solve the Eq. (10), we introduce an auxiliary variable $\mathbf{Z}\in R^{n\times c}$ and reformulate it into the following equivalent form:

$$\begin{aligned}&\min_{\mathbf{P}}\|\mathbf{L}-\mathbf{CP}-\mathbf{OQ}\|_F^2+\gamma_1\|\mathbf{Z}\|_* \\ &s.t.\ \mathbf{Z}=\mathbf{P},\end{aligned}$$

which can be solved by popular ADMM (Alternating Direction Method of Multiplier) techniques Boyd et al. [2011]. Firstly, an augmented Lagrange function is induced as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{P},\mathbf{Z},\mathbf{U};\rho)=&\|\mathbf{L}-\mathbf{CP}-\mathbf{OQ}\|_F^2+\gamma_1\|\mathbf{Z}\|_* \\ &+\langle\mathbf{U},\mathbf{P}-\mathbf{Z}\rangle+\frac{\rho}{2}\|\mathbf{P}-\mathbf{Z}\|_F^2,\end{aligned} \tag{11}$$

where $\mathbf{U}$ is a Lagrange multiplier matrix, and ρ is a penalty parameter. Then alternative optimization objectives are as follows:

$$\begin{cases}\mathbf{P}^{k+1}=\underset{\mathbf{P}}{\operatorname{argmin}}\|\mathbf{L}-\mathbf{CP}-\mathbf{OQ}\|_F^2+\frac{\rho}{2}\left\|\mathbf{P}-\mathbf{Z}^k+\frac{\mathbf{U}^k}{\rho}\right\|_F^2 \\ \mathbf{Z}^{k+1}=\underset{\mathbf{Z}}{\operatorname{argmin}}\gamma_1\|\mathbf{Z}\|_*+\frac{\rho}{2}\left\|\mathbf{Z}-\mathbf{P}^{k+1}-\frac{\mathbf{U}^k}{\rho}\right\|_F^2 \\ \mathbf{U}^{k+1}=\mathbf{U}^k+\rho\left(\mathbf{P}^{k+1}-\mathbf{Z}^{k+1}\right).\end{cases} \tag{12}$$

The optimization of $\mathbf{P}^{k+1}$ is an ordinary least squares problem that is easily solved by Eq. (13). Furthermore, inspired by Cai et al. [2010], the optimization of $\mathbf{Z}^{k+1}$ can be solved by Singular Value Thresholding (SVT).

$$\mathbf{P}^{k+1}=\left(2\mathbf{C}^\top\mathbf{C}+\rho\mathbf{I}\right)^{-1}\left(\ 2\mathbf{C}^\top(\mathbf{L}-\mathbf{OQ})+\rho\mathbf{Z}^k-\mathbf{U}^k\ \right) \tag{13}$$

$$\mathbf{Z}^{k+1}=D_{\frac{\gamma_1}{\rho}}\left(\mathbf{P}^{k+1}+\frac{\mathbf{U}^k}{\rho}\right). \tag{14}$$

Specifically, $\mathcal{D}_\tau(\mathbf{X})$ denotes the singular value thresholding operator given by $\mathcal{D}_\tau(\mathbf{X})=\mathbf{U}\mathcal{S}_\tau(\mathbf{\Sigma})\mathbf{V}^*$, where $\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ is any singular value decomposition. $\mathcal{S}_\tau: R\to R$ is the shrinkage operator $\mathcal{S}_\tau[x]=\operatorname{sgn}(x)\max(|x|-\tau,0)$ Zhuang et al. [2012].

When $\mathbf{C}, \mathbf{O}^v, \mathbf{P}$ and $\mathbf{W}$ are fixed, $\mathbf{Q}$ could be reformulated as follows:

$$\min_{\mathbf{Q}}\|\mathbf{L}-\mathbf{CP}-\mathbf{OQ}\|_F^2+\gamma_2\|\mathbf{Q}\|_1+\|\mathbf{L}-\mathbf{OQ}-\mathbf{X}^\top\mathbf{W}\|_F^2, \tag{15}$$

Which can also be solved by ADMM. The difference is that the optimization for $\mathbf{Z}_1^{k+1}$ can be achieved by employing the shrinkage operator Zhuang et al. [2012].

$$\mathbf{Z}_1^{k+1}=S_{\frac{\gamma_2}{\rho_1}}\left(\mathbf{Q}^{k+1}+\frac{\mathbf{U}_1^k}{\rho_1}\right), \tag{16}$$

where $\mathbf{Z}_1$ is an auxiliary variable, $\mathbf{U}_1$ is a Lagrange multiplier matrix, and ρ_1 is a penalty parameter.

When $\mathbf{C}, \mathbf{P}, \mathbf{Q}$ and $\mathbf{W}$ are fixed, $\mathbf{O}^v$ could be updated by

Table 1: Characteristics of the multi-view partial multi-label datasets.

DataSets	$\|S\|$	$V(S)$	$VDim(S)$	$CL(S)$	$LCard(S)$	*Domain*	*Description*	*Controlling Parameters*
Emotions	593	2	8 / 64	6	1.869	Music	Rhythm, Timbre	$p \in \{0.3, 0.7\}$ $r \in \{1, 2, 3\}$
Yeast	2,417	2	24 / 79	14	4.237	Biology	Genetic Expression, Phylogenetic Profile	
Pascal	9,963	5	42 / 100 / 196 / 370 / 310	20	1.465	Images	DenseHue, Gist, DenseSift, HSV, Tag	
EspGame5k	5,192	4	48 / 91 / 519 / 368	268	4.679	Images	DenseHue, Gist, DenseSift, HSV	
Mirflickr5k	5,000	5	48 / 93 / 112 / 359 / 318	38	4.711	Images	DenseHue, Gist, DenseSift, HSV, Tag	

the following problem:

$$\begin{aligned} \min_{\mathbf{O}^v} \sum_{v=1}^{V} & \|\mathbf{X}^v - \mathbf{X}^v(\mathbf{C} + \mathbf{O}^v)\|_F^2 + \|\mathbf{C} - \mathbf{S}^v\|_F^2 \\ & + \|\mathbf{O}^v\|_1 + \|\mathbf{L} - \mathbf{CP} - \mathbf{OQ}\|_F^2 \\ & + \|\mathbf{L} - \mathbf{OQ} - \mathbf{X}^\top \mathbf{W}\|_F^2 \\ s.t. \ & \mathbf{O} = \frac{1}{V} \sum\nolimits_{v=1}^{V} \mathbf{O}^v, \end{aligned} \tag{17}$$

which can also be solved by the ADMM techniques.

Due to page limitations, the complete pseudo-code of the algorithm can be found in the Appendix. In addition, we also analyse the time complexity of the VADIS algorithm, which can be viewed in the Supplementary Material.

3.4 PREDICTION

To predict the appropriate label set for a new instance $\boldsymbol{x}$, we employ a virtual label bipartition. Specifically, an additional virtual label y_0 is introduced as a threshold to categorize the labels as either relevant or irrelevant. Consequently, the label space $\mathcal{Y}$ is expanded to $\mathcal{Y}' = \mathcal{Y} \cup \{y_0\} = \{y_0, y_1, \ldots, y_c\}$. In this paper, $l_{\boldsymbol{x}}^{y_0}$ is set to be 0.5. Let $\mathbf{W}^* = [\boldsymbol{w}_0^*, \boldsymbol{w}_1^*, \ldots, \boldsymbol{w}_c^*]$ be the final predictive model, providing the following outputs for each class $y_j (0 < j < c)$:

$$\forall_{j=0}^{c} : f(y_j \mid \boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{w}_j^*. \tag{18}$$

Then, the predicted labels for $\boldsymbol{x}$ are obtained via splitting the outputs:

$$\zeta(\boldsymbol{x}) = \{y_j \mid f(y_j \mid \boldsymbol{x}) > f(y_0 \mid \boldsymbol{x}), 1 \leq j \leq c\}. \tag{19}$$

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets: Following Chen et al. [2020], Wu et al. [2020], Xu et al. [2022], five popular real-world multi-view multi-label datasets from different domains for experiments are selected, i.e., `Emotions` Trohidis et al. [2008], `Yeast` Elisseeff and Weston [2001], `Pascal` Everingham et al. [2010], `EspGame5k` Von Ahn and Dabbish [2004] and `Mirflickr5k` Huiskes and Lew [2008]. Table 1 summarizes the characteristics of the multi-view partial multi-label datasets. More details on the datasets can be found in Appendix.

This paper employs a widely-used approach Cour et al. [2011], Liu and Dietterich [2012], Yu and Zhang [2016] to construct MVPML examples through the introduction of candidate labels. The generation process, as shown in Table 1, relies on two controlling parameters, p and r. Here, $p \in (0, 1)$ is responsible for controlling the proportion of partially labeled samples within the dataset, while $r \in N$ is employed to regulate the quantity of false-positive labels in the candidate label set. Let $(\boldsymbol{x}, \widetilde{Y})$ represent a multi-view multi-label sample, where $\widetilde{Y}$ denotes the ground-truth label set. To construct an MVPML sample $(\boldsymbol{x}, Y)$, r false-positive labels $\Delta_r \subseteq \mathcal{Y} \setminus \widetilde{Y}$ are randomly inserted into $\widetilde{Y}$, i.e., $Y = \widetilde{Y} \cup \Delta_r$. This construction process is executed using six distinct parameter control settings, with $p \in \{0.3, 0.7\}$ and $r \in \{1, 2, 3\}$ for each real-world dataset in Table 1.

Comparing Algorithms: The performance evaluation of VADIS involves a comprehensive comparison with various state-of-the-art approaches. This comparative analysis considers multiple methodologies, each meticulously selected based on recommendations provided in the pertinent literature:

- GLADE Xu et al. [2022] where the latent label distribution is exploited from the candidate labels via the graph-fusion-based incorporation of the topological structure in the feature space to induce a predictive model [recommended configuration: $\lambda = 0.001, \gamma_1 = 5, \gamma_2 = 20$].
- FIMAN Wu et al. [2020] where the aggregated manifold structure is learned to disambiguate the candidate label set [recommended configuration: $t_d = 0.4, t_p = 0.6, k = 10$, and $\eta = 1$].
- GRADIS Chen et al. [2020] where the graph-based label propagation is adopted to learn multi-view representation and disambiguate candidate label set [recommended configuration: $\eta = 0.1, \alpha = 0.95, k = 10$].
- PML-LRS Sun et al. [2019] where the sparse and low-rank decomposition strategy are employed to learn from partial multi-label examples [recommended configuration: $\beta = 1, \gamma = 0.1, \eta = 1$].
- LSAMML Zhang et al. [2018] where the Hilbert-Schmidt Independence Criterion is used for multi-view representation [recommended configuration: grid search for $\gamma, \beta \in \left\{10^{-2}, 10^{-1}, \ldots, 10^{2}\right\}$].

Table 2: Predictive performance of each comparing method on five datasets in terms of *Ranking Loss* (mean ± std). The best performance is marked in bold (the smaller the better).

Datasets	Controlling Parameters	Comparing Approaches							
		VADIS	GLADE	F2L2IF	FIMAN	FPML	GRADIS	PML-LRS	LSAMML
Emotions	$r=1,p=0.3$	**0.143±0.027**	0.162±0.020	0.232±0.032	0.175±0.022	0.217±0.031	0.184±0.025	0.218±0.030	0.178±0.028
	$r=1,p=0.7$	**0.153±0.030**	0.167±0.025	0.233±0.019	0.186±0.025	0.217±0.027	0.231±0.032	0.217±0.026	0.213±0.047
	$r=2,p=0.3$	**0.149±0.027**	0.173±0.027	0.238±0.027	0.179±0.023	0.224±0.031	0.204±0.025	0.228±0.030	0.194±0.023
	$r=2,p=0.7$	**0.172±0.033**	0.189±0.027	0.254±0.028	0.194±0.016	0.237±0.021	0.298±0.034	0.250±0.035	0.217±0.033
	$r=3,p=0.3$	**0.144±0.026**	0.174±0.027	0.240±0.031	0.176±0.024	0.230±0.028	0.207±0.041	0.227±0.030	0.189±0.024
	$r=3,p=0.7$	**0.186±0.030**	0.189±0.033	0.285±0.036	0.188±0.026	0.267±0.024	0.333±0.043	0.316±0.029	0.229±0.030
Yeast	$r=1,p=0.3$	**0.164±0.008**	0.167±0.009	0.362±0.015	0.183±0.011	0.212±0.011	0.183±0.008	0.214±0.011	0.502±0.020
	$r=1,p=0.7$	**0.167±0.008**	0.170±0.009	0.366±0.014	0.186±0.011	0.212±0.011	0.214±0.008	0.213±0.011	0.500±0.030
	$r=2,p=0.3$	**0.168±0.009**	**0.168±0.008**	0.364±0.014	0.185±0.012	0.211±0.011	0.198±0.010	0.213±0.011	0.505±0.022
	$r=2,p=0.7$	**0.168±0.011**	0.171±0.008	0.370±0.014	0.186±0.013	0.212±0.011	0.244±0.012	0.214±0.011	0.499±0.021
	$r=3,p=0.3$	**0.169±0.010**	**0.169±0.008**	0.365±0.014	0.185±0.011	0.211±0.011	0.208±0.013	0.214±0.011	0.501±0.025
	$r=3,p=0.7$	**0.170±0.010**	**0.170±0.010**	0.367±0.014	0.187±0.013	0.213±0.011	0.266±0.011	0.213±0.011	0.505±0.020
EspGame5k	$r=1,p=0.3$	**0.182±0.005**	0.185±0.007	0.234±0.007	0.225±0.007	0.289±0.011	0.234±0.010	0.254±0.007	0.232±0.005
	$r=1,p=0.7$	**0.185±0.004**	0.188±0.005	0.241±0.007	0.229±0.007	0.289±0.011	0.272±0.008	0.259±0.008	0.233±0.005
	$r=2,p=0.3$	**0.184±0.005**	0.188±0.006	0.238±0.008	0.229±0.007	0.289±0.012	0.253±0.006	0.259±0.010	0.235±0.007
	$r=2,p=0.7$	**0.191±0.006**	0.194±0.006	0.253±0.008	0.241±0.009	0.290±0.012	0.314±0.005	0.272±0.011	0.241±0.006
	$r=3,p=0.3$	**0.188±0.004**	0.191±0.006	0.243±0.006	0.233±0.006	0.289±0.010	0.265±0.005	0.265±0.009	0.236±0.004
	$r=3,p=0.7$	**0.196±0.006**	0.198±0.008	0.256±0.009	0.243±0.009	0.289±0.012	0.336±0.008	0.272±0.010	0.242±0.006
Pascal	$r=1,p=0.3$	**0.085±0.004**	0.089±0.003	0.217±0.008	0.116±0.005	0.267±0.010	0.102±0.008	0.329±0.006	0.244±0.008
	$r=1,p=0.7$	**0.089±0.004**	0.091±0.004	0.237±0.007	0.131±0.007	0.268±0.013	0.113±0.007	0.330±0.010	0.253±0.009
	$r=2,p=0.3$	**0.087±0.004**	0.090±0.003	0.229±0.010	0.124±0.005	0.264±0.012	0.112±0.008	0.328±0.008	0.249±0.008
	$r=2,p=0.7$	**0.096±0.005**	**0.096±0.004**	0.258±0.008	0.145±0.006	0.275±0.013	0.130±0.005	0.326±0.009	0.264±0.007
	$r=3,p=0.3$	**0.089±0.004**	0.091±0.004	0.242±0.009	0.128±0.005	0.268±0.012	0.114±0.007	0.329±0.008	0.253±0.007
	$r=3,p=0.7$	**0.101±0.001**	**0.101±0.005**	0.278±0.009	0.159±0.007	0.279±0.013	0.142±0.010	0.318±0.008	0.270±0.009
Mirfickr5k	$r=1,p=0.3$	**0.102±0.007**	0.113±0.009	0.140±0.010	0.138±0.007	0.193±0.008	0.148±0.007	0.225±0.009	0.189±0.010
	$r=1,p=0.7$	**0.104±0.008**	0.116±0.009	0.142±0.010	0.139±0.007	0.194±0.007	0.190±0.007	0.224±0.009	0.191±0.010
	$r=2,p=0.3$	**0.104±0.007**	0.116±0.009	0.142±0.009	0.140±0.007	0.193±0.008	0.167±0.009	0.224±0.009	0.191±0.010
	$r=2,p=0.7$	**0.108±0.008**	0.124±0.009	0.149±0.009	0.142±0.007	0.194±0.008	0.227±0.006	0.223±0.009	0.196±0.010
	$r=3,p=0.3$	**0.105±0.008**	0.118±0.009	0.144±0.009	0.142±0.007	0.194±0.007	0.176±0.008	0.223±0.009	0.192±0.009
	$r=3,p=0.7$	**0.110±0.007**	0.135±0.009	0.155±0.010	0.145±0.008	0.193±0.007	0.250±0.009	0.216±0.010	0.200±0.010

- FPML Yu et al. [2018] where the noisy labels estimation is utilized to learn from partial multi-label samples via low-rank approximation [recommended configuration: $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 10$].
- F2L2IF Zhu et al. [2015] where the block-row sparse regularization is adopted to learn a shared subspace [recommended configuration: grid search for $\lambda_1, \lambda_2 \in \{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$].

we employ five popular multi-label metrics Zhang and Zhou [2013] for performance evaluation, i.e., *Ranking Loss*, *Coverage*, *Average Precision*, *Hamming Loss* and *One Error*. Specifically, for *Average Precision*, the greater, the better, while for others the opposite. Ten-fold cross-validation is performed on every dataset, and the mean ± std values are cataloged for each comparative approach. Details on the experimental settings can be found in Appendix.

4.2 EXPERIMENTAL RESULTS

The experimental results are displayed in Tables 2, 3. For the *Ranking Loss*, VADIS demonstrates outstanding performance across all datasets. Regarding *Average Precision*, our method achieves the best performance in 27 out of 30 settings. Specifically, for *Average Precision*, it outperforms others in datasets `EspGame5k`, `Pascal` and `Mirfickr5k`. More results on other metrics can be found in Appendix.

To analyze the relative performance among the comparing approaches in a systematic way, Friedman test Demšar [2006] is employed for the test of performance comparison. Table 4 reports the Friedman statistics F_F and the corresponding critical value in terms of each evaluation metric. It is obvious that the null hypothesis of equal performance is rejected at the significance level of 0.05. Accordingly, posthoc *Boferroni-Dunn test* Demšar [2006] is performed to compare the relative performance among the comparing approaches. Here, VADIS is treated as the control approach where the difference of average rank (over all data sets) between VADIS and one comparing approach is calibrated with critical difference (CD). The critical difference (CD) diagrams Demšar [2006] are presented in Figure 2, where the average rank of each approach is marked along the axis (the smaller the better). Based on the results, it is obvious to observe that:

- As illustrated in Figure 2, VADIS demonstrates superior performance compared to other comparative approaches. Furthermore, VADIS achieves the lowest average rank across all evaluation metrics.
- When comparing VADIS to the MVPML approach GRADIS, VADIS exhibits superior performance in terms of *Ranking Loss*, *Hamming Loss*, *Average Precision*, and *Coverage*. Similarly, when compared to the MVPML approach FIMAN, VADIS achieves superior results in *Average Precision*, *Hamming Loss*, and *Coverage*. Furthermore, in comparison to the MVPML approach GLADE, VADIS outperforms MVPML approach GLADE in terms

Table 3: Predictive performance of each comparing method on five datasets in terms of *Average Precision* (mean ± std). The best performance is marked in bold (the larger the better).

Datasets	Controlling Parameters	Comparing Approaches							
		VADIS	GLADE	F2L2IF	FIMAN	FPML	GRADIS	PML-LRS	LSAMML
Emotions	$r=1,p=0.3$	**0.821±0.032**	0.800±0.031	0.725±0.030	0.792±0.028	0.748±0.027	0.806±0.027	0.739±0.028	0.785±0.034
	$r=1,p=0.7$	**0.810±0.041**	0.786±0.030	0.725±0.022	0.786±0.032	0.748±0.025	0.801±0.033	0.742±0.019	0.753±0.046
	$r=2,p=0.3$	**0.819±0.032**	0.784±0.032	0.721±0.024	0.792±0.028	0.743±0.025	0.817±0.033	0.739±0.024	0.771±0.029
	$r=2,p=0.7$	0.802±0.032	0.775±0.033	0.720±0.027	0.775±0.018	0.737±0.019	**0.815±0.026**	0.729±0.025	0.752±0.031
	$r=3,p=0.3$	**0.819±0.031**	0.784±0.028	0.724±0.027	0.792±0.024	0.734±0.027	0.807±0.038	0.744±0.022	0.780±0.028
	$r=3,p=0.7$	**0.779±0.035**	0.769±0.043	0.684±0.033	0.772±0.027	0.703±0.027	0.730±0.042	0.654±0.030	0.734±0.034
Yeast	$r=1,p=0.3$	**0.767±0.012**	0.763±0.011	0.597±0.014	0.756±0.013	0.703±0.012	0.764±0.011	0.701±0.012	0.437±0.025
	$r=1,p=0.7$	**0.762±0.008**	0.760±0.012	0.592±0.012	0.753±0.013	0.703±0.013	0.749±0.011	0.702±0.013	0.435±0.029
	$r=2,p=0.3$	**0.767±0.011**	0.763±0.012	0.595±0.011	0.754±0.013	0.703±0.013	0.763±0.013	0.702±0.013	0.440±0.019
	$r=2,p=0.7$	**0.761±0.015**	**0.761±0.012**	0.588±0.012	0.754±0.013	0.702±0.012	0.749±0.012	0.701±0.012	0.427±0.013
	$r=3,p=0.3$	0.764±0.014	0.761±0.011	0.592±0.012	0.754±0.014	0.703±0.013	**0.768±0.013**	0.702±0.012	0.441±0.027
	$r=3,p=0.7$	0.759±0.011	**0.762±0.011**	0.588±0.012	0.752±0.013	0.703±0.013	0.758±0.010	0.702±0.013	0.417±0.021
EspGame5k	$r=1,p=0.3$	**0.425±0.011**	0.415±0.010	0.387±0.007	0.397±0.008	0.267±0.011	0.375±0.008	0.346±0.007	0.378±0.008
	$r=1,p=0.7$	**0.424±0.010**	0.415±0.010	0.379±0.008	0.391±0.006	0.267±0.011	0.349±0.008	0.332±0.011	0.376±0.008
	$r=2,p=0.3$	**0.423±0.012**	0.414±0.010	0.381±0.005	0.393±0.006	0.267±0.011	0.365±0.007	0.332±0.010	0.376±0.009
	$r=2,p=0.7$	**0.420±0.104**	0.412±0.008	0.367±0.007	0.380±0.007	0.266±0.010	0.324±0.005	0.302±0.010	0.371±0.008
	$r=3,p=0.3$	**0.422±0.010**	0.412±0.008	0.375±0.005	0.389±0.007	0.267±0.010	0.358±0.008	0.325±0.011	0.373±0.008
	$r=3,p=0.7$	**0.415±0.011**	0.408±0.011	0.353±0.009	0.371±0.009	0.266±0.011	0.315±0.008	0.295±0.011	0.368±0.007
Pascal	$r=1,p=0.3$	**0.746±0.009**	0.700±0.008	0.548±0.013	0.726±0.009	0.487±0.013	0.719±0.011	0.437±0.010	0.471±0.011
	$r=1,p=0.7$	**0.741±0.010**	0.698±0.009	0.507±0.008	0.700±0.010	0.486±0.019	0.704±0.012	0.429±0.011	0.466±0.010
	$r=2,p=0.3$	**0.742±0.009**	0.698±0.010	0.521±0.011	0.713±0.012	0.495±0.013	0.702±0.013	0.433±0.010	0.469±0.011
	$r=2,p=0.7$	**0.734±0.009**	0.693±0.009	0.476±0.009	0.676±0.010	0.476±0.015	0.683±0.010	0.428±0.010	0.458±0.009
	$r=3,p=0.3$	**0.743±0.010**	0.698±0.009	0.505±0.011	0.705±0.011	0.481±0.011	0.697±0.009	0.431±0.009	0.465±0.010
	$r=3,p=0.7$	**0.725±0.007**	0.688±0.009	0.443±0.011	0.644±0.010	0.461±0.012	0.661±0.013	0.429±0.010	0.452±0.010
Mirfickr5k	$r=1,p=0.3$	**0.694±0.015**	0.649±0.014	0.617±0.014	0.654±0.014	0.444±0.010	0.617±0.011	0.417±0.013	0.524±0.016
	$r=1,p=0.7$	**0.692±0.015**	0.649±0.013	0.613±0.014	0.653±0.012	0.441±0.010	0.581±0.013	0.417±0.013	0.522±0.015
	$r=2,p=0.3$	**0.692±0.016**	0.650±0.013	0.614±0.013	0.649±0.013	0.444±0.009	0.605±0.013	0.417±0.013	0.523±0.014
	$r=2,p=0.7$	**0.688±0.015**	0.647±0.013	0.606±0.014	0.646±0.012	0.442±0.010	0.555±0.010	0.418±0.012	0.517±0.017
	$r=3,p=0.3$	**0.690±0.017**	0.649±0.013	0.611±0.013	0.647±0.013	0.444±0.011	0.598±0.011	0.418±0.013	0.522±0.016
	$r=3,p=0.7$	**0.683±0.013**	0.646±0.012	0.598±0.015	0.638±0.012	0.441±0.011	0.543±0.010	0.419±0.013	0.513±0.015

Table 4: Friedman statistics F_F according to each evaluation metric and the critical value at the significance level of 0.05.

Evaluation metric	F_F	Critical value
Ranking Loss	54.3549	2.0549
Coverage	61.9823	
Average Precision	95.7738	
Hamming Loss	23.8863	
One Error	117.9339	

of *Hamming Loss* and *One Error*.

- When compared to the two degenerated MVML approaches, LSAMML and F2L2IF, VADIS consistently achieves superior performance across all evaluation metrics. Moreover, VADIS surpasses both PML-LRS and FPML in terms of all evaluation metrics, which are two degenerated PML approaches.
- As indicated in Tables 2, 3, the performance of VADIS over competing techniques remains consistent across various controlling parameter choices for p and r.

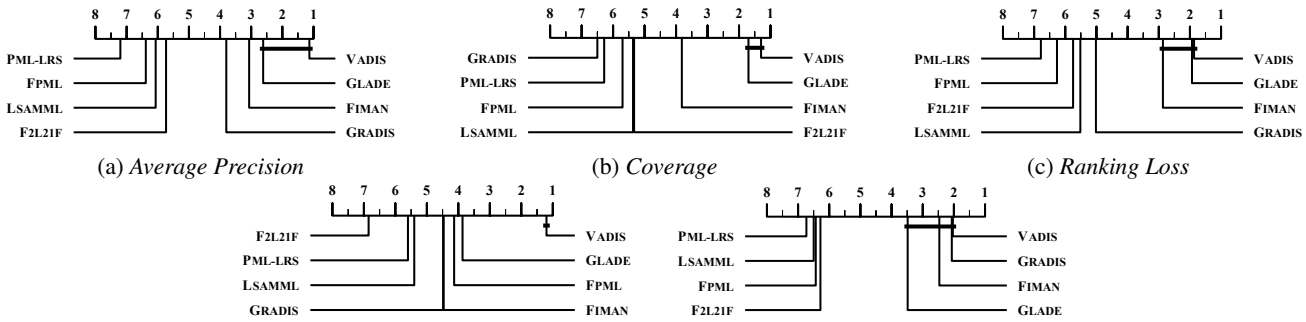
4.3 SENSITIVITY ANALYSIS

In this section, we delve into an analysis of VADIS's performance sensitivity concerning its parameters. Figure 3 serves as a visual representation of VADIS's performance across different parameter settings on Emotions and Yeast datasets, specifically measured in terms of Average Precision. It's worth noting that similar patterns of performance are observed across other datasets as well.

Table 5: The reliability of the inconsistent representations (with controlling configuration: $p = 0.7, r = 3$) on `Emotions` and `Yeast`. The best performance is marked in bold (↓ / ↑ indicates the smaller / larger, the better).

Evaluation metric	Emotions		Yeast	
	VADIS	VADIS-C	VADIS	VADIS-C
Ranking Loss ↓	**0.186**	0.220	**0.176**	0.246
Coverage ↓	**0.319**	0.351	**0.472**	0.564
Average Precision ↑	**0.779**	0.744	**0.759**	0.670
Hamming Loss ↓	**0.300**	0.349	**0.226**	0.306
One Error ↓	**0.230**	0.248	**0.206**	0.240

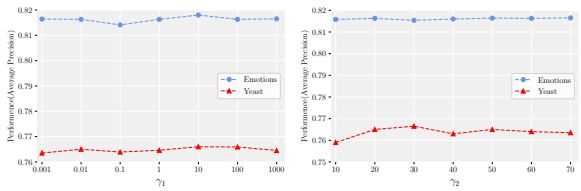
One notable observation is the consistent stability of VADIS's performance across a broad spectrum of parameter values. This stability is a crucial feature as it enables the robust application of VADIS without the necessity for fine-tuning its parameters. Consequently, this characteristic ensures the delivery of reliable classification results, enhancing the utility and effectiveness of VADIS in practical applications.



(a) *Average Precision* (b) *Coverage* (c) *Ranking Loss*

(d) *Hamming Loss* (e) *One Error*

Figure 2: Comparison of VADIS (control algorithm) with other comparative approaches using *Bonferroni-Dunn test*. In the CD diagrams, approaches not connected to VADIS are regarded to perform substantially differently than the control algorithm (CD=1.7013 at the significance level of 0.05).



(a) Varying γ_1 (b) Varying γ_2

Figure 3: Performance sensitivity analysis on `Emotions` and `Yeast` (with controlling configuration: $p = 0.3, r = 1$): (a) Performance of VADIS changes as γ_1 increases from 0.001 to 1000 (γ_2 is fixed as 15); (b) Performance of VADIS changes as γ_2 increases from 10 to 70 (γ_1 is fixed as 5).

4.4 ABLATION STUDY

VADIS leverages the feature representation biases that underlie the generation of incorrect labels in the candidate label set, enabling the identification of true labels through the learning of view-specific-inconsistent representations for disambiguation. To show the reliability of the inconsistent representations, a variant of VADIS-C is investigated. Table 5 reports the detailed results of VADIS and VADIS-C in terms of each evaluation metric on `Emotions` and `Yeast`. These results substantiate the efficacy of the view-specific-inconsistent representations in VADIS.

In addition, we have used the local semantic similarity for aligning with the global consistency representations. so we have added more ablation experiments to explain the effects. VADIS-S refers to the model performance after removing the use of the local similarity matrix for aligning global consistency representations. The detailed experimental results on `Emotions` and `Yeast` in terms of five metrics are reported in Table 6.

Additionally, for the learning of false-positive labels, we analyze the false-positive -label learning by comparing the estimated false-positive label $\mathbf{OQ}$ with the generated false-positive label in the *emotions*. We use a vanilla variant of VADIS (dubbed VADIS-O) that eliminates the regularizer of $\mathbf{Q}$ and performs the estimation of $\mathbf{OQ}$ in Table 7.

Table 6: The effect of local semantic similarity aligning (with controlling configuration: $p = 0.7, r = 3$) on `Emotions` and `Yeast`. The best performance is marked in bold (↓ / ↑ indicates the smaller / larger, the better).

Evaluation metric	Emotions		Yeast	
	VADIS	VADIS-S	VADIS	VADIS-S
Ranking Loss ↓	**0.186**	0.214	**0.176**	0.218
Coverage ↓	**0.319**	0.339	**0.472**	0.502
Average Precision ↑	**0.779**	0.757	**0.759**	0.734
Hamming Loss ↓	**0.300**	0.310	**0.226**	0.254
One Error ↓	**0.230**	0.242	**0.206**	0.222

Table 7: The estimation of false-positive labels on `Emotions` during iterations (with controlling configuration: $p = 0.7, r = 3$) on `Emotions`.

Method	Metrics	Iteration								
		1	5	10	15	20	25	30	35	40
VADIS	*Hamming loss*	0.265	0.263	0.268	0.266	0.265	0.265	0.263	0.261	0.261
	Average precision	0.543	0.547	0.571	0.580	0.588	0.592	0.595	0.596	0.596
VADIS-O	*Hamming loss*	0.294	0.277	0.278	0.273	0.271	0.270	0.269	0.270	0.269
	Average precision	0.589	0.573	0.571	0.570	0.572	0.570	0.571	0.570	0.570

5 CONCLUSION

In this paper, we introduce a novel approach called VADIS to investigate feature representation biases by leveraging the characteristics of various views for identifying true labels. Specifically, we employ the global common representation, corresponding to the local similarity matrix in the semantic space, to estimate true labels using a low-rank mapping matrix. Additionally, for identifying incorrect labels, we recover the view-specific inconsistent representation based on the sparsity assumption. A substantial number of experiments have demonstrated the effectiveness of VADIS.

Author Contributions

Briefly list author contributions. This is a nice way of making clear who did what and to give proper credit. This section is optional.

H. Q. Bovik conceived the idea and wrote the paper. Coauthor One created the code. Coauthor Two created the figures.

Acknowledgements

Briefly acknowledge people and organizations here.

All acknowledgements go in this section.

References

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

Ze-Sen Chen, Xuan Wu, Qing-Guo Chen, Yao Hu, and Min-Ling Zhang. Multi-view partial multi-label learning with graph-based disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3553–3560, 2020.

Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. *Advances in neural information processing systems*, 14:681–687, 2001.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 39–43, 2008.

Liping Liu and Thomas Dietterich. A conditional multinomial mixture model for superset label learning. *Advances in neural information processing systems*, 25:557–565, 2012.

Meng Liu, Yong Luo, Dacheng Tao, Chao Xu, and Yonggang Wen. Low-rank multi-view learning in matrix completion for multi-label image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pages 2778–2784, 2015.

Yong Luo, Dacheng Tao, Chang Xu, Chao Xu, Hong Liu, and Yonggang Wen. Multiview vector-valued manifold regularization for multilabel image classification. *IEEE transactions on neural networks and learning systems*, 24 (5):709–722, 2013.

Lijuan Sun, Songhe Feng, Tao Wang, Congyan Lang, and Yi Jin. Partial multi-label learning by low-rank and sparse decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5016–5023, 2019.

Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, Ioannis P Vlahavas, et al. Multi-label classification of music into emotions. In *ISMIR*, volume 8, pages 325–330, 2008.

Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, 2004.

Deng-Bao Wang, Li Li, and Min-Ling Zhang. Adaptive graph guided disambiguation for partial label learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 83–91, 2019.

Jing-Han Wu, Xuan Wu, Qing-Guo Chen, Yao Hu, and Min-Ling Zhang. Feature-induced manifold disambiguation for multi-view partial multi-label learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 557–565, 2020.

Xuan Wu, Qing-Guo Chen, Yao Hu, Dengbao Wang, Xiaodong Chang, Xiaobo Wang, and Min-Ling Zhang. Multi-view multi-label learning with view-specific information extraction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3884–3890, 2019.

Ming-Kun Xie and Sheng-Jun Huang. Partial multi-label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 4302–4309, 2018.

Ming-Kun Xie and Sheng-Jun Huang. Partial multi-label learning with noisy label identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (7):3676–3687, 2021.

Yuying Xing, Guoxian Yu, Carlotta Domeniconi, Jun Wang, and Zili Zhang. Multi-label co-training. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2882–2888, 2018.

Ning Xu, Yun-Peng Liu, and Xin Geng. Partial multi-label learning with label distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6510–6517, 2020.

Ning Xu, Yong-Di Wu, Congyu Qiao, Yi Ren, Minxue Zhang, and Xin Geng. Multi-view partial multi-label learning via graph-fusion-based label enhancement. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. In *Asian conference on machine learning*, pages 96–111, 2016.

Guoxian Yu, Xia Chen, Carlotta Domeniconi, Jun Wang, Zhao Li, Zili Zhang, and Xindong Wu. Feature-induced partial multi-label learning. In *2018 IEEE International Conference on Data Mining*, pages 1398–1403, 2018.

Wang Zhan and Min-Ling Zhang. Inductive semi-supervised multi-label learning with co-training. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1305–1314, 2017.

Changqing Zhang, Ziwei Yu, Qinghua Hu, Pengfei Zhu, Xinwang Liu, and Xiaobo Wang. Latent semantic aware multi-view multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 4414–4421, 2018.

Min-Ling Zhang and Jun-Peng Fang. Partial multi-label learning via credible label elicitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10): 3587–3599, 2020.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2013.

Zhi-Hua Zhou and Ming Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24: 415–439, 2010.

Pengfei Zhu, Qi Hu, Qinghua Hu, Changqing Zhang, and Zhizhao Feng. Multi-view label embedding. *Pattern recognition*, 84:126–135, 2018.

Xiaofeng Zhu, Xuelong Li, and Shichao Zhang. Block-row sparse multiview multilabel learning for image classification. *IEEE transactions on cybernetics*, 46(2):450–461, 2015.

Liansheng Zhuang, Haoyuan Gao, Zhouchen Lin, Yi Ma, Xin Zhang, and Nenghai Yu. Non-negative low rank and sparse graph for semi-supervised learning. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2328–2335, 2012.

Supplementary Material of VADIS: Investigating Inter-View Representation Biases for Multi-View Partial Multi-Label Learning

Jie Wang[1] **Ning Xu**[1] **Xin Geng**[1]

[1]School of Computer Science and Engineering, Southeast University, Nanjing, China

A APPENDIX

In this section, we will add some content from the main text section.

A.1 DETAILS OF DATASETS

In this paper, details of the dataset are as follows: For the `Emotions` dataset, the two views per sample equivalent to the rhythm features and timbre features of a piece of music; For the `EspGame5k` dataset, the four views of each sample equivalent to the DenseHue, Gist, DenseSift, and HSV features of an image; In addition to the four views used by `EspGame5k`, the `Pascal` and `Mirflickr5k` datasets adopt the tag features to represent each sample; For the `Yeast` dataset, the two views of each sample equivalent to the genetic expression and phylogeny profile of a gene.

Furthermore, the specific symbols are explained below:

- $|\mathcal{S}|$: shows the number of samples in each dataset.
- $V(\mathcal{S})$: shows the number and the details of views in each dataset.
- $V\,\mathrm{Dim}(\mathcal{S})$: shows the dimensionality of each view in each dataset.
- $CL(\mathcal{S})$: shows the number of class labels in each dataset.
- $LCard(\mathcal{S})$: shows the average number of ground-truth labels corresponding to each sample (i.e. label cardinality) in each dataset.
- *Domain*: shows the domain associated with each view in each dataset.
- *Description*: shows the feature description associated with each view in each dataset.

A.2 PSEUDO-CODE OF VADIS

The complete pseudo-code of the algorithm is shown in Algorithm 1.

A.3 DETAILS OF EXPERIMENTAL SETTINGS

The code implementation is based on PyTorch, and all the experiments are conducted on NVIDIA RTX 3090. In the experiments, we set the value of the K-nearest neighbor parameter, K, to 10. As for the control parameters γ_1 and γ_2, we consistently fix them at 5 and 15, respectively, across all experiments. Moreover, the number of iterations for optimization in the experiments is fixed at 200 times. The stopping threshold for the optimization algorithm is set to 1e-4. The pseudo-code for VADIS is shown in Algorithm 1, where the initialization of $\mathbf{P}$ and $\mathbf{Q}$ employs torch.rand, while the initialization of $\mathbf{W}$ and $\mathbf{C}$ adopts torch.zeros. Ten-fold cross-validation is conducted on each dataset, and the mean and the standard deviation (mean $\pm$ std) values are recorded for each comparative approach.

Algorithm 1 The pseudo-code of VADIS

Input: the multi-view partial multi-label training set $\mathcal{D}$, the trade-off parameters γ_1, γ_2 in objective function Eq. (7), the local similarity of the v-th view S^v ($1 \leq v \leq V$), the unseen instance $\boldsymbol{x}$;

Calculate the self-representation matrix $\mathbf{Z}^v$ by Eq. (1) and initialize $\mathbf{C}^{(0)}, \mathbf{O}^{v(0)}, \mathbf{P}^{(0)}, \mathbf{Q}^{(0)}$ and $\mathbf{W}^{(0)}$;
repeat
    Update $\mathbf{C}^{(t+1)}$ by solving problem Eq. (8) with the BFGS strategy;
    Update $\mathbf{W}^{(t+1)}$ by solving problem Eq. (9) with the ridge regression problem;
    Update $\mathbf{P}^{(t+1)}$ by solving problem Eq. (12) with the Alternating Direction Method of Multiplier;
    Update $\mathbf{Q}^{(t+1)}$ by solving problem Eq. (15) with the Alternating Direction Method of Multiplier;
    Update $\mathbf{O}^{v(t+1)}$ by solving problem Eq. (17) with the Alternating Direction Method of Multiplier;
    $t = t + 1$;
until convergence
The final predictive model is obtained by setting $\mathbf{W}^* = \mathbf{W}^{(t)}$;
Return a proper label set Y according to Eq. (19).

Output: the predicted label set Y for $\boldsymbol{x}$

A.4 MORE RESULTS OF MVPML DATASETS

Tables 8 ,9 and 10 below show the predictive performance of VADIS with other comparing methods on five datasets in terms of other evaluation metrics (i.e, *Hamming Loss, Coverage and One Error*). The best performance is marked in bold (the smaller the better).

Compared to the comparative methods in terms of *Hamming Loss*, our approach has delivered comparable performance across all datasets. Similarly, in the CD Figures (b) and (d) in our paper, both *Coverage* and *Hamming Loss* metrics reflect the excellent performance of VADIS. Regarding *One-error*, VADIS exhibits strong performance on the `EspGame5k`, `Pascal`, and `Mirflickr5k` datasets, and in CD Figure (e) in the paper, our method is positioned furthest to the right, indicating overall superior performance compared to other approaches.

A.5 TIME COMPLEXITY ANALYSIS

What's more, we discuss the time complexity of our method, which encompasses two aspects: the initiation procedure and the optimization procedure. The cost of initializing $\mathbf{C}^{(0)}$, $\mathbf{P}^{(0)}$, $\mathbf{Q}^{(0)}$, $\mathbf{W}^{(0)}$, and $\mathbf{O}^{(0)}$ is $O\left(n^2 + 2nc + c\sum_{v=1}^{V} d_v + Vn^2\right)$. Then the time consumption of updating $\mathbf{C}$ is $O\left(T1\left(n^2c + \sum_{v=1}^{V} d_v\, n^2\right)\right)$,where $T1$ is the iteration number of BFGS in this step. The time consumption of updating $\mathbf{W}$ is $O\left(T2\left(nc\, \sum_{v=1}^{V} d_v\right)\right)$, where $T2$ is the iteration number of ridge regression problem in this step. The time consumption of updating $\mathbf{P}, \mathbf{Q}, and \mathbf{O}^v$ are $O\left(T3\left(n^2c\right)\right), O\left(T4\left(n^2c\right)\right)$, and $O\left(T5\left(\sum_{v=1}^{V} d_v n^2 + n^2c\right)\right)$ respectively, where $T3,\ T4,\ and\ T5$ are the iteration number of Alternating Direction Method of Multiplier in this step. Therefore, the total time complexity is the merging of the above components.

Table 8: Predictive performance of each comparing method on five datasets in terms of *Hamming Loss* (mean ± std). The best performance is marked in bold (the smaller the better).

Datasets	Controlling Parameters	Comparing Approaches							
		VADIS	GLADE	F2L2IF	FIMAN	FPML	GRADIS	PML-LRS	LSAMML
Emotions	$r=1,p=0.3$	**0.195±0.018**	0.207±0.017	0.422±0.024	0.228±0.019	0.244±0.013	0.218±0.016	0.256±0.018	0.214±0.022
	$r=1,p=0.7$	**0.204±0.027**	0.208±0.018	0.425±0.017	0.235±0.016	0.236±0.015	0.272±0.021	0.237±0.018	0.264±0.035
	$r=2,p=0.3$	**0.193±0.022**	0.206±0.022	0.428±0.022	0.227±0.017	0.233±0.016	0.258±0.019	0.243±0.017	0.235±0.024
	$r=2,p=0.7$	**0.214±0.018**	0.227±0.015	0.440±0.024	0.238±0.011	0.404±0.033	0.358±0.022	0.400±0.024	0.333±0.026
	$r=3,p=0.3$	**0.199±0.017**	0.212±0.016	0.430±0.025	0.233±0.019	0.250±0.017	0.290±0.037	0.236±0.018	0.263±0.020
	$r=3,p=0.7$	**0.230±0.017**	0.236±0.019	0.450±0.030	0.248±0.015	0.679±0.012	0.353±0.025	0.675±0.012	0.536±0.026
Yeast	$r=1,p=0.3$	**0.203±0.010**	0.229±0.012	0.310±0.009	0.211±0.008	0.232±0.008	0.209±0.007	0.236±0.007	0.303±0.005
	$r=1,p=0.7$	**0.204±0.010**	0.234±0.010	0.314±0.010	0.211±0.007	0.232±0.008	0.235±0.007	0.253±0.007	0.303±0.004
	$r=2,p=0.3$	**0.204±0.010**	0.230±0.009	0.315±0.009	0.213±0.008	0.232±0.008	0.224±0.010	0.250±0.007	0.303±0.005
	$r=2,p=0.7$	**0.206±0.009**	0.233±0.012	0.317±0.008	0.211±0.008	0.236±0.011	0.273±0.011	0.377±0.008	0.303±0.005
	$r=3,p=0.3$	**0.203±0.008**	0.228±0.011	0.313±0.009	0.209±0.007	0.232±0.008	0.244±0.011	0.279±0.010	0.303±0.004
	$r=3,p=0.7$	**0.206±0.011**	0.239±0.009	0.317±0.009	0.211±0.008	0.254±0.010	0.312±0.009	0.590±0.007	0.303±0.004
EspGame5k	$r=1,p=0.3$	**0.053±0.001**	**0.053±0.001**	0.081±0.001	0.111±0.002	**0.053±0.001**	0.057±0.001	**0.053±0.001**	0.056±0.002
	$r=1,p=0.7$	**0.053±0.001**	**0.053±0.001**	0.082±0.001	0.119±0.003	**0.053±0.001**	0.063±0.001	**0.053±0.001**	0.056±0.002
	$r=2,p=0.3$	**0.053±0.001**	**0.053±0.001**	0.082±0.001	0.116±0.003	**0.053±0.001**	0.061±0.001	**0.053±0.001**	0.056±0.002
	$r=2,p=0.7$	**0.053±0.001**	**0.053±0.001**	0.083±0.002	0.130±0.004	**0.053±0.001**	0.075±0.001	**0.053±0.001**	0.057±0.002
	$r=3,p=0.3$	**0.053±0.001**	**0.053±0.001**	0.083±0.001	0.118±0.003	**0.053±0.001**	0.067±0.001	**0.053±0.001**	0.056±0.002
	$r=3,p=0.7$	**0.053±0.001**	**0.053±0.001**	0.085±0.001	0.141±0.004	**0.053±0.001**	0.087±0.002	**0.053±0.001**	0.058±0.002
Pascal	$r=1,p=0.3$	**0.051±0.002**	0.121±0.004	0.174±0.003	0.109±0.004	0.067±0.001	**0.051±0.001**	0.083±0.007	0.074±0.001
	$r=1,p=0.7$	**0.052±0.002**	0.129±0.005	0.176±0.002	0.132±0.003	0.067±0.002	**0.052±0.002**	0.082±0.007	0.075±0.001
	$r=2,p=0.3$	**0.052±0.002**	0.126±0.005	0.175±0.002	0.121±0.005	0.067±0.001	**0.052±0.001**	0.083±0.007	0.075±0.001
	$r=2,p=0.7$	**0.053±0.002**	0.143±0.005	0.177±0.002	0.161±0.005	0.068±0.001	**0.053±0.001**	0.087±0.007	0.077±0.001
	$r=3,p=0.3$	**0.053±0.002**	0.131±0.004	0.176±0.002	0.127±0.005	0.067±0.001	**0.053±0.001**	0.084±0.006	0.076±0.001
	$r=3,p=0.7$	**0.055±0.002**	0.155±0.005	0.178±0.002	0.191±0.009	0.069±0.001	**0.055±0.001**	0.108±0.009	0.080±0.001
Mirfickr5k	$r=1,p=0.3$	**0.107±0.003**	0.163±0.006	0.121±0.002	0.112±0.003	0.124±0.004	0.111±0.004	0.186±0.004	0.125±0.004
	$r=1,p=0.7$	**0.108±0.004**	0.167±0.006	0.122±0.003	0.114±0.003	0.124±0.004	0.122±0.003	0.214±0.003	0.126±0.004
	$r=2,p=0.3$	**0.107±0.003**	0.166±0.006	0.121±0.002	0.113±0.003	0.124±0.004	0.120±0.004	0.208±0.003	0.125±0.004
	$r=2,p=0.7$	**0.110±0.004**	0.170±0.006	0.122±0.002	0.117±0.003	0.124±0.004	0.142±0.004	0.271±0.003	0.127±0.004
	$r=3,p=0.3$	**0.109±0.003**	0.166±0.006	0.122±0.003	0.114±0.003	0.124±0.004	0.129±0.005	0.231±0.002	0.126±0.004
	$r=3,p=0.7$	**0.112±0.004**	0.174±0.006	0.124±0.002	0.120±0.003	0.124±0.004	0.162±0.003	0.353±0.006	0.129±0.004

Table 9: Predictive performance of each comparing method on five datasets in terms of *Coverage* (mean ± std). The best performance is marked in bold (the smaller the better).

Datasets	Controlling Parameters	Comparing Approaches							
		VADIS	GLADE	F2L2IF	FIMAN	FPML	GRADIS	PML-LRS	LSAMML
Emotions	$r=1,p=0.3$	**0.284±0.023**	0.300±0.023	0.357±0.029	0.314±0.026	0.350±0.031	0.372±0.032	0.346±0.028	0.311±0.024
	$r=1,p=0.7$	**0.291±0.022**	0.300±0.023	0.358±0.021	0.323±0.021	0.350±0.027	0.494±0.030	0.345±0.025	0.340±0.043
	$r=2,p=0.3$	**0.288±0.023**	0.308±0.025	0.364±0.027	0.315±0.026	0.356±0.032	0.405±0.022	0.356±0.028	0.323±0.025
	$r=2,p=0.7$	**0.313±0.030**	0.324±0.028	0.385±0.024	0.335±0.020	0.370±0.026	0.593±0.029	0.384±0.034	0.344±0.030
	$r=3,p=0.3$	**0.284±0.025**	0.307±0.027	0.363±0.030	0.313±0.026	0.359±0.029	0.432±0.047	0.356±0.031	0.323±0.025
	$r=3,p=0.7$	**0.319±0.024**	0.320±0.019	0.400±0.025	0.321±0.023	0.394±0.018	0.655±0.030	0.430±0.020	0.352±0.025
Yeast	$r=1,p=0.3$	**0.451±0.014**	0.455±0.013	0.644±0.012	0.481±0.016	0.485±0.012	0.516±0.014	0.492±0.012	0.747±0.016
	$r=1,p=0.7$	**0.460±0.015**	**0.460±0.013**	0.648±0.013	0.489±0.016	0.486±0.013	0.606±0.008	0.492±0.012	0.748±0.023
	$r=2,p=0.3$	**0.458±0.014**	0.461±0.013	0.646±0.012	0.486±0.016	0.485±0.013	0.545±0.018	0.492±0.012	0.750±0.020
	$r=2,p=0.7$	0.471±0.013	**0.460±0.012**	0.655±0.012	0.489±0.020	0.487±0.013	0.671±0.013	0.494±0.012	0.748±0.018
	$r=3,p=0.3$	**0.462±0.015**	**0.462±0.014**	0.646±0.013	0.484±0.018	0.484±0.013	0.561±0.021	0.490±0.012	0.745±0.015
	$r=3,p=0.7$	**0.472±0.014**	**0.472±0.013**	0.649±0.013	0.488±0.017	0.486±0.013	0.708±0.015	0.487±0.012	0.752±0.018
EspGame5k	$r=1,p=0.3$	**0.379±0.008**	0.380±0.010	0.456±0.011	0.446±0.011	0.526±0.012	0.482±0.011	0.462±0.012	0.444±0.007
	$r=1,p=0.7$	**0.383±0.007**	0.387±0.009	0.464±0.010	0.453±0.009	0.527±0.012	0.569±0.010	0.472±0.013	0.445±0.005
	$r=2,p=0.3$	**0.381±0.008**	0.385±0.010	0.461±0.011	0.453±0.009	0.527±0.013	0.521±0.012	0.469±0.015	0.450±0.009
	$r=2,p=0.7$	**0.396±0.009**	0.400±0.009	0.481±0.012	0.470±0.014	0.527±0.013	0.657±0.004	0.484±0.013	0.460±0.009
	$r=3,p=0.3$	**0.387±0.011**	0.391±0.011	0.468±0.012	0.456±0.011	0.526±0.012	0.544±0.008	0.477±0.011	0.449±0.004
	$r=3,p=0.7$	**0.403±0.012**	0.405±0.011	0.482±0.013	0.471±0.015	0.527±0.016	0.698±0.013	0.486±0.012	0.459±0.008
Pascal	$r=1,p=0.3$	**0.129±0.005**	0.132±0.003	0.277±0.010	0.171±0.006	0.343±0.011	0.152±0.010	0.387±0.007	0.302±0.007
	$r=1,p=0.7$	**0.134±0.005**	0.135±0.005	0.297±0.009	0.188±0.008	0.341±0.014	0.164±0.008	0.389±0.010	0.312±0.009
	$r=2,p=0.3$	**0.132±0.005**	0.134±0.004	0.290±0.013	0.181±0.006	0.337±0.012	0.165±0.010	0.388±0.011	0.307±0.007
	$r=2,p=0.7$	**0.140±0.007**	0.142±0.006	0.317±0.009	0.204±0.008	0.348±0.013	0.184±0.005	0.390±0.010	0.327±0.007
	$r=3,p=0.3$	**0.135±0.005**	**0.135±0.005**	0.302±0.011	0.186±0.007	0.343±0.014	0.166±0.008	0.391±0.010	0.312±0.006
	$r=3,p=0.7$	**0.146±0.009**	0.148±0.007	0.337±0.010	0.219±0.008	0.355±0.014	0.197±0.011	0.390±0.007	0.332±0.009
Mirfickr5k	$r=1,p=0.3$	**0.304±0.016**	0.310±0.016	0.356±0.016	0.396±0.020	0.419±0.015	0.412±0.015	0.438±0.015	0.418±0.018
	$r=1,p=0.7$	**0.310±0.016**	0.314±0.017	0.363±0.014	0.397±0.019	0.421±0.014	0.534±0.008	0.436±0.015	0.423±0.017
	$r=2,p=0.3$	**0.310±0.014**	0.313±0.015	0.362±0.015	0.401±0.018	0.420±0.015	0.458±0.016	0.437±0.015	0.423±0.017
	$r=2,p=0.7$	**0.319±0.017**	**0.319±0.016**	0.378±0.014	0.405±0.017	0.421±0.015	0.621±0.015	0.437±0.014	0.434±0.017
	$r=3,p=0.3$	0.322±0.016	**0.316±0.016**	0.367±0.014	0.405±0.020	0.420±0.014	0.478±0.022	0.436±0.015	0.426±0.015
	$r=3,p=0.7$	0.326±0.015	**0.323±0.015**	0.389±0.016	0.408±0.021	0.422±0.014	0.663±0.018	0.435±0.015	0.441±0.016

Table 10: Predictive performance of each comparing method on five datasets in terms of *One Error* (mean ± std). The best performance is marked in bold (the smaller the better).

Datasets	Controlling Parameters	Comparing Approaches							
		VADIS	GLADE	F2L2IF	FIMAN	FPML	GRADIS	PML-LRS	LSAMML
Emotions	$r=1,p=0.3$	**0.231±0.057**	0.273±0.056	0.396±0.054	0.271±0.060	0.340±0.046	0.238±0.048	0.368±0.036	0.293±0.057
	$r=1,p=0.7$	0.258±0.076	0.314±0.057	0.391±0.050	0.284±0.065	0.337±0.049	**0.219±0.064**	0.363±0.024	0.341±0.078
	$r=2,p=0.3$	**0.220±0.054**	0.309±0.060	0.400±0.052	0.276±0.052	0.345±0.045	0.229±0.055	0.362±0.035	0.323±0.059
	$r=2,p=0.7$	0.253±0.062	0.317±0.073	0.388±0.069	0.295±0.038	0.359±0.059	**0.201±0.058**	0.358±0.044	0.342±0.042
	$r=3,p=0.3$	0.244±0.067	0.304±0.051	0.398±0.055	0.283±0.039	0.373±0.050	**0.176±0.049**	0.354±0.030	0.300±0.038
	$r=3,p=0.7$	0.300±0.054	0.331±0.090	0.432±0.057	0.320±0.048	0.386±0.053	**0.189±0.056**	0.434±0.044	0.378±0.057
Yeast	$r=1,p=0.3$	**0.211±0.028**	0.223±0.016	0.386±0.028	0.216±0.025	0.249±0.022	0.214±0.019	0.249±0.022	0.685±0.056
	$r=1,p=0.7$	0.213±0.019	0.224±0.017	0.395±0.026	0.218±0.024	0.251±0.022	**0.209±0.016**	0.249±0.022	0.694±0.068
	$r=2,p=0.3$	**0.205±0.031**	0.220±0.021	0.383±0.018	0.220±0.026	0.250±0.022	0.206±0.023	0.250±0.022	0.677±0.039
	$r=2,p=0.7$	0.231±0.030	0.224±0.018	0.401±0.025	0.223±0.025	0.250±0.021	**0.201±0.024**	0.249±0.022	0.705±0.033
	$r=3,p=0.3$	0.224±0.029	0.224±0.022	0.386±0.025	0.220±0.027	0.249±0.021	**0.198±0.023**	0.249±0.022	0.681±0.062
	$r=3,p=0.7$	0.226±0.018	0.220±0.019	0.401±0.031	0.222±0.022	0.249±0.022	**0.192±0.016**	0.249±0.022	0.736±0.047
EspGame5k	$r=1,p=0.3$	**0.549±0.027**	0.557±0.022	0.574±0.014	0.560±0.023	0.758±0.019	0.560±0.023	0.633±0.018	0.586±0.014
	$r=1,p=0.7$	**0.554±0.022**	0.555±0.021	0.583±0.014	0.567±0.017	0.758±0.019	0.560±0.018	0.670±0.023	0.588±0.015
	$r=2,p=0.3$	**0.550±0.023**	0.555±0.024	0.582±0.010	0.559±0.021	0.758±0.019	0.558±0.019	0.663±0.016	0.592±0.011
	$r=2,p=0.7$	**0.555±0.211**	0.559±0.025	0.585±0.014	0.567±0.017	0.758±0.019	0.559±0.016	0.703±0.016	0.595±0.016
	$r=3,p=0.3$	**0.556±0.024**	0.560±0.024	0.585±0.015	0.567±0.018	0.758±0.019	0.559±0.019	0.668±0.017	0.590±0.013
	$r=3,p=0.7$	0.560±0.023	0.562±0.026	0.606±0.021	0.584±0.017	0.758±0.019	**0.555±0.017**	0.718±0.021	0.599±0.011
Pascal	$r=1,p=0.3$	**0.302±0.014**	0.390±0.015	0.528±0.017	**0.302±0.014**	0.550±0.018	0.325±0.015	0.575±0.021	0.594±0.017
	$r=1,p=0.7$	**0.306±0.014**	0.390±0.016	0.577±0.010	0.328±0.014	0.553±0.024	0.338±0.018	0.585±0.021	0.597±0.017
	$r=2,p=0.3$	**0.308±0.012**	0.389±0.016	0.561±0.010	0.314±0.018	0.540±0.017	0.342±0.015	0.583±0.021	0.594±0.018
	$r=2,p=0.7$	**0.310±0.012**	0.389±0.018	0.615±0.012	0.354±0.012	0.570±0.023	0.358±0.017	0.584±0.022	0.599±0.015
	$r=3,p=0.3$	**0.305±0.016**	0.388±0.012	0.576±0.011	0.321±0.016	0.559±0.013	0.351±0.014	0.582±0.021	0.597±0.017
	$r=3,p=0.7$	**0.317±0.009**	0.391±0.013	0.654±0.012	0.397±0.014	0.585±0.016	0.384±0.021	0.583±0.021	0.605±0.016
Mirfickr5k	$r=1,p=0.3$	**0.234±0.017**	0.301±0.020	0.301±0.019	**0.234±0.026**	0.564±0.021	0.294±0.020	0.576±0.019	0.429±0.020
	$r=1,p=0.7$	**0.237±0.019**	0.300±0.020	0.305±0.022	**0.237±0.025**	0.567±0.019	0.288±0.020	0.576±0.019	0.428±0.021
	$r=2,p=0.3$	**0.237±0.018**	0.299±0.019	0.305±0.018	**0.237±0.021**	0.563±0.020	0.287±0.020	0.577±0.019	0.424±0.017
	$r=2,p=0.7$	**0.236±0.020**	0.303±0.019	0.311±0.020	**0.236±0.026**	0.562±0.023	0.292±0.017	0.576±0.019	0.434±0.023
	$r=3,p=0.3$	**0.237±0.023**	0.300±0.016	0.307±0.014	0.238±0.026	0.564±0.022	0.289±0.021	0.576±0.019	0.433±0.023
	$r=3,p=0.7$	**0.248±0.020**	0.301±0.018	0.317±0.027	**0.248±0.019**	0.565±0.022	0.289±0.017	0.575±0.019	0.430±0.022