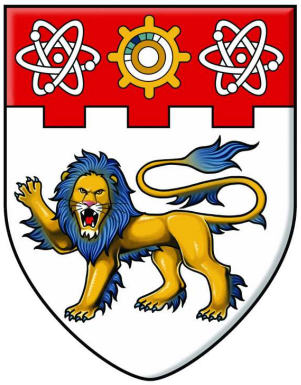


NANYANG TECHNOLOGICAL UNIVERSITY

DOMAIN ADAPTATION AND GENERALIZATION FOR VISUAL RECOGNITION

LI NIU

INTERDISCIPLINARY GRADUATE SCHOOL
RAPID-RICH OBJECT SEARCH (ROSE) LAB

2016

DOMAIN ADAPTATION AND GENERALIZATION FOR VISUAL RECOGNITION

LI NIU

Interdisciplinary Graduate School
Rapid-Rich Object SEarch (ROSE) Lab

A thesis submitted to the Nanyang Technological University in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2016

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

. . . .

Date | Student Name

Abstract

In many visual recognition tasks, the domain distribution mismatch between the training set (*i.e.*, source domain) and the test set (*i.e.*, target domain) may cause the performance of the classifier learnt from the training set to be significantly degraded on the test set. The solutions to address the domain distribution mismatch can be classified into Domain Adaptation (DA) and Domain Generalization (DG). Specifically, DA utilizes the unlabeled target domain data in the training phase to reduce the domain distribution mismatch while DG aims to learn the classifier on the source domain which can generalize well to any unseen target domain. This thesis focuses on DA and DG for visual recognition.

Most of the existing DA and DG approaches require well labeled training data. Since collecting labeled data is often time consuming and expensive, some recent works utilize freely available web images/videos for visual recognition. Therefore, the DA and DG methods can be categorized based on learning from web data or well labeled data.

For learning from web data, besides the data distribution mismatch between the web training data and test data, there also exist some other problems such as label noise of web data and extra information associated with web data (*i.e.*, privileged information). All the existing DA and DG methods only consider the domain distribution mismatch, but ignore the label noise and privileged information. To this end, we propose a DA framework and a DG method for learning from web data, which leads to the first and second work in this thesis respectively.

In the first work, we propose our DA framework named Domain Adaptive Multi-Instance Learning using Privileged Information (MIL-PI-DA) for visual recognition by learning from web data, which can handle the label noise, utilize the privileged information, and reduce the domain distribution mismatch at the same time.

In the second work, we propose our DG method named Weakly Supervised Domain Generalization (WSDG) for visual recognition by learning from web data, which can cope with the label noise, take advantage of the privileged information, and generalize well to any unseen target domain at the same time.

For learning from well labeled data, there are also some issues with the existing DA and DG approaches. One issue is how to utilize multiple types of features in the multi-view scenario. Although multi-view DA has been studied, no multi-view approach has been proposed for DG, which motivates our third work in this thesis. In the third work, we propose a framework named Exemplar-based Multi-View Domain Generalization (EMVDG) for visual recognition based on the consensus principle and complementary principle, which is the first work to explore the DG problem under the multi-view setting.

Another issue when learning from well labeled data is that for global feature representations (*e.g.*, Fisher vector encoded based on Gaussian Mixture Model (GMM)), the codebook (*e.g.*, GMM) learnt on the source domain may not well capture the distribution of the target domain. There is no existing DA approach considering this issue, which motivates our fourth work in this thesis. In the fourth work, we propose a Domain Adaptive method based on Fisher Vector (DAFV) for visual recognition, which is specifically designed for Fisher vector. Our key idea is to reduce the domain distribution mismatch by selecting domain invariant components of Fisher vectors.

For all our proposed DA or DG methods, we conduct extensive experiments and comparisons with the state-of-the-art methods. The experimental results demonstrate the superior performance of our proposed methods under different scenarios.

Acknowledgments

First of all, I would like to express my gratitude to my advisor Prof. Xu Dong and Prof. Jianfei Cai sincerely. They guide me through the whole Ph.D stage about how to do research and how to be a researcher, specifically, how to figure out idea, design experimental settings, conduct comprehensive experiments, write papers, and get papers published. Their patient supervision and rigorous attitude not only make me understand the frontier of computer vision, but also improve the cutting-edge technologies for a variety of applications.

Then, I would like to thank my senior lab mate Dr. Wen Li, who is my main collaborator and provided me lots of valuable help as well as precious advices in the early stage of my Ph.D. We cooperated on many papers and fortunately get all of them published on top conferences or journals. Moreover, I would like to express my thanks to my co-supervisor Prof. Junsong Yuan, mentor Prof. Jialin Pan, and TAC committee member Prof. Yap Kim Hui for their informed comments and enlightening insights, which contribute a lot to enhancing the quality of this thesis.

I also appreciate the accompany of all my friends and the staffs in the Rapid Rich Object Search (ROSE) lab and Centre for Multimedia and Network Technology (CeMNet) of the Nanyang Technological University for their daily care and technical support. Special thanks to my friendly and considerate roommates, Guanyu Gao and Kai Qian, without whom I can hardly finish my Ph.D journey and manage to submit my Ph.D thesis in time.

Last but not the least, I want to express my genuine thanks to my parents and my girlfriend because they always stand on my side with generous encouragement and support.

Table of Contents

Table Captions

Figure Captions

Abbreviations

In this thesis, we use lowercase character (*i.e.*, a) to represent scalar, bold lowercase character (*i.e.*, $\mathbf{a}$) to represent vector, and bold uppercase character (*i.e.*, $\mathbf{A}$) to represent matrix. The other frequently used notations and abbreviations are listed as follows,

List of Notations:

$\mathbf{x}_i$	Feature vector of the i-th sample
y_i	Label of the i-th sample
$\mathbf{y}$	Label vector of the given training data
n	Number of training samples
$'$	Transpose of a vector or matrix
$\mathcal{B}_l$	the l-th bag
$\mathbf{0}_n$	$n \times 1$ vector of all zeros
$\mathbf{1}_n$	$n \times 1$ vector of all ones
$\boldsymbol{\alpha} \circ \boldsymbol{\beta}$	Element-wise product of two vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
$\mathbf{A} \circ \mathbf{B}$	Element-wise product of two matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{I}_n$	$n \times n$ identity matrix
$\mathbf{K}$	Kernel matrix
$\mathbf{K}_m$	the m-th Kernel matrix
$\phi(\cdot)$	Nonlinear feature mapping function
$k(\cdot, \cdot)$	Kernel function induced by $\phi(\cdot)$

List of Abbreviations:

MIL	Multi-instance Learning
PI	Privileged Information
LUPI	Learning using Privileged Information
DA	Domain Adaptation
FV	Fisher Vector
GMM	Gaussian Mixture Model
MMD	Maximum Mean Discrepancy
SVM	Support Vector Machine
ESVM	Exemplar Support Vector Machine
SVM+	Support Vector Machine using Privileged Information

MKL	Multiple Kernel Learning
QP	Quadratic Programming
SMO	Sequential Minimal Optimization

Chapter 1

Introduction

1.1 Background

The research interest in developing visual recognition algorithms grows rapidly for a variety of real-world applications including object recognition, human action recognition, and event recognition. The conventional method is to learn a classifier based on the training samples and then apply the learnt classifier on the test samples. In real-world applications, the training data and test data may come from different domains, for example, the training and test data are sampled from different datasets or captured in different environments. When the training data come from one domain (*i.e.*, source domain) and the test data come from another domain (*i.e.*, target domain), the data distributions between the training data and test data are considerably different, which is generally referred to as domain distribution mismatch. Due to the domain distribution mismatch, the performance of the classifier learnt from the training set will be significantly degraded on the test set. Therefore, how to address the domain distribution mismatch between the source domain and target domain becomes a very important research topic in the field of computer vision and has attracted a large number of researchers. The solutions to address the domain distribution mismatch can be classified into domain adaptation and domain generalization. The key difference between domain adaptation and domain generalization lies in whether the unlabeled target domain is available in the training stage. In this thesis, we study both domain adaptation and domain generalization for visual recognition.

For domain adaptation, the unlabeled target domain is available during the training procedure. Thus, the unlabeled target domain data can be utilized in the train-

ing phase to reduce the domain distribution mismatch between the source domain and the target domain. The existing domain adaptation methods can be classified into feature based methods [3, 7, 19, 53, 60, 62, 86, 119], SVM based methods [16, 39, 41], instance-reweighting methods [71], dictionary learning methods [136], and low-rank based methods [76, 133]. Specifically, feature based methods tend to interpolate the subspaces of two domains [19, 60, 62], project two subspaces into a common subspace [7, 119], learn a common metric [86], or align the source subspace to the target one [3, 53]. SVM based approaches either used Multiple Kernel Learning (MKL) [39, 41] or progressively added informative target domain samples into the training set [16]. The instance-reweighting approach in [71] tended to assign larger weights to the source domain samples which are closer to the target domain. Dictionary learning method [136] was proposed to learn a common discriminative dictionary for both source and target domain samples. Low-rank based methods [76, 133] assumed the source domain samples can be reconstructed by using target domain samples or vice versa and the reconstruction matrix should be low-rank. Interested readers can refer to the surveys [121, 122] on domain adaptation for more details.

For domain generalization, the target domain is unseen in the training phase, and thus the unlabeled target domain data cannot be utilized to reduce the domain distribution mismatch between the source domain and the target domain. Instead, the domain generalization method aims to learn a classifier from the source domain which can be generalized well to any arbitrary target domain, by using existing multiple source domains or discovering multiple latent domains from one source domain. The existing domain generalization techniques can be roughly categorized into learning from multiple source domains [80, 110] and discovering latent domains explicitly or implicitly [59, 69, 164]. In particular, for learning from multiple source domains, domain invariant feature representations are learnt in [110] by reducing marginal distribution mismatch between different source domains while maintaining the conditional distribution on each source domain. In [80], an SVM based approach was proposed to undo the dataset bias problem by learning one classifier for each source domain. For discovering latent domains explicitly [59, 69], the training samples on the source domain are clustered into different hidden latent domains in [69] while the sum of distribution mismatch between each pair of different

latent domains is maximized in [59]. After the latent domains are discovered, the classifiers learnt based on each hidden latent domain can be fused to predict the test data. Besiding discovering latent domains explicitly, Xu *et al.* [164] aims to exploit the latent domain structure implicitly based on exemplar classifiers, in which the positive training sample are assumed to come from multiple latent domains and the exemplar classifiers corresponding to the positive training samples from the same latent domain should be similar with each other. In the testing stage, for each test sample, the scores of the exemplar classifiers which obtain the top highest scores on this test sample are fused as the final score.

1.2 Motivation and Contributions

In terms of the existing domain adaptation and domain generalization approaches proposed for visual recognition, most of them require well labeled source domain data. However, collecting labeled training data is often time consuming and expensive. As more and more people upload images or videos to public websites (*e.g.*, Flickr, youtube) in recent years, there is a rising research interest on how to learn a classifier based on web images/videos for visual recognition such as [22, 41] because a large amount of web data are freely available. Therefore, based on the type of used training data, the existing domain adaptation and domain generalization methods can also be categorized into web data based methods and well labeled data based methods.

For learning from web data, besides the data distribution mismatch between the web training data and test data, there also exist some other problems such as the label noise of web data and extra information associated with the web data. Specifically, since the training web data for each category are crawled from the searching engine of public website by using the category name as the query, the labels of crawled training web data are usually very noisy and inaccurate, which will degrade the performance of the learnt classifier on the test data. Moreover, the training web data are generally associated with rich textual information (*e.g.*, tags, captions, and short descriptions), which are not available for the test data. In this case, the additional textual information associated with the web data is referred to as privileged information [98, 150]. Although

the privileged information is not available for the test data, it can still be used to help train a more robust classifier in the training stage. To the best of our knowledge, all the existing domain adaptation and domain generalization methods only consider the domain distribution mismatch, but ignore the label noise and privileged information of web data, which is not suitable for the application of learning from web data. To this end, we propose a domain adaptation framework and a domain generalization method for learning from web data, which leads to the first and second work in this thesis respectively.

In the first work, we propose our domain adaptation framework named Domain Adaptive Multi-Instance Learning using Privileged Information (MIL-PI-DA) for visual recognition by learning from web data, which can handle the label noise of web data, utilize the privileged information, and simultaneously reduce the domain distribution mismatch by using unlabeled target domain data in the training stage. As far as we know, this is the first work to tackle with the three major issues of learning from web data at the same time.

In the second work, we propose our domain generalization method named weakly supervised domain generalization (WSDG) for visual recognition by learning from web data, which can cope with the label noise, take advantage of the privileged information, and generalize well to any unseen target domain. In order to enhance the generalization ability of the learnt classifier, we assume the source domain contains multiple latent domains and learn one classifier for each latent domain so that the integrated classifier can be generalized to arbitrary target domain. To the best of our knowledge, this is the first work to explore the domain generalization problem under the weakly supervised learning setting.

For learning from well labeled data, there are also some issues with the existing domain adaptation and generalization approaches. One issue is that in many applications, both the training data and test data are associated with multiple types of features, in which exploiting the relation among multiple types of features can further improve the performance instead of simple early fusion (*i.e.*, concatenating multiple types of features) or late fusion (*i.e.*, averaging the decision values on multiple views). Although multi-view (*i.e.*, multiple types of features) domain adaptation has been studied in [11, 39, 166, 174], no multi-view approach has been proposed for domain generalization, which motivates our third work in this thesis.

In the third work, we propose a framework named Exemplar-based Multi-View Domain Generalization (EMVDG) for visual recognition based on the exemplar SVM classifiers, following both the consensus principle and complementary principle. Specifically, on one hand, we propose an EMVDG_CO method by enforcing the cluster structures of exemplar SVM classifiers on different views to be consistent based on the consensus principle. On the other hand, we also propose another EMVDG_MK method by fusing the exemplar SVM classifiers from different views based on the complementary principle. To the best of our knowledge, this is the first work to explore the domain generalization problem in the multi-view scenario.

Another issue when learning from well labeled data is how to design domain adaptation or generalization methods for global feature representations constructed based on local descriptors of images/videos, in which a set of local descriptors are extracted from each image/video and encoded into a high dimensional vector by using different encoding methods. Among the encoding methods, Fisher vector [74] encoded based on Gaussian Mixture Model (GMM) achieves satisfactory performances in many visual recognition tasks. However, the GMM learnt on the source domain does not take the data distribution of target domain into consideration and thus cannot generalize well to the target domain, which has not been specified by previous domain adaptation approaches. Considering that the GMM learnt on the source domain may not capture the data distribution of target domain very well, we propose a domain adaptation method which is specifically designed for Fisher vector, resulting in the fourth work in this thesis.

In the fourth work, we propose a Domain Adaptation method based on Fisher Vector (DAFV) for visual recognition, which is the first work to select domain invariant components of Fisher vectors with each component of Fisher vector corresponding to one Gaussian model in the GMM. Specifically, we assign higher weights on the domain invariant components of Fisher vectors, and thus the transformed Fisher vectors become more invariant across the source domain and the target domain.

For all our proposed domain adaptation and domain generalization methods, comprehensive experiments are conducted on benchmark datasets and our methods are compared with the state-of-the-art baselines. The experimental results indicate the effectiveness of our proposed methods for visual recognition under different settings.

1.3 Thesis Structure

This thesis contains seven chapters and the rest of the thesis is organized as follows. The structure of this thesis is shown in Figure 1.1, from which we can observe that Chapter 3 and Chapter 4 focus on learning from web data while Chapter 5 and Chapter 6 focus on learning from well labeled data. We can also observe that Chapter 3 and Chapter 6 focus on domain adaptation while Chapter 4 and Chapter 5 focus on domain generalization. In Chapter 2, we investigate previous works related to multi-instance learning, learning using privileged information, domain adaptation, and domain generalization. In Chapter 3, we first propose a new MIL-PI framework together with three instantiations sMIL-PI, mi-SVM-PI and MIL-CPB-PI, in which we take advantage of the additional textual information associated with the training web data and effectively cope with the label noise as well, and then propose a new MIL-PI-DA framework and three instantiations sMIL-PI-DA, mi-SVM-PI-DA and MIL-CPB-PI-DA for domain adaptation. In Chapter 4, we first propose a novel weakly supervised domain generalization approach WSDG, which is able to handle the label noise in training web data and has good generalization ability to any unseen target domain, and then extended our WSDG approach to WSDG-PI by utilizing additional textual features as privileged information. In Chapter 5, we propose an exemplar-based multi-view domain generalization (EMVDG) framework, which can enhance the domain generalization capability to any unseen target domain and simultaneously exploit the relation among multiple types of features. We also extend our EMVDG framework to EMVDA framework for domain adaptation. In Chapter 6, we propose a Domain Adaptation method based on Fisher Vector (DAFV), which is specifically designed for Fisher vectors by selecting the domain invariant components of Fisher vectors corresponding to the common Gaussian model. The effectiveness of all our proposed methods for visual recognition has been demonstrated by comprehensive experiments. In Chapter 7, we conclude our work and propose future research directions.

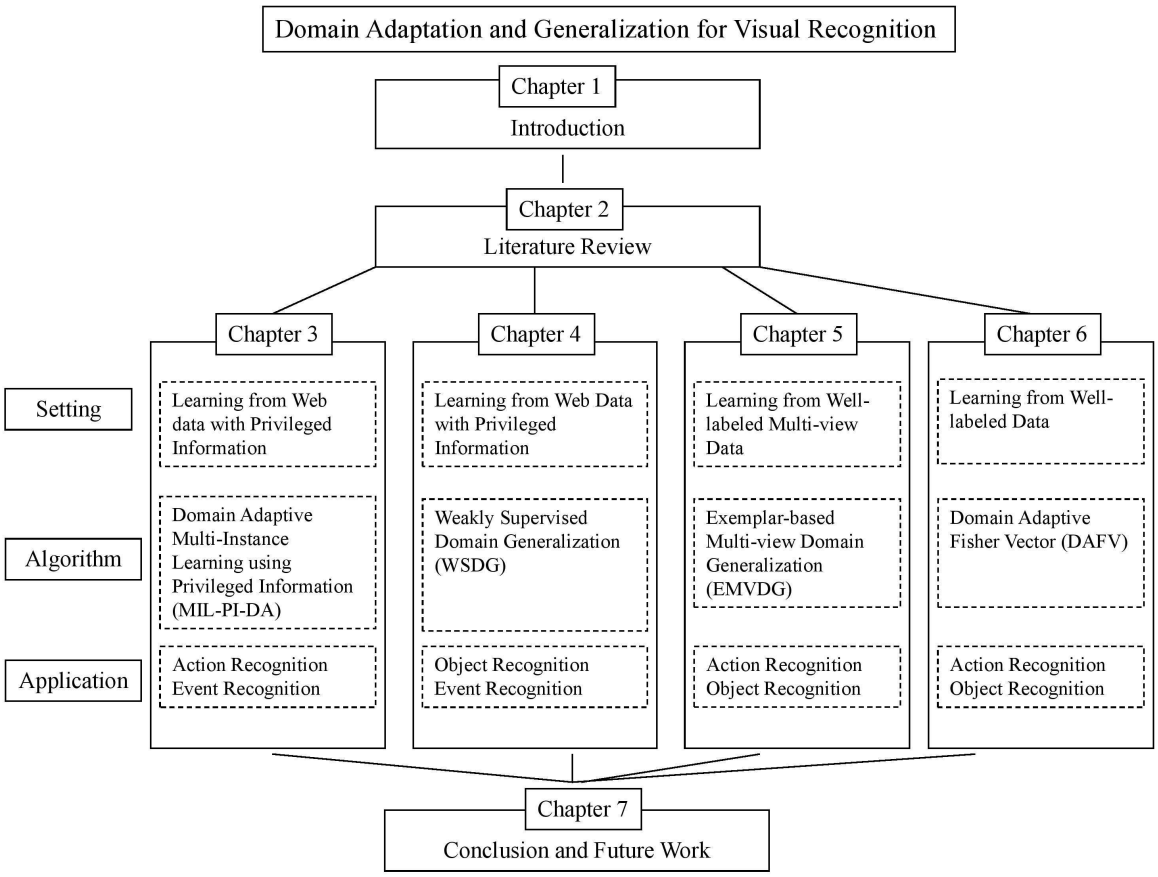


Figure 1.1: The structure of this thesis.

Chapter 2

Literature Review

In this chapter, we conduct the literature survey in four aspects related to our proposed methods, namely, multi-instance learning that deals with label noise, learning using privileged information that exploits privileged information, domain adaptation that can reduce domain distribution mismatch, and domain generalization that can learn classifiers generalizable to arbitrary target domain. For ease of presentation, throughout the rest of this thesis, we use a lowercase/uppercase letter in boldface to denote a vector/matrix (*e.g.*, $\mathbf{a}$ denotes a vector and $\mathbf{A}$ denotes a matrix). The superscript $'$ denotes the transpose of a vector or a matrix. We denote $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ as the n-dim column vectors of all zeros and all ones, respectively. For simplicity, we also use $\mathbf{0}$ and $\mathbf{1}$ instead of $\mathbf{0}_n$ and $\mathbf{1}_n$ when the dimension is obvious. Similarly, we use $\mathbf{I}$ and $\mathbf{O}$ to denote the identity matrix and the matrix of all zeros, respectively. We use $\mathbf{A}^{-1}$ to denote the inverse matrix of $\mathbf{A}$ and $\mathbf{A} \circ \mathbf{B}$ to denote the element-wise product between two matrices $\mathbf{A}$ and $\mathbf{B}$. The inequality $\mathbf{a} \leq \mathbf{b}$ means that $a_i \leq b_i$ for $i = 1, \ldots, n$. Moreover, we denote the indicator function as $\delta(a = b)$, in which $\delta(a = b) = 0$ if $a \neq b$, and $\delta(a = b) = 1$, otherwise.

2.1 Multi-instance Learning

In traditional supervised learning paradigm, the positive and negative samples are unambiguously labeled. However, it takes expensive and time-consuming labor to collect data. Moreover, the labels of collected data may not be reliable. For instance, many tag based retrieved web images are in fact irrelevant. As is well known, few training samples will damage the performance of trained classiers. Since unambiguously labeled samples can

not be easily obtained, we can utilize the ambiguously labeled samples to assist in training a more robust classier, which is named weakly supervised learning or ambiguous learning.

Multi-instance learning (MIL) is a weakly supervised learning paradigm which at rst partitions the training images into clusters and then treats each cluster as a "bag" and the images in each bag as "instances". MIL was originally applied to solve the drug prediction problem in biochemistry [33]. In this problem, a good drug molecule (positive bag) will bind very tightly to the target binding site, while a poor drug (negative bag) molecule will not. The variant instances (instances in bag) are alternative conformations of the molecule-alternative shapes that the molecule can adopt by rotating its bonds. One or a few of the instances from positive bags function in binding to the target binding site and generate the positive observed result. Thus the traditional assumption in MIL problem is that each positive bag contains at least one positive instance and each negative bag only contains negative instances. Formally, let us represent the training data in MIL as $\{(\mathcal{B}_l,\, Y_l)_{l=1}^{L}\}$, $\mathcal{B}_l$ is a training bag, $Y_l \in \{+1, 1\}$ is the corresponding bag label, and L is the total number of training bags. Each training bag $\mathcal{B}_l$ consists of a number of training instances, *i.e.*, $\mathcal{B}_l = \{(\mathbf{x}_i, \tilde{\mathbf{x}}_i, y_i)|i \in \mathcal{I}_l\}$, where $\mathcal{I}_l$ is the set of indices for the instances inside $\mathcal{B}_l$, $\mathbf{x}_i$ is the visual feature of the i-th sample and $y_i \in \{+1, 1\}$ is the ground truth label of the instance which is unknown. Without loss of generality, we assume the positive bags are the rst L^+ training bags with a total number of n^+ positive training instances, and the total number of training instances is denoted as n. According to the traditional MIL assumption, we have

$$\begin{cases} \sum_{i\in\mathcal{I}_l} \frac{y_i+1}{2} \geq 1, & \forall Y_l = 1, \\ y_i = -1, & \forall i \in \mathcal{I}_l \text{ and } Y_l = -1, \end{cases} \tag{2.1}$$

In recent years, MIL has been applied to other areas such as computer vision, *e.g.*, image retrieval and categorization [96, 177]. In [96], the MIL assumption is different from traditional assumption, which claims that each positive bag contains at least a portion of positive instances instead of one positive instance. The generalized MIL assumption can be formulated as follows,

$$\begin{cases} \sum_{i\in\mathcal{I}_l} \frac{y_i+1}{2} \geq \sigma|\mathcal{B}_l|, & \forall Y_l = 1, \\ y_i = -1, & \forall i \in \mathcal{I}_l \text{ and } Y_l = -1, \end{cases} \tag{2.2}$$

The earliest algorithms for MIL problem was proposed in [5, 33, 105]. These methods solve MIL problem by using axis-aligned rectangle. Then a kernel based approach was suggested in [28], which derives bag-level MI-kernels from instance-level dened kernels. More recently, Stuart Andrews proposed two MIL methods in [56], *i.e.*, mi-SVM and MI-SVM, which are conceptual modications based on SVM. The rst method mi-SVM alternatively infers hidden labels and maximizes soft margins given inferred labels. The second method MI-SVM generalizes instance-level margin to bag-level margin and aims to maximize bag-level margin directly. MI-SVM actually selects the most positive sample from each positive bag. When each bag contains only one instance, which is named Single-instance Learning (SIL), these two methods can be reduced to standard soft-margin SVM. Similar with mi-SVM, MIL-CPB [96] also explicitly infers sample labels. However, mi-SVM can only reach local optimum while MIL-CPB aims at global optimum by using multiple kernel learning.

In contrast with the instance-level methods mentioned above, a bag-level method sparse MIL [17] was proposed in favor of the situation that positive bags can be very sparse with a small fraction of positive instances. This method treats each positive bag as a positive sample and adjusts the margin based on the positive ratio of positive bags. Another bag-level method MILES [24] maps each bag into feature space via instance similarity within each bag and trains more robust classiers by using selected more important features.

In summary, multi-instance learning methods can be generally classied into bag-level methods and instance-level methods. The bag-level MIL methods [17, 24] focus on the classication of the bags. Note the labels of training bags are known. By transforming each training bag to one training sample, the MIL problem becomes a supervised learning problem. Different from the bag-level MIL methods, the instance-level MIL methods [4, 96] directly solve the classication problem for the instances. However, the labels of training instances are unknown, so one needs to infer the instance labels when learning the MIL classier.

2.2 Learning using Privileged Information

Privileged information [151] is the information that is only available during training process but not available during test process. One metaphor used in [151] is that students learn privileged knowledge from teachers in class but cannot gain knowledge from teachers during examination. The new learning paradigm learning using privileged information is exploiting the information which is only available during training process to help train a more robust classifier via modeling loss by using correcting function *w.r.t.* privileged information. Different types of privileged information are used in different areas, such as bioinformatics, nance market prediction and digit recognition.

Let us denote the training data as $\{(\mathbf{x}_i, \tilde{\mathbf{x}}_i, y_i)|_{i=1}^n\}$, where $\mathbf{x}_i$ is feature representation for the i-th training sample, $\tilde{\mathbf{x}}_i$ is the corresponding feature representation of privileged information which is not available for test data, $y_i \in \{+1, 1\}$ is the label, and n is the number of training samples. The goal of SVM+ [151] is to learn the classier $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x})+b$, where $\phi(\cdot)$ is a nonlinear feature mapping function. The objective of SVM+ is as follows, **SVM+:** SVM+ builds up the traditional SVM by further exploiting privileged information in training data. The objective of SVM+ is as follows,

$$\begin{aligned} \min_{\tilde{\mathbf{w}},\tilde{b},\mathbf{w},b} \quad & \frac{1}{2}\left(\|\mathbf{w}\|^2 + \gamma\|\tilde{\mathbf{w}}\|^2\right) + C\sum_{i=1}^{n}\xi(\tilde{\mathbf{x}}_i), \\ \text{s.t.} \quad & y_i(\mathbf{w}'\phi(\mathbf{x}_i) + b) \geq 1 - \xi(\tilde{\mathbf{x}}_i), \quad \xi(\tilde{\mathbf{x}}_i) \geq 0, \quad \forall i, \end{aligned} \tag{2.3}$$

where γ and C are tradeoff parameters, $\xi(\tilde{\mathbf{x}}_i) = \tilde{\mathbf{w}}'\tilde{\phi}(\tilde{\mathbf{x}}_i) + \tilde{\mathbf{b}}$ is the slack function, which replaces the slack variable ξ_i in the hinge loss in SVM. Such a slack function plays a role of the teacher in the training process. Recall the slack variable ξ_i in SVM tells about how difficult to classify the training sample $\mathbf{x}_i$. The slack function $\xi(\tilde{\mathbf{x}}_i)$ is expected to model the optimal slack variable ξ_i by using privileged information analogous to the comments and explanations from the teacher in human learning [151]. Similar with SVM, SVM+ can be solved in the dual form by optimizing a quadratic programming problem.

Several extensions of SVM+ are mentioned in [151] and one of them is partial SVM+ (pSVM+) where privileged information is only available for a part of training data. Assume the training data with privileged information is $\{(\mathbf{x}_i, \tilde{\mathbf{x}}_i, y_i)|_{i=1}^n\}$ and the training data without privileged information is $\{(\mathbf{x}_i, \tilde{\mathbf{x}}_i, y_i)|_{i=n+1}^l\}$. In this situation, we can use

correcting function for the training data with privileged information and normal slack variable which is the same as soft-margin SVM for the training data without privileged information. The primal form is written as follows,

$$\begin{aligned}
\min_{\tilde{\mathbf{w}},\tilde{b},\mathbf{w},b,\boldsymbol{\eta}} \quad & \frac{1}{2}\left(\|\mathbf{w}\|^2+\gamma\|\tilde{\mathbf{w}}\|^2\right)+C_1\sum_{i=1}^{l}\xi(\tilde{\mathbf{x}}_i)+\sum_{i=l+1}^{n}\eta_i, \\
\text{s.t.} \quad & y_i(\mathbf{w}'\phi(\mathbf{x}_i)+b)\geq 1-\xi(\tilde{\mathbf{x}}_i), \quad \forall i=1,\ldots,l, \\
& \xi(\tilde{\mathbf{x}}_i)\geq 0, \quad \forall i=1,\ldots,l, \\
& y_i(\mathbf{w}'\phi(\mathbf{x}_i)+b)\geq 1-\eta_i, \quad \forall i=l+1,\ldots,n, \\
& \eta_i\geq 0, \quad \forall i=l+1,\ldots,n,
\end{aligned} \tag{2.4}$$

Similar with SVM+, we can obtain the dual form which can also be solved by optimizing a quadratic programming problem.

2.3 Domain Adaptation

Supervised learning is widely employed in computer vision and achieves good performance in many tasks, but the performance of supervise learning is limited by the amount of labeled training data. However, collecting labeled data is expensive and time-consuming. When the labeled training data is limited in one domain, another domain with sufficient labeled training data is expected to help improve the performance. However, training on one domain and testing on another domain may harm the performance due to the domain distribution mismatch between source domain and target domain. This problem is generally referred to as covariate shift [138] or dataset bias [80, 144]. Some special issues in domain adaptation have been studied under different names, such as covariate shift [138], class imbalance [75], and sample selection bias [66, 171].

Some machine learning problems are related to but different from domain adaptation, such as transfer learning [121], self-taught learning [129], semi-supervised learning [181] and multi-view learning [134]. The definition of transfer learning has changed through different periods. The generalized transfer learning contain domain adaptation, multi-task learning, cross-category analysis and knowledge transfer. Multi-task learning consider multiple tasks where source domain distribution and target domain distribution are the

same. In self-taught learning, the labeled data is limited and a large amount of mildly related unlabeled data is utilized. Semi-supervised learning is closely related to domain adaptation. The unlabeled data which has the same distribution as labeled data is exploited to remedy the lack of labeled data. In domain adaptation situation, assuming source domain and target domain have no domain difference, if we treat the labeled source domain data as labeled data and the unlabeled target domain data as unlabeled data, then the resulting problem is identical with the semi-supervised learning problem. In many computer vision applications, multiple views of data are commonly used. Multi-view learning (also known as cross-view or multi-modal) aims at nding the correspondence among multiple views, which is essentially different from domain adaptation.

In domain adaptation problem, there are labeled source domain data as well as labeled or unlabeled target domain data. Let us denote X_s (*resp.*, Y_s) as training data (resp., label) and X_t(resp., Y_t) as test data (*resp.*, label). We denote $P_s(X_s, Y_s)$ (*resp.*, $P_t(X_t, Y_t)$) as joint probability of source domain (target domain), which are usually unknown. We denote $P_s(X_s)$ (*resp.*, $P_t(X_t)$) as marginal probability of source domain (target domain). We also denote $P_s(Y_s|X_s)$ (*resp.*, $P_t(Y_t|X_t)$) as conditional probability of source domain (*resp.*, target domain).

- Covariate Shift: $P_t(Y_t|X_t) = P_s(Y_s|X_s)$ but $P_t(X_t) = P_s(X_s)$. This kind of domain dierence is known as covariate shift [138] or sample selection bias [66, 171].

- Class Imbalance: $P_t(X_t|Y_t) = P_s(X_s|Y_s)$ but $P_t(Y_t) = P_s(Y_s)$, This kind of domain dierence is known as class imbalance [75].

Generally speaking, domain adaptation approaches can be roughly categorized into instance reweighting methods, feature-level methods, classifier-level methods, dictionary learning methods, and deep learning methods.

For instance reweighting methods, regarding two specic cases in domain adaptation mentioned above, training samples can be assigned different weights to make the classier benecial for predicting target domain samples. For covariate shift, we can reweight each training sample by using the ratio $\frac{P_t(X_s)}{P_s(X_s)}$. Different methods such as non-parametric method [138, 142] and kernel based method [71] can be employed to estimate the ratio

$\frac{P_t(X_s)}{P_s(X_s)}$. For class imbalance, we can reweight each training sample by using $\frac{P_t(Y_s)}{P_s(Y_s)}$, similarly as the solution to covariate shift.

For feature-level methods, feature augmentation (also known as feature replication) [31] is one of the simplest feature-level methods. This method transforms N-dimensional features to $3N$-dimensional augmented features. The first N-dimension component represents the commonality between source domain and target domain. The second N-dimensional component is specifically for source domain while the last N-dimensional component is specically for target domain. The main idea of feature augmentation is to treat labeled target domains samples more importantly than labeled source domain samples. Then the idea of feature augmentation is extended to manifold based approaches which consider intermediate domains as in SGF [62]. By treating source domain subspace and target domain subspace as two points on Grassmann manifold, the sampled intermediate points correspond to meaningful intermediate subspaces. Based on the work of SGF, Gong [60] proposes to interpolate innite number of intermediate points which correspond to continuous intermediate subspaces. Some recent approaches aim to learn a domain invariant subspace [7] or align the subspaces from source domain and target domain [53]. Subspace based methods can be unied into one formulation as follows,

$$\min_{\mathbf{P},\mathbf{U}} \|\mathbf{PS} - \mathbf{UT}\|, \tag{2.5}$$

where $\mathbf{S}$ (*resp.*, $\mathbf{T}$) is the subspace of source domain (*resp.*, target domain) and $\mathbf{P}$ (*resp.*, $\mathbf{U}$) is the transform matrix of source domain (*resp.*, target domain).

For the classifier-level methods, SVM can be directly used by combining source domain samples with labeled target domain samples, which can generally produce better results than only using labeled target domain samples. Some classier level domain adaptation methods are modied based on SVM. One of the earliest SVM based domain adaptation approaches A-SVM [165] utilizes the decision values of auxiliary classiers and adds perturbation function which is learned by using labeled target domain data. Then L. Duan proposes domain transfer SVM (DTSVM) [43] which aims to reduce domain distribution mismatch and minimize the structural risk function at the same time. DTSVM uses Multiple Kernel Learning (MKL) to learn the weight of various types of kernels. Inspired by A-SVM, DTSVM is further combined with the decision values of auxiliary

classiers, which leads to a new domain adaptation approach A-MKL [41]. SVM based domain adaptation approaches can also be applied to multiple source domain such as Domain Adaptation Machine (DAM) [43] and Domain Selection Machine (DSM) [45]. DAM learns the weight of source domains by making the target classier decision values close to the weighted combination of decision values of multiple source domain classiers on unlabeled target domain data. Its worth mentioning that DAM is essentially support vector regression. Based on DAM, DSM further adds a binary domain selector to choose related source domains in order to avoid negative transfer. Another SVM based approach DASVM [16] is different from all the approaches mentioned above. This approach progressively label target domain samples and remove labeled source domain samples such that the learnt classier is gradually getting benecial for predicting on target domain.

For dictionary learning methods, dictionary learning has been widely used in various computer vision applications since it provides robust discriminative representation. However, the representation learnt on one domain may not adapt well to target domain if source domain and target domain have distinctive domain distributions. Several dictionary learning based approaches have been proposed to handle the domain issue, such as [128, 136]. The approach in [136] project samples in both domains into low-dimensional common subspace and learn dictionary based on projected common subspace.

For deep learning methods, deep learning has achieved tremendous improvement (state-of-arts results) in many computer vision and machine learning tasks. Deep learning can extract deep features which are more domain invariant than shallow features. However, there still exist domain distribution difference between source domain deep features and target domain deep features. Thus, domain adaptation approach based on deep learning can further improve the performance, such as [27, 58]. Some works [8] use fine-tuning to adapt the pretrained neural network to a new dataset, but fine-tuning requires the labels for the new dataset, which are not available for unsupervised domain adaptation.

2.4 Domain Generalization

When target domain data is unseen in the training stage, we aim to learn the source classifier which can be generalized to any unseen target domain, which leads to the

domain generalization problem. In [110], domain invariant feature representations are learnt by reducing marginal distribution mismatch between different latent domains while maintaining the conditional distribution on each view. In [80], an SVM based approach was proposed to undo the dataset bias problem by learning one classifier for each domain. However, in the above works, latent domain labels in the source domain are required, which are generally unavailable in the real-world applications. In order to exploit latent domain structures, Xu *et al.* [164] proposed an approach based on exemplar classifiers, in which the positive training sample are assumed to come from multiple latent domains and the exemplar classifiers corresponding to the positive training samples from the same latent domain should be similar with each other. In the testing stage, for each test sample, we fuse the scores of the exemplar classifiers which obtain the top highest scores on this test sample. In this way, we assume this test sample is more likely to be sampled from the latent domain consisting of the selected positive training samples. Therefore, the integrated classifier based on learnt exemplar classifiers can be generalized to arbitrary target domain.

Domain generalization is closely related to the latent domain discovering methods [59, 69]. In [69], the training samples on the source domain are clustered into different hidden latent domains. In [59], the sum of distribution mismatch between each pair of different latent domains is maximized. After the latent domains are discovered, the classifiers learnt based on each hidden latent domain can be fused to predict the test data. Two fusing strategies are employed when fusing the learnt classifiers, which are referred to as the "ensemble" strategy and "match" strategy, respectively. The "ensemble" strategy is to reweight the decision values from different SVM classifiers by using the prelearnt domain probabilities while the "match" strategy is to select the most relevant domain based on the MMD criterion followed by using the corresponding SVM classier to predict the test samples.

2.5 Visual Recognition

In this thesis, our applications focus on visual recognition including object recognition, action recognition, and event recognition.

For object recognition, which plays an important role in the field of computer vision, myriad of approaches have been developed to improve the recognition accuracy. Based on whether the training samples are used or not, object recognition methods can be categorized into supervised learning methods and unsupervised learning methods. Based on whether the parameters related to the data distribution are used or not, object recognition methods can be classified into parametric methods and non-parametric methods. Based on which kind of pixel information is used, object recognition methods can be roughly categorized into per-pixel methods and sub-pixel methods. Based on whether spatial information is used or not, object recognition methods can be coarsely classified into spectral methods, contextual methods, and spectral-contextual methods. The commonly used datasets for object recognition comprise Bing-Caltech [9], Office-Caltech [60], Pascal VOC [46], Imagenet [32], and so on.

For action and event recognition, the research interest has been rising rapidly due to the wide range of applications of action/event technologies including video search and retrieval, intelligent video surveillance, and human computer interaction. Note that the two terms actions and events are often interchangeably used in many previous works [2, 13], high-level events generally consist of a sequence of interactions or stand-alone actions [78]. Recognizing actions/events from videos is still a challenging task due to considerable camera motion, cluttered backgrounds, and large intra-class variations. In recent years, abundant approaches have been proposed for action recognition [70, 90, 102, 109, 137, 147, 154, 157, 169, 172, 180] and event recognition[21, 89, 113, 132, 143, 162]. Interested readers can refer to the recent surveys [2] and [78] for more details. The popular benchmark datasets for action/event recognition contains Kodak [106], CCV [79], HMDB51 [85], ACT42 [26], Online RGBD Action Dataset (ORGBD) [168], etc.

Chapter 3

Exploiting Privileged Information from Web Data for Visual Recognition

In the conventional approaches for action and event recognition, sufficient labelled training videos are generally required to learn robust classifiers with good generalization capability on new test videos. However, collecting labelled training videos is often time consuming and expensive. In this chapter, we propose new learning frameworks to train robust classifiers for action and event recognition by using freely available web videos as training data. We aim to address three challenging issues: 1) the training web videos are generally associated with rich textual descriptions, which are not available in test videos; 2) the labels of training web videos are noisy and may be inaccurate; 3) the data distributions between training and test videos are often considerably different. To address the first two issues, we propose a new framework called multi-instance learning with privileged information (MIL-PI) together with three new MIL methods, in which we not only take advantage of the additional textual descriptions of training web videos as privileged information, but also explicitly cope with noise in the loose labels of training web videos. When the training and test videos come from different data distributions, we further extend our MIL-PI as a new framework called domain adaptive MIL-PI (MIL-PI-DA). We also propose another three new domain adaptation methods, which can additionally reduce the data distribution mismatch between training and test videos. Comprehensive experiments for action and event recognition demonstrate the effectiveness of our proposed approaches.

3.1 Introduction

There is an increasing research interest in developing new action and event recognition technologies for a broad range of real-world applications including video search and retrieval, intelligent video surveillance and human computer interaction. While the two terms, actions and events, are often interchangeably used in several existing works [2, 13], high-level events generally consist of a sequence of interactions or stand-alone actions [78].

It is still a challenging computer vision task to recognize actions/events from videos due to considerable camera motion, cluttered backgrounds and large intra-class variations. Recently, a large number of approaches have been proposed for action recognition [70, 90, 102, 109, 137, 147, 154, 157, 169, 172, 180] and event recognition[21, 89, 113, 132, 143, 162]. Interested readers can refer to the recent surveys [2] and [78] for more details. However, all the above methods follow the conventional approaches, in which a set of action/event lexicons are first defined and then a large corpus of training videos are collected with the action/event labels assigned by human annotators.

Collecting labelled training videos are often time-consuming and expensive. Meanwhile, rich and massive social media data are being posted to the video sharing websites like *Flickr* and *Youtube* everyday, in which web videos are generally associated with valuable contextual information (*e.g.*, tags, captions, and surrounding texts). Consequently, several recent works [22, 41] proposed to perform keywords (also called tags) based search to collect a set of relevant and irrelevant web videos, which are directly used as positive and negative training data for learning robust classifiers for action/event recognition. However, those works cannot effectively utilize the textual descriptions of training web videos because the test videos (e.g., the videos in the HMDB51 dataset) do not contain such textual descriptions.

In this chapter, we propose new learning frameworks for action and event recognition by using freely available web videos as training data. Specifically, as shown in Fig 3.1, we aim to address three challenging issues 1) the training web videos are usually accompanied with rich textual descriptions, while such textual descriptions are not available in the test videos; 2) the labels of training web videos are noisy (i.e., some labels are inaccurate); 3) the feature distributions of training and test videos may have very different statistical properties such as mean, intra-class variance and inter-class variance [39, 41].

To utilize the additional textual descriptions from the training web videos, we extract both visual features and textual features from the training videos. While we do not have textual features in the test videos, such textual features extracted from the training videos can still be used as privileged information, as shown in the recent work [150]. Their work is motivated by human learning, where a teacher provides the students with hidden information through explanations, comments, comparisons *etc.* [150]. Similarly, we observe that the surrounding textual descriptions more or less describe the content of training data. So the textual features can additionally provide hidden information for learning robust classifiers by bridging the semantic gap between the low-level visual features and the high-level semantic concepts.

To cope with noisy labels of relevant training samples, we further employ the multi-instance learning (MIL) technologies because the MIL methods can still be used to learn classifiers even when the label of each training instance is unknown. Inspired by the recent works [93, 96, 152], we first partition the training web videos into small subsets. By treating each subset as a "bag" and the videos in each bag as "instances", the MIL methods such as Sparse MIL (sMIL) [17], mi-SVM [4] and MIL-CPB [96] can be readily adopted to learn robust classifiers by using loosely labeled web videos as training data.

To address the first two challenging issues for action/event recognition, we propose our first framework called multi-instance learning with privileged information (MIL-PI). In this framework, we not only take advantage of the additional textual features from training web videos as privileged information, but also explicitly cope with noise in the loose labels of relevant training web videos. We also develop three new MIL approaches called sMIL-PI, mi-SVM-PI, and MIL-CPB-PI based on three existing MIL methods sMIL, mi-SVM and MIL-CPB, respectively. Moreover, we also observe that the action/event recognition performance could degrade when the training and test videos come from different data distributions, which is known as the *dataset bias* problem [144]. To explicitly reduce the data distribution mismatch between the training and test videos, we further extend our MIL-PI framework by additionally introducing a Maximum Mean Discrepancy (MMD) based regularizer, which leads to our new MIL-PI-DA framework. We further extend sMIL-PI, mi-SVM-PI, and MIL-CPB-PI as sMIL-PI-DA, mi-SVM-PI-DA and MIL-CPB-PI-DA, respectively.

(a) (b) (c)

Figure 3.1: Three challenging issues when learning from loosely labeled web videos: (a) the training web videos are additionally associated with rich textual descriptions, (b) the labels of relevant training web videos retrieved using the textual query "sports" are noisy, and (c) there is domain distribution mismatch between the training web videos and test consumer videos.

We conduct comprehensive experiments to evaluate our new approaches for action and event recognition. The results show that our newly proposed methods sMIL-PI, mi-SVM-PI and MIL-CPB-PI not only improve the existing MIL methods (*i.e.*, sMIL, mi-SVM and MIL-CPB), but also outperform the learning methods using privileged information as well as other related baselines. Moreover, our newly proposed domain adaptation methods sMIL-PI-DA, mi-SVM-PI-DA and MIL-CPB-PI-DA are better than sMIL-PI, mi-SVM-PI and MIL-CPB-PI, respectively, and they also outperform the existing domain adaptation approaches.

3.2 Related Work

3.2.1 Learning from Web Data

Researchers have proposed effective methods to employ massive web data for various computer vision applications [52, 72, 131, 145]. Torralba *et al.* [145] used a nearest neighbor (NN) based approach for object and scene recognition by leveraging a large dataset with 80 million tiny images. Fergus *et al.* [52] proposed a topic model based approach for object categorization by exploiting the images retrieved from Google image search, while Hwang and Grauman [72] employed kernel canonical correlation analysis (KCCA) for image retrieval using different features. Recently, Chen *et al.* [23] proposed the NEIL system for automatically labeling instances and extracting the visual relationships.

This chapter is more related to [91, 93, 94, 96, 97, 152], which used multi-instance learning approaches to explicitly cope with noise in the loose labels of web images or web videos. In particular, those works first partitioned the training images into small subsets. By treating each subset as a "bag" and the images in each bag as "instances", they formulated this task as a multi-instance learning problem. The bag-based MIL method Sparse MIL as well as its variant were used in [152] for image categorization, while an instance-based approach called MIL-CPB was developed in [96] for image retrieval. Moreover, a weighted MILBoost approach was proposed in [91] for video categorization. Besides the above multi-instance learning methods, some other approaches were also proposed to cope with label noise. For instance, Natarajan *et al.* [112] proposed two approaches to modify the loss function for learning with noisy labels, in which the first approach uses the unbiased estimator of loss function and the second approach uses a weighted loss function. Bootkrajang and Kaban [14] proposed a robust Multiple Kernel Logistic Regression algorithm (rMKLR), which incorporates the label flip probabilities in the loss function. However, the works in [14, 91, 96, 112, 152] did not consider the additional features in training data, and thus they can only employ the visual features for learning MIL classifiers for action/event recognition[1]. In contrast, we propose a new action/event recognition framework MIL-PI by incorporating the additional textual features of training samples as privileged information.

3.2.2 Learning with Additional Information

Our approach is motivated by the work on learning using privileged information (LUPI) [150], in which training data contains additional features (*i.e.*, privileged information) that are not available in the testing stage. Privileged information was also used for distance metric learning [55], multiple task learning [101] and learning to rank [135]. However, all those works only considered the supervised learning scenario using training data with accurate supervision. In contrast, we formulate a new MIL-PI framework in order to cope with noise in the loose labels of relevant training web videos.

[1]The work in [96] used both visual and textual features in the training process. However, it also requires the textual features in the testing process.

This chapter is also related to attribute based approaches [48, 54], in which the attribute classifiers are learnt to extract the mid-level features. However, the mid-level features can be extracted from both training and test images. Similarly, the classeme based approaches [92, 146] were proposed to use the training images from additionally annotated concepts to obtain the mid-level features. Those methods can be readily applied to our application by using the mid-level features as the main features to replace our current visual features (*i.e.*, the improved dense trajectory features [155] in our experiments). However, the additional textual features, which are not available in the test samples, can still be used as privileged information in our MIL-PI framework. Moreover, those works did not explicitly reduce the data distribution mismatch between the training and test samples as in our MIL-PI-DA framework.

3.2.3 Domain Adaptation

This chapter is also related to the domain adaptation methods [7, 9, 16, 39, 41, 45, 53, 60, 63, 71, 87, 95]. The previous domain adaptation approaches have been discussed in Section 2.1. However, these method requires the labeled training samples from the target domain, which are not required in our domain adaptation framework MIL-PI-DA. Moreover, our MIL-PI-DA framework achieves the best results for action/event recognition when the training and test samples are from different datasets.

3.3 Multi-Instance Learning Using Privileged Information

3.3.1 Problem Statement

Our task is to learn robust classifiers for action/event recognition by using loosely labeled web videos. Given any action/event name, relevant and irrelevant web videos can be automatically collected as training data by using tag-based video retrieval. Those relevant (*resp.*, irrelevant) videos can be used as positive (*resp.*, negative) training samples for learning classifiers for action/event recognition. However, not all those relevant videos are semantically related to the action/event name, because the web videos are generally

associated with noisy tags. Hence, we refer to those automatically collected web videos as loosely labeled web videos.

Moreover, although the test videos do not contain textual information, the additional textual features extracted from the training videos can still be used to improve the recognition performance. As shown in [150], the additional features that are only available in training data can be utilized as privileged information to help learn more robust classifiers for the main features (*i.e.*, the features that are available for both training and test data).

To this end, we propose a new learning framework called multi-instance learning using privileged information (MIL-PI) for action/event recognition, in which we not only take advantage of the additional textual descriptions (*i.e.*, privileged information) in training data but also effectively cope with noise in the loose labels of relevant training videos.

In particular, to cope with label noise in training data, we partition the relevant and irrelevant web videos into bags as in the recent works [91, 96, 152]. The training bags constructed from relevant samples are labeled as positive and those from irrelevant samples are labeled as negative. Let us represent the training data as $\{(\mathcal{B}_l, Y_l)\,|_{l=1}^{L}\}$, where $\mathcal{B}_l$ is a training bag, $Y_l \in \{+1, -1\}$ is the corresponding bag label, and L is the total number of training bags. Each training bag $\mathcal{B}_l$ consists of a number of training instances, *i.e.*, $\mathcal{B}_l = \{(\mathbf{x}_i, \tilde{\mathbf{x}}_i, y_i)|_{i\in\mathcal{I}_l}\}$, where $\mathcal{I}_l$ is the set of indices for the instances inside $\mathcal{B}_l$, $\mathbf{x}_i$ is the visual feature vector extracted from the i-th web video, $\tilde{\mathbf{x}}_i$ is the corresponding textual feature extracted from its surrounding textual descriptions, $y_i \in \{+1, -1\}$ is the ground truth label that indicates whether the i-th video is semantically related to the action/event name. Note the ground truth label y_i is unknown. Without loss of generality, we assume the positive bags are the first L^+ training bags.

In our framework, we use the generalized constraints for the MIL problem [96]. As shown in [96], the relevant samples usually contain a portion of positive samples, while it is more likely that the irrelevant samples are all negative samples. Namely, we have

$$\begin{cases} \sum_{i\in\mathcal{I}_l} \frac{y_i+1}{2} \geq \sigma|\mathcal{B}_l|, & \forall Y_l = 1, \\ y_i = -1, & \forall i \in \mathcal{I}_l \text{ and } Y_l = -1, \end{cases} \tag{3.1}$$

where $|\mathcal{B}_l|$ is the cardinality of the bag $\mathcal{B}_l$, and $\sigma > 0$ is a predefined ratio based on prior information. In other words, each positive bag is assumed to contain at least a portion

of true positive instances, and all instances in a negative bag are assumed to be negative samples.

Recall the textual descriptions associated with the training videos are also noisy, so privileged information may not be always reliable as in [135, 150]. Considering the labels of instances in the negative bags are known to be negative [96, 152], and the results after employing noisy privileged information for the instances in the negative bags are generally worse (see our experiments in Section 3.5.3), we only utilize privileged information for positive bags in our methods. However, it is worth mentioning that our method can be readily used to employ privileged information for the instances in all training bags.

In the following, we firstly introduce two LUPI approaches called SVM+ and partial SVM+ (pSVM+) that are related to this chapter. Then we propose a new bag-level MIL-PI method called sMIL-PI in Section 3.3.3 based on Sparse MIL (sMIL) [17], and also propose two instance-level MIL-PI methods called mi-SVM-PI and MIL-CPB-PI in Section 3.3.4 based on mi-SVM [4] and MIL-CPB [96], respectively.

3.3.2 Learning using Privileged Information

Let us denote the training data as $\{(\mathbf{x}_i, \tilde{\mathbf{x}}_i, y_i)|_{i=1}^n\}$, where $\mathbf{x}_i$ is main feature for the i-th training sample, $\tilde{\mathbf{x}}_i$ is the corresponding feature representation of privileged information which is not available for test data, $y_i \in \{+1, -1\}$ is the class label, and n is the total number of training samples. Here the class label y_i of each training sample is assumed to be given. The goal of Learning Using Privileged Information (LUPI) is to learn the classifier $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$, where $\phi(\cdot)$ is a nonlinear feature mapping function. We also define another nonlinear feature mapping function $\tilde{\phi}(\cdot)$ for privileged information. In some situations, privileged information may not be available for all the training samples. Particularly, when the training dataset contains l samples $\{(\mathbf{x}_i, \tilde{\mathbf{x}}_i, y_i)|_{i=1}^l\}$ with privileged information and $n - l$ samples $\{(\mathbf{x}_i, y_i)|_{i=l+1}^n\}$ without privileged information, the slack function can only be introduced for the l training samples with privileged information. We refer to this case of SVM+ as partial SVM+ or pSVM+ for short. The primal forms of SVM+ and pSVM+ have been fully introduced in Section 2.2 and thus we omit the details here.

3.3.3 Bag-level MIL using Privileged Information

The bag-level MIL methods [17, 24] focus on the classification of bags. As the labels of training bags are known, by transforming each training bag to one training sample, the MIL problem becomes a supervised learning problem. Such a strategy can also be applied to our MIL-PI framework, and we refer to our new method as *sMIL-PI*.

3.3.3.1 sMIL-PI

Let us denote $\psi(\mathcal{B}_l)$ as the feature mapping function which converts a training bag into a single feature vector. The feature mapping function in sMIL is defined as the mean of instances inside the bag, *i.e.*, $\psi(\mathcal{B}_l) = \frac{1}{|\mathcal{B}_l|}\sum_{i\in\mathcal{I}_l}\phi(\mathbf{x}_i)$, where $|\mathcal{B}_l|$ is the cardinality of the bag $\mathcal{B}_l$. Recall the labels for negative instances are assumed to be negative, so we only apply the feature mapping function on the positive training bags. For ease of presentation, we denote a set of virtual training samples $\{\mathbf{z}_j|_{j=1}^m\}$, in which $\mathbf{z}_1, \ldots, \mathbf{z}_{L^+}$ are the samples mapped from the positive bags $\{\psi(\mathcal{B}_j)|_{j=1}^{L^+}\}$, the remaining samples $\mathbf{z}_{L^+\!+1}, \ldots, \mathbf{z}_m$ are the instances $\{\phi(\mathbf{x}_i)|i \in \mathcal{I}_l, Y_l = -1\}$ in the negative bags.

When there is additional privileged information for training data, we define another feature mapping function $\tilde{\psi}(\mathcal{B}_l)$ on each training bag as the mean of instances inside the bag by using privileged information, *i.e.*, $\tilde{\mathbf{z}}_j = \tilde{\psi}(\mathcal{B}_j) = \frac{1}{|\mathcal{B}_j|}\sum_{i\in\mathcal{I}_j}\tilde{\phi}(\tilde{\mathbf{x}}_i)$ for $j = 1, \ldots, L^+$. Based on the SVM+ formulation, the objective of our sMIL-PI can be formulated as,

$$
\begin{aligned}
\min_{\mathbf{w},b,\tilde{\mathbf{w}},\tilde{b},\boldsymbol{\eta}} \quad & \frac{1}{2}\left(\|\mathbf{w}\|^2 + \gamma\|\tilde{\mathbf{w}}\|^2\right) + C_1\sum_{j=1}^{L^+}\xi(\tilde{\mathbf{z}}_j) + \sum_{j=L^+\!+1}^{m}\eta_j, \\
\text{s.t.} \quad & \mathbf{w}'\mathbf{z}_j + b \geq p_j - \xi(\tilde{\mathbf{z}}_j), \quad \forall j = 1, \ldots, L^+, && (3.2)\\
& \mathbf{w}'\mathbf{z}_j + b \leq -1 + \eta_j, \quad \forall j = L^+\!+1, \ldots, m, && (3.3)\\
& \xi(\tilde{\mathbf{z}}_j) \geq 0, \quad \forall j = 1, \ldots, L^+, && (3.4)\\
& \eta_j \geq 0, \qquad \forall j = L^+\!+1, \ldots, m, && (3.5)
\end{aligned}
$$

where $\mathbf{w}$ and b are the variables of the classifier $f(\mathbf{z}) = \mathbf{w}'\mathbf{z} + b$, γ, C_1 are the tradeoff parameters, $\boldsymbol{\eta} = [\eta_{L^+\!+1}, \ldots, \eta_m]'$, the slack function is defined as $\xi(\tilde{\mathbf{z}}_j) = \tilde{\mathbf{w}}'\tilde{\mathbf{z}}_j + \tilde{b}$, and p_j is the virtual label for the virtual sample $\mathbf{z}_j$. In sMIL [17], the virtual label is calculated

by leveraging the instance labels of each positive bag. As sMIL assumes that there is at least one true positive sample in each positive bag, the virtual label of positive virtual sample $\mathbf{z}_j$ is $p_j = \frac{1-(|\mathcal{B}_j|-1)}{|\mathcal{B}_j|} = \frac{2-|\mathcal{B}_j|}{|\mathcal{B}_j|}$. Similarly, for our sMIL-PI using the generalized MIL constraints in (3.1), we can derive it as $p_j = \frac{\sigma|\mathcal{B}_j|-(1-\sigma)|\mathcal{B}_j|}{|\mathcal{B}_j|} = 2\sigma - 1$. Note the difference between (3.2) and pSVM+ is that we use the bag-level features instead of instance-level features and change the margin in the constraint from 1 to p_j.

By introducing dual variable $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]'$ for the constraints in (3.2) and (3.3), and also introducing dual variable $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{L^+}]'$ for the constraints in (3.4), respectively, we arrive at the dual from of (3.2) as follows,

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & -\mathbf{p}'\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{K} \circ \mathbf{y}\mathbf{y}')\boldsymbol{\alpha} + \frac{1}{2\gamma}(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1})'\tilde{\mathbf{K}}(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1}), \\ \text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{y} = 0, \quad \mathbf{1}'(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1}) = 0, \\ & \bar{\boldsymbol{\alpha}} \leq \mathbf{1}, \quad \boldsymbol{\alpha} \geq \mathbf{0}, \quad \boldsymbol{\beta} \geq \mathbf{0}, \end{aligned} \tag{3.6}$$

where $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{L^+}$ and $\bar{\boldsymbol{\alpha}} \in \mathbb{R}^{m-L^+}$ are from $\boldsymbol{\alpha} = [\hat{\boldsymbol{\alpha}}', \bar{\boldsymbol{\alpha}}']'$, $\mathbf{y} = [\mathbf{1}'_{L^+}, -\mathbf{1}'_{m-L^+}]'$ is the label vector, $\mathbf{p} = [p_1, \ldots, p_{L^+}, \mathbf{1}'_{m-L^+}]' \in \mathbb{R}^m$, $\mathbf{K} \in \mathbb{R}^{m\times m}$ is the kernel matrix constructed by using the visual features, $\tilde{\mathbf{K}} \in \mathbb{R}^{L^+\times L^+}$ is the kernel matrix constructed by using privileged information (*i.e.*, the textual features). The above problem is jointly convex in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and can be solved by optimizing a quadratic programming problem. sMIL-PI can also be solved in the primal form (3.2) using stochastic gradient descent, but this is not the focus of this work.

3.3.4 Instance-level MIL using Privileged Information

Different from the bag-level MIL methods, the instance-level MIL methods [4, 96] directly solve the classification problem for the instances. However, the labels of training instances are unknown, so one needs to infer the instance labels when learning the MIL classifier. Inspired by the works in [4, 96], we formulate the instance-level MIL-PI problem

as follows,

$$\min_{\substack{\mathbf{y}\in\mathcal{Y}\\ \tilde{\mathbf{w}},\tilde{b},\mathbf{w},b,\boldsymbol{\eta}}} \quad \frac{1}{2}\left(\|\mathbf{w}\|^2+\gamma\|\tilde{\mathbf{w}}\|^2\right)+\ C_1\sum_{i=1}^{n^+}\xi(\tilde{\phi}(\tilde{\mathbf{x}}_i))+\sum_{i=n^+\!+1}^{n}\eta_i,$$

$$\text{s.t.} \quad y_i(\mathbf{w}'\phi(\mathbf{x}_i)+b)\geq 1-\xi(\tilde{\phi}(\tilde{\mathbf{x}}_i)), \tag{3.7}$$

$$\xi(\tilde{\phi}(\tilde{\mathbf{x}}_i))\geq 0, \qquad i=1,\ldots,n^+, \tag{3.8}$$

$$y_i(\mathbf{w}'\phi(\mathbf{x}_i)+b)\geq 1-\eta_i, \tag{3.9}$$

$$\eta_i\geq 0, \qquad i=n^+\!+1,\ldots,n, \tag{3.10}$$

where $\mathcal{Y}=\{\mathbf{y}|\mathbf{y}$ satisfies the constraints in (3.1)$\}$ is the feasible set of labelings for training instances with $\mathbf{y}=[y_1,\ldots,y_n]'$ being a feasible label vector, $\boldsymbol{\eta}=[\eta_{n^+\!+1},\ \ldots,\ \eta_n]'$, γ and C_1 are the tradeoff parameters, and $\xi(\tilde{\phi}(\tilde{\mathbf{x}}))=\tilde{\mathbf{w}}'\tilde{\phi}(\tilde{\mathbf{x}})+\tilde{b}$ is the slack function similarly as in sMIL-PI. The difference between (3.7) and pSVM+ is that the label vector $\mathbf{y}$ is also a variable which needs to be optimized in (3.7).

Note in this formulation, we need to infer the instance labels in the label vector $\mathbf{y}$, and simultaneously learn the classifier. It is a nontrivial mixed-integer programming problem, because the number of all possible labelings (*i.e.*, $|\mathcal{Y}|$) increases exponentially *w.r.t.* the number of positive instances n^+. In mi-SVM [4], an iterative approach is adopted to learn an SVM classifier and update the label vector $\mathbf{y}$ by using the prediction from the learnt classifier. In MIL-CPB [96], a multiple kernel learning (MKL) based approach is proposed to learn an optimal kernel by optimizing the linear combination of the label kernels associated with all possible label vectors. We respectively apply those two strategies to our objective function in (3.7), and develop two instance-level MIL-PI approaches, mi-SVM-PI and MIL-CPB-PI.

3.3.4.1 mi-SVM-PI

In mi-SVM-PI, we adopt the strategy in mi-SVM [4] and use the similar iterative updating approach to solve our instance based MIL-PI problem in (3.7). Specifically, as shown in Algorithm 1, we first initialize the label vector $\mathbf{y}$ by setting the labels of instances as their corresponding bag labels. Then we employ the alternating optimization method to iteratively solve a pSVM+ problem by using the current label vector $\mathbf{y}$, and infer $\mathbf{y}$ by

Algorithm 1 The optimization algorithm for solving the objective function of our mi-SVM-PI

Require: Training data $\{(\mathcal{B}_l, Y_l)|_{l=1}^{L}\}$ (see Section 3.1).

1: Initialize $\mathbf{y} = [\mathbf{1}'_{n^+}, -\mathbf{1}'_{n-n^+}]'$.
2: **repeat**
3: Train $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ by solving a pSVM+ problem based on $\mathbf{y}$.
4: Calculate the decision values of training instances by using the learnt $f(\mathbf{x})$.
5: Based on the decision values, obtain $\mathbf{y}$ that satisfies the constraints in (3.1).
6: **until** The labeling vector $\mathbf{y}$ does not change.

Ensure: The learnt classifier $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$.

using the learnt classifier $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ at the previous iteration. For any positive bag $\mathcal{B}_l$ where the constraint in (3.1) is not satisfied, we additionally set the labels of $\sigma|\mathcal{B}_l|$ instances with the largest decision values in this positive bag to be positive. The above process is repeated until $\mathbf{y}$ does not change.

3.3.4.2 MIL-CPB-PI

The instance-level MIL-PI formulation in (3.7) can also be solved by optimizing an MKL problem as in MIL-CPB, as discussed in [96]. The main idea is to firstly relax the duality of (3.7) to its tight lower bound. Then we show that the relaxed problem shares a similar form with the MKL problem, and thus can be similarly optimized by solving a convex problem in the primal form.

To derive the solution of our MIL-CPB-PI method, we absorb the bias term b in (3.7) into $\mathbf{w}$ by augmenting the feature vector $\phi(\mathbf{x}_i)$ with an additional dimension with its value being 1 similarly as in [96]. By respectively introducing the dual variables $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{n^+}$, $\bar{\boldsymbol{\alpha}} \in \mathbb{R}^{n-n^+}$ and $\boldsymbol{\beta} \in \mathbb{R}^{n^+}$ for the constraints in (3.7), (3.9), and (3.8), and defining $\boldsymbol{\alpha} = [\hat{\boldsymbol{\alpha}}', \bar{\boldsymbol{\alpha}}']' \in \mathbb{R}^n$, we arrive at the dual problem of (3.7) as follows,

$$\min_{\mathbf{y}\in\mathcal{Y}} \max_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathcal{S}} \quad \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{Q}\circ\mathbf{y}\mathbf{y}')\boldsymbol{\alpha} - \frac{1}{2\gamma}(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1})'\tilde{\mathbf{K}}(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1}),$$

where $\mathbf{Q} = \mathbf{K} + \mathbf{1}\mathbf{1}'$ with $\mathbf{K} \in \mathbb{R}^{n \times n}$ being the kernel matrix constructed by using the visual features, $\tilde{\mathbf{K}} \in \mathbb{R}^{n^+ \times n^+}$ is the kernel matrix constructed by using the textual features, $\mathcal{S} = \{(\boldsymbol{\alpha}, \boldsymbol{\beta}) | \mathbf{1}'(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1}) = 0, \bar{\boldsymbol{\alpha}} \leq \mathbf{1}, \boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta} \geq \mathbf{0}\}$ is the feasible set.

Note that each label vector $\mathbf{y}$ forms a label kernel $\mathbf{y}\mathbf{y}'$ in the duality in (3.11). Inspired by [96], instead of directly optimizing an optimal label kernel $\mathbf{y}\mathbf{y}'$, we seek for an optimal linear combination of all possible label kernels. We write the relaxed problem as follows,

$$\min_{\mathbf{d} \in \mathcal{D}} \max_{(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{S}} \quad \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}' \left(\sum_{t=1}^{T} d_t \mathbf{Q} \circ \mathbf{y}_t \mathbf{y}_t' \right) \boldsymbol{\alpha} - \frac{1}{2\gamma} (\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1})' \tilde{\mathbf{K}} (\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1}),$$

where $\mathbf{y}_t \in \mathcal{Y}$ is the t-th label vector in the feasible set $\mathcal{Y}$, $T = |\mathcal{Y}|$ is the total number of label vectors in $\mathcal{Y}$, d_t is the combination coefficient of the label kernel $\mathbf{y}_t\mathbf{y}_t'$, $\mathbf{d} = [d_1, \ldots, d_T]'$ is the vector which contains all the combination coefficients, and $\mathcal{D} = \{\mathbf{d} | \mathbf{d}'\mathbf{1} = 1, \mathbf{d} \geq \mathbf{0}\}$ is the feasible set of $\mathbf{d}$.

Intuitively, for the optimization problem in (3.11), we search for an optimal $\mathbf{y}\mathbf{y}'$ in $\mathcal{Y}$, which is a set of discrete points in the space $\mathbb{R}^{n \times n}$. The optimization problem in (3.11) is a Mixed Integer Programming (MIP) problem and is NP-hard. In contrast, the optimization problem in (3.11) is in the convex hull of all possible $\mathbf{y}_t\mathbf{y}_t'$'s in $\mathbb{R}^{n \times n}$, which is a continuous region and makes the problem easier to be solved. Actually, by considering each $(\mathbf{Q} \circ \mathbf{y}_t\mathbf{y}_t')$ as a base kernel, the optimization problem in (3.11) shares a similar form with the MKL problem, which can be solved by optimizing a convex optimization problem in its primal problem [83].

The main challenge for applying the existing MKL techniques to solve (3.11) is that we have too many base kernels, *i.e.*, $T = |\mathcal{Y}|$ is possibly exponential to the number of positive instances n^+. Inspired by Infinite Kernel Learning (IKL) [57], we employ the cutting-plane algorithm to solve it. Specifically, by introducing a dual variable τ for the constraint $\mathbf{d}'\mathbf{1} = 1$ in $\mathcal{D}$, we arrive at the duality of (3.11) as follows,

$$\begin{aligned} \max_{\tau, (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{S}} \quad & 1'\boldsymbol{\alpha} - \frac{1}{2\gamma} (\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1})' \tilde{\mathbf{K}} (\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1}) - \tau, \\ \text{s.t.} \quad & \frac{1}{2}\boldsymbol{\alpha}' (\mathbf{Q} \circ \mathbf{y}_t\mathbf{y}_t') \boldsymbol{\alpha} \leq \tau, \quad \forall t = 1, \ldots, T. \end{aligned} \tag{3.11}$$

As each of the constraints in (3.11) corresponds to a base kernel $\mathbf{Q} \circ \mathbf{y}_t\mathbf{y}_t'$, there are many constraints (*i.e.*, $T = |\mathcal{Y}|$) in the above problem. The main idea of the cutting-plane

Algorithm 2 Approximately find the most violated y_t

1: Initialize $y_i = 1$ for all instances in positive bags $\{(\mathcal{B}_l, Y_l)|_{l=1}^{L^+}\}$.
2: **repeat**
3: **for** each positive bag $\mathcal{B}_l$ **do**
4: Fix the labeling of all the other positive bags, find the optimal $\mathbf{y}_l$ that maximizes (3.12) by enumerating all the feasible labeling candidates of y_l in $\mathcal{B}_l$.
5: **end for**
6: **until** no labels are changed.

Algorithm 3 The optimization algorithm for solving the objective function of our MIL-CPB-PI

Require: Training data $\{(\mathcal{B}_l, Y_l)|_{l=1}^{L}\}$ (see Section 3.1).

1: Initialize $\mathcal{C} = \{\mathbf{y}_0\}$ with $\mathbf{y}_0 = [\mathbf{1}'_{n^+}, -\mathbf{1}'_{n-n^+}]'$, and set $r = 0$.
2: **repeat**
3: Set $r \leftarrow r + 1$.
4: Based on $\mathcal{Y} = \mathcal{C}$, solve for $(\mathbf{d}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ by optimizing the MKL problem in (3.11).
5: Set $\mathcal{C} \leftarrow \mathcal{C} \bigcup \mathbf{y}_r$ where $\mathbf{y}_r$ is obtained by solving (3.12).
6: **until** The objective of (3.11) converges.

Ensure: The learnt classifier $f(\mathbf{x})$.

algorithm is to approximate (3.11) by using only a few constraints. Specifically, we start from one constraint, and solve for $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and τ. If there is any constraint that cannot be satisfied, we add this constraint into the current optimization problem, and resolve for $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and τ again. The above process is repeated until all constraints are satisfied.

To find the violated constraint, we maximize the left-hand side of the constraint in (3.11), which can be written as follows,

$$\max_{\mathbf{y} \in \mathcal{Y}} \mathbf{y}'(\mathbf{Q} \circ \boldsymbol{\alpha}\boldsymbol{\alpha}')\mathbf{y}, \tag{3.12}$$

which can be solved approximately by enumerating the instance labels in a bag-by-bag fashion when the size of each bag is not too large, as discussed in Algorithm 2.

The algorithm of our MIL-CPB-PI is listed in Algorithm 3. We first initialize the labeling set as $\mathcal{C} = \{\mathbf{y}_0\}$. Then we iteratively train an MKL classifier by solving (3.11) based on $\mathcal{Y} = \mathcal{C}$ and update the labeling set $\mathcal{C}$ by adding the violated $\mathbf{y}_r$, which is obtained by solving (3.12) based on the current $\boldsymbol{\alpha}$. This process is repeated until the objective of (3.11) converges.

As we only need to solve an MKL problem based on a small set of base kernels at each iteration, the optimization procedure is much more efficient. It can be solved similarly as in the existing MKL solver in [83].

Moreover, the objective of (3.11) decreases monotonously as r increases, because the labeling set is enlarged at each iteration. The final classifier can be presented as $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ with $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \tilde{y}_i \phi(\mathbf{x}_i)$, where α_i is the i-th entry in the final dual variable $\boldsymbol{\alpha}$, and $\tilde{y}_i = \sum_{t=1}^{r} d_t y_{t,i}$ with $y_{t,i}$ being the i-th entry of $\mathbf{y}_t$.

3.4 Domain Adaptive MIL-PI

The training web videos often have very different statistical properties from the test videos, which is also known as the dataset bias problem [144]. To reduce the domain distribution mismatch, we proposed a new domain adaptation framework by re-weighting the source domain samples when learning the classifiers. In the following, we develop our domain adaptation framework, which is referred to as MIL-PI-DA. Moreover, we also extend sMIL-PI (*resp.*, mi-SVM-PI, MIL-CPB-PI) to sMIL-PI-DA (*resp.*, mi-SVM-PI-DA, MIL-CPB-PI-DA).

This chapter is inspired by the Kernel Mean Matching (KMM) method [71], in which the source domain samples are reweighted by minimizing the Maximum Mean Discrepancy (MMD) between two domains. However, KMM is a two-stage method, in which they first learn the weights for the source domain samples and then utilize the weights to train a weighted SVM. Though the recent work [28] proposed to combine the primal formulation of weighted-SVM and a regularizer based on the MMD criterion, their objective function is non-convex, and thus the global optimal solution cannot be guaranteed. To this end, we propose a convex formulation by adding the regularizer based on the MMD criterion to the dual formulation of our MIL-PI framework, which leads to a convex objective function as discussed in Section 3.4.1. Formally, let us denote the target

domain samples as $\{\mathbf{x}_i^t|_{i=1}^{n_t}\}$, and denote $\phi(\mathbf{x}_i^t)$ as the corresponding nonlinear feature. To distinguish the two domains, we append a superscript s to the source domain samples, *i.e.*, $\{\mathbf{x}_i^s|_{i=1}^{n_s}\}$ and denote $\phi(\mathbf{x}_i^s)$ as the corresponding nonlinear feature.

3.4.1 Bag-level Domain Adaptive MIL-PI

We propose a bag-level domain adaptive MIL-PI method sMIL-PI-DA, which is extended from sMIL-PI. We denote the objective in (3.6) as $H(\boldsymbol{\alpha},\boldsymbol{\beta}) = -\mathbf{p}'\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{K}\circ\mathbf{y}\mathbf{y}')\boldsymbol{\alpha} + \frac{1}{2\gamma}(\hat{\boldsymbol{\alpha}}+\boldsymbol{\beta}-C_1\mathbf{1})'\tilde{\mathbf{K}}(\hat{\boldsymbol{\alpha}}+\boldsymbol{\beta}-C_1\mathbf{1})$, and also denote the weights for source domain samples as $\boldsymbol{\theta} = [\theta_1,\ldots,\theta_m]'$ with each θ_i being the weight for the i-th source domain sample. We also denote $\{\mathbf{z}_i^s|_{i=1}^m\}$ (*resp.*, $\{\mathbf{z}_i^t|_{i=1}^{n_t}\}$) as the set of virtual samples in the source (*resp.*, target) domain, which are used in our sMIL-PI-DA. Note that $\mathbf{z}_i^s$'s and $\mathbf{z}_i^t$'s denote the visual features. Then, we formulate our sMIL-PI-DA method as follows,

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta}} \quad H(\boldsymbol{\alpha},\boldsymbol{\beta}) + \frac{\mu}{2}\|\frac{1}{m}\sum_{i=1}^{m}\theta_i\mathbf{z}_i^s - \frac{1}{n_t}\sum_{i=1}^{n_t}\mathbf{z}_i^t\|^2 \tag{3.13}$$

$$\text{s.t.} \quad \boldsymbol{\alpha}'\mathbf{y} = 0, \quad \mathbf{1}'(\hat{\boldsymbol{\alpha}}+\boldsymbol{\beta}-C_1\mathbf{1}) = 0, \tag{3.14}$$

$$\bar{\boldsymbol{\alpha}} \le \mathbf{1}, \quad \boldsymbol{\beta} \ge \mathbf{0} \tag{3.15}$$

$$\mathbf{0} \le \boldsymbol{\alpha} \le C_2\boldsymbol{\theta}, \quad \mathbf{1}'\boldsymbol{\theta} = m, \tag{3.16}$$

where C_2 is a parameter and θ_i is the weight for $\mathbf{z}_i^s$. The last term in (3.13) is a regularizer based on the MMD criterion, which aims to reduce the domain distribution mismatch between two domains by reweighting the source domain samples as in KMM, and the constraints in (3.14) and (3.15) are from sMIL-PI. Note in (3.16), we use the box constraint $\mathbf{0} \le \boldsymbol{\alpha} \le C_2\boldsymbol{\theta}$ to regularize the dual variable $\boldsymbol{\alpha}$, which is similarly used in weighted SVM [71]. In (3.16), the second constraint $\mathbf{1}'\boldsymbol{\theta} = m$ is used to enforce the expectation of sample weights to be 1. The problem in (3.13) is jointly convex with respect to $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, and thus we can obtain the global optimum by optimizing a quadratic programming problem.

Interestingly, the primal form of (3.13) is closely related to the formulation of SVM+, as described below,

Proposition 1 *The primal form of (3.13) is equivalent to the following problem,*

$$\min_{\mathbf{w},b,\tilde{\mathbf{w}},\tilde{b},\hat{\mathbf{w}},\hat{b},\boldsymbol{\eta}} \quad J(\mathbf{w},b,\tilde{\mathbf{w}},\tilde{b},\boldsymbol{\eta}) + \frac{\lambda}{2}\|\hat{\mathbf{w}}-\rho\mathbf{v}\|^2 + C_2\sum_{i=1}^{m}\zeta(\mathbf{z}_i^s), \tag{3.17}$$

$$s.t. \quad \mathbf{w}'\mathbf{z}_i^s + b \geq p_i - \xi(\tilde{\mathbf{z}}_i^s) - \zeta(\mathbf{z}_i^s), \forall i = 1,\ldots,L^+, \tag{3.18}$$

$$\mathbf{w}'\mathbf{z}_i^s + b \leq -1 + \eta_i + \zeta(\mathbf{z}_i^s), \forall i = L^+ + 1,\ldots,m, \tag{3.19}$$

$$\xi(\tilde{\mathbf{z}}_i^s) \geq 0, \quad \forall i = 1,\ldots,L^+, \tag{3.20}$$

$$\eta_i \geq 0, \qquad \forall i = L^+ + 1,\ldots,m, \tag{3.21}$$

$$\zeta(\mathbf{z}_i^s) \geq 0, \quad \forall i = 1,\ldots,m, \tag{3.22}$$

where $J(\mathbf{w},b,\tilde{\mathbf{w}},\tilde{b},\boldsymbol{\eta}) = \frac{1}{2}\left(\|\mathbf{w}\|^2 + \gamma\|\tilde{\mathbf{w}}\|^2\right) + C_1\sum_{j=1}^{L^+}\xi(\tilde{\mathbf{z}}_j^s) + \sum_{j=L^+1}^{m}\eta_j$ *is the objective function in* (3.2), $\zeta(\mathbf{z}_i^s) = \hat{\mathbf{w}}'\mathbf{z}_i^s + \hat{b}$, $\mathbf{v} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{z}_i^s - \frac{1}{n_t}\sum_{i=1}^{n_t}\mathbf{z}_i^t$, $\lambda = \frac{(mC_2)^2}{\mu}$ *and* $\rho = \frac{mC_2}{\lambda}$.

The detailed proof is provided in Appendix A.

Compared with the objective function in (3.2), we introduce one more slack function $\zeta(\mathbf{z}_i^s) = \hat{\mathbf{w}}'\mathbf{z}_i^s + \hat{b}$, and also regularize the weight vector of this slack function by using the regularizer $\|\hat{\mathbf{w}} - \rho\mathbf{v}\|^2$. Recall that the witness function in MMD is defined as $g(\mathbf{z}) = \frac{1}{\|\mathbf{v}\|}\mathbf{v}'\mathbf{z}$ [64], which can be deemed as the mean similarity between $\mathbf{z}$ and the source domain samples (*i.e.*, $\frac{1}{m}\sum_{i=1}^{m}\mathbf{z}_i^{s\prime}\mathbf{z}$) minus the mean similarity between $\mathbf{z}$ and the target domain samples (*i.e.*, $\frac{1}{n_t}\sum_{i=1}^{n_t}\mathbf{z}_i^{t\prime}\mathbf{z}$). In other words, we conjecture that the witness function outputs a lower value when the sample $\mathbf{z}$ is closer to the target domain samples and vice versa. By using the regularizer $\|\hat{\mathbf{w}} - \rho\mathbf{v}\|^2$, we expect the new slack function $\zeta(\mathbf{z}_i^s) = \hat{\mathbf{w}}'\mathbf{z}_i^s + \hat{b}$ shares the similar trend[2] with the witness function $g(\mathbf{z}_i^s) = \frac{1}{\|\mathbf{v}\|}\mathbf{v}'\mathbf{z}_i^s$. As a result, the training error of the training sample $\mathbf{z}_i^s$ (*i.e.*, $\xi(\tilde{\mathbf{z}}_i^s) + \zeta(\mathbf{z}_i^s)$ for the samples in the positive bags or $\eta_i + \zeta(\mathbf{z}_i^s)$ for the negative samples) will tend to be lower if it is closer to the target domain, which is helpful for learning a more robust classifier to better predict the target domain samples.

3.4.2 Instance-level Domain Adaptive MIL-PI

Besides the bag-level MIL method sMIL, we can also incorporate the instance-level MIL methods, mi-SVM and MIL-CPB, into our MIL-PI-DA framework. We refer to our new

[2]The bias term $\hat{b}$ and the scalar terms ρ and $\frac{1}{\|\mathbf{v}\|}$ will not change the trend of functions.

approaches as mi-SVM-PI-DA and MIL-CPB-PI-DA, respectively.

3.4.2.1 mi-SVM-PI-DA

To derive the formulation of mi-SVM-PI-DA, we firstly write the duality of the mi-SVM-PI problem in (3.7) as follows,

$$\begin{aligned}\min_{\mathbf{y}\in\mathcal{Y}}\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & J(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{y}) \doteq \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{K}\circ\mathbf{y}\mathbf{y}')\boldsymbol{\alpha} - \frac{1}{2\gamma}(\hat{\boldsymbol{\alpha}}+\boldsymbol{\beta}-C_1\mathbf{1})'\tilde{\mathbf{K}}(\hat{\boldsymbol{\alpha}}+\boldsymbol{\beta}-C_1\mathbf{1}),\\ \text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{y}=0, \quad \mathbf{1}'(\hat{\boldsymbol{\alpha}}+\boldsymbol{\beta}-C_1\mathbf{1})=0,\\ & \bar{\boldsymbol{\alpha}}\leq\mathbf{1}, \quad \boldsymbol{\alpha}\geq\mathbf{0}, \quad \boldsymbol{\beta}\geq\mathbf{0}.\end{aligned}$$

where $\boldsymbol{\alpha}=[\hat{\boldsymbol{\alpha}}',\bar{\boldsymbol{\alpha}}']'$ and $\boldsymbol{\beta}$ are the dual variables defined similarly as in the duality of MIL-CPB-PI in (3.11).

Similarly as in sMIL-PI-DA, we also introduce the MMD based regularizer to (3.23) in order to reduce the domain distribution mismatch, which leads to our mi-SVM-PI-DA problem as follows,

$$\begin{aligned}\min_{\mathbf{y}\in\mathcal{Y}}\max_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta}} \quad & J(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{y}) - \frac{\mu}{2}\|\frac{1}{n_s}\sum_{i=1}^{n_s}\theta_i\mathbf{x}_i^s - \frac{1}{n_t}\sum_{i=1}^{n_t}\mathbf{x}_i^t\|^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{y}=0, \quad \mathbf{1}'(\hat{\boldsymbol{\alpha}}+\boldsymbol{\beta}-C_1\mathbf{1})=0,\\ & \bar{\boldsymbol{\alpha}}\leq\mathbf{1}, \quad \boldsymbol{\beta}\geq\mathbf{0},\\ & \mathbf{0}\leq\boldsymbol{\alpha}\leq C_2\boldsymbol{\theta}, \quad 1'\boldsymbol{\theta}=n_s,\end{aligned} \tag{3.23}$$

where n_s is the number of source domain samples, n_t is the number of target domain samples. Similarly as in weighted SVM, the box constraint $\mathbf{0}\leq\boldsymbol{\alpha}\leq C_2\boldsymbol{\theta}$ is used, in which each θ_i is the weight for the i-th source domain sample. Note we minus the MMD based regularizer in (3.23), as the inner optimization problem is a maximization problem.

Similarly as in mi-SVM-PI, we solve the optimization problem in (3.23) in an iterative approach. When the label vector $\mathbf{y}$ is fixed, the subproblem can be written as,

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta}} \quad & -J(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{y}) + \frac{\mu}{2}(\frac{1}{n_s^2}\boldsymbol{\theta}'\mathbf{K}\boldsymbol{\theta} - \frac{2}{n_s n_t}\boldsymbol{\theta}'\mathbf{K}_{st}\mathbf{1}) \\ \text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{y}=0, \quad \mathbf{1}'(\hat{\boldsymbol{\alpha}}+\boldsymbol{\beta}-C_1\mathbf{1})=0,\\ & \bar{\boldsymbol{\alpha}}\leq\mathbf{1}, \quad \boldsymbol{\beta}\geq\mathbf{0},\\ & \mathbf{0}\leq\boldsymbol{\alpha}\leq C_2\boldsymbol{\theta}, \quad 1'\boldsymbol{\theta}=n_s,\end{aligned} \tag{3.24}$$

Algorithm 4 The optimization algorithm for solving the objective function of our mi-SVM-PI-DA

Require: Training data $\{(\mathcal{B}_l, Y_l)|_{l=1}^{L}\}$ (see Section 3.1).

1: Initialize $\mathbf{y} = [\mathbf{1}'_{n^+}, -\mathbf{1}'_{n_s-n^+}]'$.

2: **repeat**

3: Train $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ by solving the subproblem in (3.24) based on $\mathbf{y}$.

4: Calculate the decision values of training instances by using the learnt $f(\mathbf{x})$.

5: Based on the decision values, obtain $\mathbf{y}$ that satisfies the constraints in (3.1).

6: **until** The labeling vector $\mathbf{y}$ does not change.

Ensure: The learnt classifier $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$.

where $\mathbf{K}_{st} \in \mathbb{R}^{n_s \times n_t}$ is the kernel matrix measuring the similarity between the training samples and test samples by using visual features.

We describe the algorithm for solving mi-SVM-PI-DA in Algorithm 4. We first initialize the label vector $\mathbf{y}$ by setting the labels of instances as their corresponding bag labels. Then we iteratively solve the inner optimization problem based on the current $\mathbf{y}$, and infer $\mathbf{y}$ by using the learnt classifier $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ from the previous iteration. The inner optimization problem can be solved by optimizing a convex quadratic programming problem as in (3.24). For any positive bag $\mathcal{B}_l$ where the constraint in (3.1) is not satisfied, we additionally set the labels of $\sigma|\mathcal{B}_l|$ instances with the largest decision values in this positive bag to be positive. The above process is repeated until $\mathbf{y}$ does not change.

Similarly as sMIL-PI-DA, the primal form of (3.23) is is also related to the formulation of SVM+ and the details are ignored here.

3.4.2.2 MIL-CPB-PI-DA

Let us denote the objective of the duality of MIL-CPB-PI in (3.11) as $J(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{d}) = \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}'\left(\sum_{t=1}^{T} d_t \mathbf{Q} \circ \mathbf{y}_t\mathbf{y}_t'\right)\boldsymbol{\alpha} - \frac{1}{2\gamma}(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1})'\tilde{\mathbf{K}}(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1})$. Similarly, we can reduce the domain distribution mismatch by using a MMD based regularizer. We arrive at the objective function of our MIL-CPB-PI-DA as follows,

$$\begin{aligned} \min_{\mathbf{d}\in\mathcal{D}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta}} \quad & J(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{d}) - \frac{\mu}{2}(\frac{1}{n_s^2}\boldsymbol{\theta}'\mathbf{K}\boldsymbol{\theta} - \frac{2}{n_s n_t}\boldsymbol{\theta}'\mathbf{K}_{st}\mathbf{1}) \\ \text{s.t.} \quad & \mathbf{1}'(\hat{\boldsymbol{\alpha}} + \boldsymbol{\beta} - C_1\mathbf{1}) = 0, \\ & \bar{\boldsymbol{\alpha}} \leq \mathbf{1}, \quad \boldsymbol{\beta} \geq \mathbf{0}, \\ & \mathbf{0} \leq \boldsymbol{\alpha} \leq C_2\boldsymbol{\theta}, \quad 1'\boldsymbol{\theta} = n_s, \end{aligned} \tag{3.25}$$

which can be solved similarly as in Algorithm 3. The only difference is that we have one more MMD regularizer in the inner optimization problem. Therefore, for MIL-CPB-PI-DA, we need solve for $(\mathbf{d}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta})$ at Step 4 of Algorithm 3 by optimizing the MKL problem in (3.25) based on the current $\mathcal{Y}$. The final classifier can be presented as $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b$ with $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \tilde{y}_i \phi(\mathbf{x}_i)$, where α_i is the i-th entry in the final dual variable $\boldsymbol{\alpha}$, and $\tilde{y}_i = \sum_{t=1}^{r} d_t y_{t,i}$ with $y_{t,i}$ being the i-th entry of $\mathbf{y}_t$.

Similarly as sMIL-PI-DA, the primal form of (3.25) is is also related to the formulation of SVM+ and the details are ignored here.

Discussion: In summary, we propose sMIL-PI (3.2), mi-SVM-PI (3.7), and MIL-CPB-PI (3.11) for multi-instance learning using privileged information, in which mi-SVM-PI and MIL-CPB-PI shares the common general primal form (3.7). Then, we extend sMIL-PI (*resp.*, mi-SVM-PI and MIL-CPB-PI) to sMIL-PI-DA (3.13) (*resp.*, mi-SVM-PI-DA (3.23) and MIL-CPB-PI-DA (3.25)) for domain adaptation by adding an MMD regularizer in the dual form.

3.5 Experiments

In this chapter, we evaluate our proposed methods for action and event recognition by using loosely labeled web videos as training data. However, it is worth mentioning that our newly proposed methods can be readily used for other applications like image retrieval and image categorization.

3.5.1 Video Event Recognition

Datasets and Features: We evaluate our proposed methods for video event recognition on the benchmark datasets Kodak [106] and CCV [79].

We construct a new training dataset called "Flickr", which contains the web videos crawled from Flickr by using six event names (*i.e.*, "birthday", "picnic", "parade", "show", "sports" and "wedding") as the queries. We remove the web videos if they are too short (*i.e.*, the file size is smaller than 5M) or too long (*i.e.*, the file size is larger than 100M). Finally we keep the top 300 web videos for each query as the relevant videos. For each query, we randomly sample the same number of Flickr videos that do not contain this query as one of the textual descriptions as irrelevant videos.

The Kodak dataset was used in [41] and [44], which contains 195 consumer videos from six event classes (*i.e.*, "birthday", "picnic", "parade", "show", "sports" and "wedding"). The CCV dataset [79] collected by Columbia University was also used in [44]. It consists of a training set of 4659 videos and a test set of 4658 videos from 20 semantic categories. Following [44], we only use the videos from the event related categories and we also merge "wedding ceremony", "wedding reception" "wedding dance" as "wedding", "non-music performance", "music performance" as "show", and "baseball", "basketball", "biking", "ice skating", "skiing", "soccer", "swimming" as "sports". Finally, there are 2440 videos from five event classes (*i.e.*, "birthday", "parade", "show", "sports", and "wedding"). Since different datasets have different numbers of event classes, we use the 6 (*resp.*, 5) overlapped event classes between Flickr and Kodak (*resp.*, CCV) for performance evaluation.

We extract both textual features and improved dense trajectory features [155] from the training web videos. The textual features are used as privileged information.

- **Textual feature:** A 2000-dim term-frequency (TF) feature is extracted for each video by using the top-2000 words with the highest frequency as the vocabulary. Stop-word removal is performed to remove the meaningless words.

- **Visual feature:** We extract improved dense trajectory features using the source code provided in [155]. Specifically, three types of space-time (ST) features (*i.e.*, 96-dim Histogram of Oriented Gradient, 108-dim Histogram of Optical Flow and 192-dim Motion Boundary Histogram) are used, in which we set the trajectory length as 50, the sampling stride as 16, and all the other parameters as their default values. We construct the codebook by using k-means clustering on the ST features

from all videos in the training dataset to generate 2000 clusters, and then use the bag-of-words model for each type of ST features. Finally, each video is represented as a 6000-dim feature by concatenating the 2000-dim TF feature from each type of ST feature.

As the test data does not contain textual information, we only extract improved dense trajectory features for the videos in the test set, and each test video is also represented as a 6000-dim feature.

3.5.1.1 Experimental Results without Domain Adaptation

Baselines: We firstly compare our methods under the MIL-PI framework with two sets of baselines: the recent LUPI methods including pSVM+ [150] and Rank Transfer (RT) [135], as well as the conventional MIL method sMIL [17]. We also include SVM as a baseline, which is trained by using the visual features only. Moreover, we also compare our MIL methods with Classeme [146] and multi-view learning methods Kernel Canonical Correlation Analysis (KCCA) and SVM-2K, because they can also be used for our application.

- *KCCA* [65]: We apply KCCA on the training set by using the textual features and visual features, and then train the SVM classifier by using the common representations of visual features. In the testing process, the visual features of test videos are transformed into their common representations for the prediction.

- *SVM-2K* [49]: We train the SVM-2K classifiers by using the visual features and textual features from the training samples, and apply the visual feature based classifier on the test samples for the prediction.

- *Classeme* [146]: For each word in the 2000-dim textual features, we retrieve relevant and irrelevant videos to construct positive bags and negative bags, respectively. Then we follow [92] to use mi-SVM to train the classeme classifier for each word. For each training video and test video, 2000 decision values are obtained by using 2000 learnt classeme classifiers and the decision values are augmented with the visual features. Finally, we train the SVM classifiers for classifying the test videos based on the augmented features.

We also compare our MIL methods with MIML [179]. While we can treat the top 2000 words in the textual descriptions as noisy class labels, MIML cannot be directly applied to our task because the 2000 words may be different from the concept names. Thus, we use the decision values from the MIML classifiers as the features, similarly as in Classeme. Moreover, we additionally compare our MIL methods with the MILBoost method proposed in [91] which was used for video classification.

Experimental Settings: We train the classifiers by using the videos crawled from Flickr and evaluate the performances of different methods on Kodak and CCV datasets, respectively. Similarly as in [96], we uniformly partition the 300 relevant videos crawled from Flickr into positive bags, and also randomly partition the 300 irrelevant videos into negative bags. We obtain 60 positive bags and 60 negative bags by respectively using relevant videos and irrelevant videos, in which each training bag contains 5 instances. The positive ratio is set as $\sigma = 0.6$, as suggested in [96]. In our experiments, we use Gaussian kernel for visual features and linear kernel for textual features for our method and the baseline methods except RankTransfer (RT). The objective function of RT is solved in the primal form, so we can only use linear kernel instead of Gaussian kernel for visual features.

For performance evaluation, we report the Mean Average Precision (MAP) based on all test videos. For our method, we empirically fix $C_1 = 10^2, \gamma = 10^2$ (*resp.*, $C_1 = 10^{-2}, \gamma = 10$) for sMIL-PI (*resp.*, mi-SVM-PI, MIL-CPB-PI). For the baseline methods, we choose the optimal parameters based on their MAPs on the test dataset.

Experimental Results: The MAPs of all methods are reported in Table 3.1. By additionally exploiting textual information, pSVM+, Classme, MIML, KCCA, and SVM-2K are generally better than SVM. The RT method is worse than SVM due to the use of linear kernel for visual features. The MILBoost method is also much worse than SVM, although we have carefully tuned the parameters. It is worth mentioning that pSVM+ achieves better results than Classme, MIML, RT, MILBoost and the multi-view methods (*i.e.*, SVM-2K and KCCA) on both datasets, which demonstrates it is helpful to use textual features as privileged information.

Our MIL-PI methods generally achieve similar results. MIL-CPB-PI is the best when using Kodak as the test set. While mi-SVM-PI outperforms sMIL-PI and MIL-CPB-

Table 3.1: MAPs (%) of different methods without using domain adaptation. The results in boldface are from our methods.

Method	Test Set	
	Kodak	CCV
SVM	42.84	47.16
pSVM+	44.54	48.04
RT	36.22	34.16
Classeme	43.84	46.89
MIML	42.94	47.75
MILBoost	32.77	36.63
KCCA	44.46	47.91
SVM-2K	43.69	47.78
sMIL	42.94	47.90
sMIL-PI	**46.07**	**49.13**
mi-SVM	44.23	47.68
mi-SVM-PI	**45.89**	**49.32**
MIL-CPB	44.81	47.87
MIL-CPB-PI	**46.19**	**49.21**

PI when using CCV as the test set, MIL-CPB-PI also achieves comparable results as mi-SVM-PI.

Our MIL-PI methods (*i.e.*, sMIL-PI, mi-SVM-PI, MIL-CPB-PI) are better than pSVM+, RT, MIML, Classeme, and two existing multi-view learning methods, which demonstrates that it is beneficial to further cope with label noise of web videos as in our MIL-PI framework. Moreover, each of our MIL-PI methods also outperforms its corresponding conventional MIL method (*i.e.*, sMIL-PI *v.s* sMIL, mi-SVM-PI *v.s* mi-SVM, MIL-CPB-PI *v.s* MIL-CPB), which again demonstrates it is beneficial to exploit the additional textual features from web data as privileged information.

Table 3.2: MAPs (%) of SVM, sMIL-PI, mi-SVM-PI, MIL-CPB-PI and different domain adaptation methods. For SA, TCA, DIP, KMM, GFK and SGF, the first number is obtained by using the SVM classifier and the second number in the parenthesis is the best result obtained by using one of our MIL-PI methods. The results in boldface are from our domain adaptation methods.

Method	Test Set	
	Kodak	CCV
SVM	42.84	47.16
sMIL-PI	46.07	49.13
sMIL-PI-DA	**47.55**	**50.32**
mi-SVM-PI	45.89	49.32
mi-SVM-PI-DA	**47.59**	**50.75**
MIL-CPB-PI	46.19	49.21
MIL-CPB-PI-DA	**49.16**	**50.66**
DASVM	45.86	47.67
STM	44.93	49.00
SA	40.30(41.34)	47.21(49.47)
TCA	44.24(45.92)	48.91(49.10)
DIP	41.56(45.69)	44.49(46.28)
KMM	43.94(46.29)	48.97(49.03)
GFK	44.19(45.79)	45.93(48.84)
SGF	41.19(46.41)	47.71(48.69)

3.5.1.2 Experimental Results with Domain Adaptation

Baselines: We compare our methods sMIL-PI-DA, mi-SVM-PI-DA, and MIL-CPB-PI-DA in our MIL-PI-DA framework with the existing domain adaptation methods GFK [60], SGF [63], SA [53], TCA [119], KMM [71], DIP [7], DASVM [16] and STM [28]. We notice that the feature-based domain adaptation methods such as GFK, SGF, SA, TCA, DIP can be combined with the SVM classifier or our MIL-PI classifiers (*i.e.*, sMIL-PI, mi-SVM-PI, and MIL-CPB-PI), so we report two results for each domain adaptation

baseline method by using the SVM classifier and the best classifier from our MIL-PI framework.

Experiment Settings: We use the same setting as in Section 3.5.1.1. In our MIL-PI-DA framework, we have two more parameters (i.e., C_2 and λ) when compared with the MIL-PI framework. Recall that $\lambda = \frac{(C_2 m)^2}{\mu}$, where m is the number of source training samples and μ is the parameter used in dual form of MIL-PI-DA framework. We empirically fix $C_2 = 10$ (*resp.*, $C_2 = 10^{-5}$), $\lambda = 10^2$ for sMIL-PI-DA (*resp.*, mi-SVM-PI-DA, MIL-CPB-PI-DA). For the baseline methods, we choose the optimal parameters based on their best MAPs on the test dataset.

Experimental Results: The MAPs of all methods are reported in Table 3.2.

When using the SVM classifier, some existing feature-based domain adaptation methods (SA, DIP, and SGF on Kodak, and TCA, DIP, and GFK on CCV) are worse when compared with SVM. One possible explanation is that those two-step methods may not well preserve the discriminability of features when reducing the domain distribution mismatch in the first step. For these feature-based baselines, their results after using our MIL-PI framework are better when compared with those using the SVM classifier, which again shows the effectiveness of our MIL-PI framework for video event recognition by coping with label noise and simultaneously taking advantage of the additional textual features as privileged information. However, the results of the feature-based baselines after using our MIL-PI framework are still worse than our MIL-PI-DA methods. The experimental results clearly demonstrate our domain adaptation approaches are more effective than those two-step feature-based baseline methods.

Our framework is more related to KMM and STM. We also report two results for KMM because KMM can be combined with SVM or our sMIL-PI method. Particularly, the instance weights are learnt in the first step by using KMM and then we use the learnt instance weights to reweight the loss function of SVM or sMIL-PI in the second step. We observe that our sMIL-PI-DA method is better than STM and KMM when using the SVM or sMIL-PI classifier. One possible explanation is our sMIL-PI-DA method can achieve the global optimal solution by solving a convex optimization problem in one step while KMM is a two-step approach and STM can only achieve the local optimum.

We also observe that each of our methods under the domain adaptation framework MIL-PI-DA outperforms its corresponding version under the MIL-PI framework (*i.e.*,

sMIL-PI-DA *v.s* sMIL-PI, mi-SVM-PI-DA *v.s* mi-SVM-PI, MIL-CPB-PI-DA *v.s* MIL-CPB-PI), which shows it is helpful to reduce the domain distribution mismatch by introducing the MMD based regularizer. Moreover, our MIL-PI-DA methods also outperform all the existing domain adaptation baselines, which demonstrates the effectiveness of our MIL-PI-DA framework.

Finally, our newly proposed instance-level MIL-PI-DA methods (*i.e.*, mi-SVM-PI-DA and MIL-CPB-PI-DA) achieve better results than the bag-level MIL-PI-DA method sMIL-PI-DA on both test sets, which shows it is useful to infer the instance labels in the positive bags on both datasets.

3.5.2 Human Action Recognition

Experimental Settings: In this section, we evaluate our MIL-PI and MIL-PI-DA framework for human action recognition on the benchmark dataset HMDB51 [85].

We collect a new training dataset for human action recognition by crawling short videos and their surrounding textual descriptions from YouTube website using 51 action names from the HMDB51 dataset as the queries. We use the top 200 web videos for each query as the relevant videos and randomly sample the same number of web videos that do not contain the query as one of the surrounding texts as the irrelevant videos. Then, for each action class, we construct 40 training bags, in which the size of each training bag is 5. The HMDB51 dataset contains 6766 clips from 51 action classes. As suggested in [85], we use 3 testing splits as the test set, in which each split contains 30 videos for each action class.

For the YouTube dataset, we extract both the textual features and the visual features for each video. For the textual features, we extract the same 2000-dim term frequency (TF) features from the surrounding textual descriptions as in Section 3.5.1. For the visual features, we follow [155] by utilizing Fisher vector encoding, which has shown excellent performance for human action recognition. Specifically, we adopt the improved dense trajectory features and extract four types of descriptors (*i.e.*, 30-dim trajectory, 96-dim Histogram of Oriented Gradient, 108-dim Histogram of Optical Flow, and 192-dim Motion Boundary Histogram). Then, we generate the Fisher vector features by using

256 Gaussian Mixture Models (GMMs) for each type of descriptors, and then use PCA to reduce the dimension of the concatenated Fisher vector to 10000. As the HMDB51 dataset does not contain textual descriptions, we only extract the visual features for each video in the HMDB51 dataset.

As suggested in [85], we evaluate the baseline methods and our methods on 3 testing splits, and report the mean accuracy over 3 splits for performance evaluation. For our MIL-PI methods and MIL-PI-DA methods, we use the same parameters as in Section 3.5.1. For the baseline methods, we choose the optimal parameters based on their mean accuracies on the test dataset. The other experimental settings are the same as in Section 3.5.1.

Experimental Results: The accuracies of all methods are reported in Table 3.3. From the left subtable, we observe that multi-instance learning methods sMIL, mi-SVM and MIL-CPB outperform SVM, which indicates the effectiveness of multi-instance learning methods for coping with label noise. By additionally taking advantage of textual information, pSVM+, RT, Classme, MIML, KCCA, and SVM-2K are better than SVM, and each of our MIL-PI methods is also better than its corresponding conventional MIL method (*i.e.*, sMIL-PI *v.s* sMIL, mi-SVM-PI *v.s* mi-SVM, or MIL-CPB-PI *v.s* MIL-CPB) respectively.

From the left subtable, we also observe that our MIL-PI methods (*i.e.*, sMIL-PI, mi-SVM-PI, MIL-CPB-PI) are better than the baseline methods (*i.e.*, pSVM+, RT, MIML, Classeme, and multi-view learning methods), which can additionally utilize the textual features. A possible explanation is that we additionally cope with label noise of web videos by utilizing the multi-instance learning techniques.

From the right subtable, we observe that the existing domain adaptation methods DASVM, SA, DIP, KMM, GFK, and SGF are better than SVM by utilizing the unlabeled target domain samples to reduce the domain distribution mismatch. It is interesting that STM and TCA are worse than SVM, although we have carefully tuned their parameters. We also observe that our sMIL-PI-DA (*resp.*, mi-SVM-PI-DA, MIL-CPB-PI-DA) outperforms sMIL-PI (*resp.*, mi-SVM-PI, MIL-CPB-PI), which shows it is beneficial to reduce the domain distribution mismatch by using our domain adaptation approach. Moreover, our MIL-PI-DA methods also outperform all the existing domain adaptation baselines.

Table 3.3: The left subtable shows the accuracies (%) of different methods on the HMDB51 dataset without considering the domain distribution mismatch. The right subtable shows the accuracies (%) of SVM, our MIL-PI methods, and different domain adaptation methods on the HMDB51 dataset. In the right subtable, for SA, TCA, DIP, KMM, GFK and SGF, the first number is obtained by using the SVM classifier and the second number in the parenthesis is the best result obtained by using one of our MIL-PI methods. The results in boldface are from our methods

Method	Accuracy
SVM	50.94
pSVM+	52.64
RT	51.42
Classeme	51.63
MIML	51.76
KCCA	51.24
SVM-2K	51.91
sMIL	51.96
sMIL-PI	**53.62**
mi-SVM	52.11
mi-SVM-PI	**53.22**
MIL-CPB	53.62
MIL-CPB-PI	**55.38**

Method	Accuracy
SVM	50.94
sMIL-PI	53.62
sMIL-PI-DA	**55.45**
mi-SVM-PI	53.22
mi-SVM-PI-DA	**57.65**
MIL-CPB-PI	55.38
MIL-CPB-PI-DA	**57.31**
DASVM	51.98
STM	37.43
SA	53.16(55.58)
TCA	43.12(46.95)
DIP	51.20(55.73)
KMM	53.51(53.77)
GFK	52.90(54.27)
SGF	51.31(52.77)

In order to further evaluate our domain adaptation approaches, we combine the feature-based domain adaptation methods (*i.e.*, SA, TCA, DIP, GFK, and SGF) with our MIL-PI methods (sMIL-PI, mi-SVM-PI, and MIL-CPB-PI) and combine KMM with our sMIL-PI method, similarly as discussed in Section 3.5.1. For each feature-based domain adaptation method, we report the best result obtained by using one of our three MIL-PI methods. The feature-based domain adaptation methods and KMM after using the best classifier learnt from one of our three MIL-PI methods (*i.e.*, sMIL-PI, mi-SVM-PI, or MIL-CPB-PI) achieve better results, because our MIL-PI methods can help handle label

noise and simultaneously utilize privileged information.

Our instance-level methods mi-SVM-PI-DA and MIL-CPB-PI-DA outperform the feature-based domain adaptation methods combined with our MIL-PI methods. For SA and DIP, the results in the parenthesis are slightly better than our sMIL-PI-DA (see the right subtable in Table 3.3). However, SA and DIP are both combined with our MIL-CPB-PI method. When SA and DIP are combined with our sMIL-PI method, the result of SA and DIP are 54.01% and 53.94%, respectively, which are still worse than our sMIL-PI-DA method.

3.5.3 How to Utilize Privileged Information

As discussed in Section 3.3, in our MIL-PI framework, we use privileged information for relevant videos (*i.e.*, positive bags) only, because privileged information (*i.e.*, textual features) may not be always reliable. To verify it, we evaluate SVM+ by utilizing privileged information for all training samples. The MAPs of SVM+ are 44.08% and 47.49% when using Kodak and CCV as the test sets, respectively, which are worse than pSVM+ on those two datasets (44.54% and 48.04% reported in Table 3.1).

Similarly, we also evaluate our MIL-PI methods under two settings (*i.e.*, full privileged information (PI) and partial privileged information (PI)). We report the results of our MIL-PI methods under two settings on two datasets in Table 3.4. We observe that the MAPs of our MIL-PI methods under the full PI setting are lower than their corresponding results under the partial PI setting on both datasets. These results verify our conjecture that privileged information of irrelevant web videos may not be helpful for learning robust classifiers, because the labels of irrelevant videos are generally correct while the textual features are not always reliable.

Since our MIL-PI methods with partial PI achieve better results than those with full PI, we further conjecture it may be useful to additionally learn the importance of privileged information of training samples during the training process. However, it is a non-trivial task under our setting where the labels of training samples are noisy. So we leave how to learn the importance of privileged information as our future work.

Table 3.4: MAPs (%) of our MIL-PI methods when using partial privileged information (PI) and full PI.

Method	partial PI		full PI	
	Kodak	CCV	Kodak	CCV
sMIL-PI	**46.07**	**49.13**	45.58	48.55
mi-SVM-PI	**45.89**	**49.32**	45.41	48.38
MIL-CPB-PI	**46.19**	**49.21**	45.51	48.04

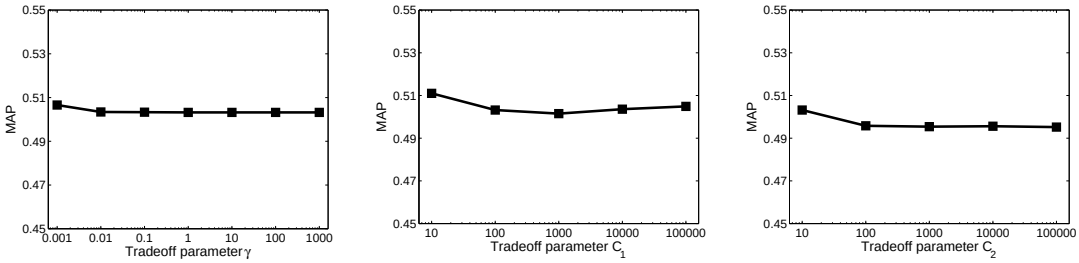


Figure 3.2: MAPs of sMIL-PI-DA on the CCV dataset when using different trade-off parameters.

3.5.4 Robustness to the Parameters

Our methods are relatively robust when the trade-off parameters are set in certain ranges. Here, we study the performance variation of our sMIL-PI-DA method with respect to one parameter while fixing other parameters as their default values. Let us take the CCV dataset as an example, the MAPs of sMIL-PI-DA are in the range of $[50.32\%, 50.66\%]$ (*resp.*, [50.15%, 51.10%]) when we set $\gamma \in [10^{-3}, 10^{3}]$ (*resp.*, $C_1 \in [10^{1}, 10^{5}]$), as shown in the left (*resp.*, middle) subfigure in Fig 3.2. For the parameters C_2 and λ, we observe our methods are relatively robust when $\frac{C_2}{\lambda}$ is empirically fixed as 10^4. The MAPs of sMIL-PI-DA are in the range of $[49.52\%, 50.32\%]$ when we set $C_2 \in [10^{1}, 10^{5}]$, as shown in the right subfigure in Fig 3.2. We also have similar observations for our other methods and on other datasets. We will study how to decide the optimal parameters in our future work.

Table 3.5: Training time of our sMIL-PI method and the baseline methods without domain adaptation on the CCV dataset.

Method	SVM	pSVM+	RT	Classeme	MIML	KCCA	SVM-2K	sMIL	sMIL-PI
Time(s)	22.17	35.21	1501.51	1618.15	8785.27	88.13	96.98	18.31	21.86

Table 3.6: Training time of our sMIL-PI-DA method and the existing domain adaptation methods on the CCV dataset.

Method	DASVM	STM	SA	TCA	DIP	KMM	GFK	SGF	sMIL-PI-DA
Time(s)	1130.05	204.74	615.79	972.95	1089.95	111.23	1932.82	3592.87	151.71

3.5.5 Comparison of Training Time

In this section, we take sMIL-PI and sMIL-PI-DA as two examples to compare the training time with the corresponding MIL method sMIL as well as other baselines. As shown in (3.6), our sMIL-PI method can be formulated as a quadratic programming (QP) problem with respect to two variables $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$. Compared with sMIL, which can be formulated as a QP problem with respect to one variable $\boldsymbol{\alpha}$ only, the size of the QP problem in (3.6) is larger. However, it can still be efficiently solved with the existing QP solvers. Specifically, we take the CCV dataset as an example to compare the training time of sMIL-PI with other baseline methods. From Table 3.5, we observe that the training time of sMIL-PI is only slightly longer than sMIL, and our sMIL-PI method is much more efficient than other baseline methods.

Similarly, our sMIL-PI-DA method can also be solved as a QP problem w.r.t. three variables $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}\}$. So it can also be efficiently solved by using the existing QP solvers. In Table 3.6, we take the CCV dataset as an example to compare the training time of sMIL-PI-DA with existing methods. We observe that our sMIL-PI-DA method is faster than other baseline methods except KMM. A possible explanation is that KMM solves a smaller scale QP problem *w.r.t.* $\boldsymbol{\theta}$ before training an SVM classifier in the second step.

3.6 Summary

In this chapter, we have proposed new MIL approaches for action and event recognition by learning from loosely labeled web data. We firstly propose a new MIL-PI framework together with three instantiations sMIL-PI, mi-SVM-PI and MIL-CPB-PI, in which we not only take advantage of the additional textual features in the training web videos but also effectively cope with noise in the loose labels of relevant training web videos. We further propose a new MIL-PI-DA framework and three instantiations sMIL-PI-DA, mi-SVM-PI-DA and MIL-CPB-PI-DA, which can additionally reduce the data distribution mismatch between the training and test videos. By using freely available web videos as training data, our approaches are inherently not limited by any predefined lexicon. Extensive experiments clearly demonstrate our proposed approaches are effective for action and event recognition.

Chapter 4

Visual Recognition by Learning from Web Data via Weakly Supervised Domain Generalization

In this chapter, a weakly supervised domain generalization method is proposed for real-world visual recognition tasks, in which we train classifiers by using web data (*e.g.*, web images and web videos) with noisy labels. In particular, two challenging problems need to be solved when learning robust classifiers, in which the first issue is to cope with the label noise of training web data from the source domain while the second issue is to enhance the generalization capability of learnt classifiers to arbitrary target domain. In order to handle the first problem, the training samples within each category are partitioned into clusters, where we use "bag" to denote each cluster and "instances" to denote the samples in each cluster. Then, we identify a proportion of good training samples in each bag and train robust classifiers by using the good training samples, which leads to a multi-instance learning (MIL) problem. In order to handle the second problem, we assume that the training samples possibly form a set of hidden domains, with each hidden domain associated with a distinctive data distribution. Then, for each category and each hidden latent domain, we propose to learn one classifier by extending our MIL formulation, which leads to our Weakly Supervised Domain Generalization (WSDG) approach. In the testing stage, our approach can obtain better generalization capability by effectively integrating multiple classifiers from different latent domains in each category. Moreover, our WSDG approach is further extended to utilize additional textual descriptions associated with web data as privileged information although test data do not have such privileged information. Extensive experiments on three benchmark datasets indicate that our newly proposed methods are effective for real-world visual recognition tasks by learning from web data.

4.1 Introduction

The research interest on utilizing web images/videos as the training data to recognize new images/videos grows rapidly in recent years. Nevertheless, as mentioned in [144], the data distributions of training and test samples are most likely to be different, which leads to the dataset bias problem [144]. In order to tackle this issue, researchers have proposed abundant domain adaptation approaches for different computer vision tasks [9, 22, 25, 40, 42, 44, 45, 63, 98, 120]. In the case that target domain data are unseen in the training stage, the problem is called domain generalization. Compared with domain adaptation, domain generalization targets at learning robust classifiers that have excellent generalization ability to arbitrary target domain [80, 110, 116, 164], which is very important in real-world visual recognition tasks. For instance, different datasets consisting of photos/videos captured by different users with different cameras can be treated as different target domains which have different visual feature distributions. Due to privacy issues, some users may be reluctant to upload their photos/videos to public websites and thus we are lacking of data from some target domains. In such case, it is crucial to develop effective approaches for domain generalization, which do not require target domain data during the training stage.

In this chapter, the domain generalization problem is explored by utilizing freely available source domain data (*i.e.*, web images/videos). Specifically, a novel method called weakly supervised domain generalization (WSDG) is developed in Section 4.3. Two important issues are considered: 1) web images/videos are often associated with inaccurate labels, *i.e.*, they are loosely labeled; 2) the data distributions between the source domain and the target domain are usually quite different. Moreover, during the training stage, the target domain data is generally unseen.

To tackle the inaccurate labels of training images/videos, the training samples within each category are first partitioned into clusters. We use "bag" to denote each cluster and "instances" to denote the samples in each cluster. We only have the labels of each training bags, but the instance labels in each training bag are unknown. Inspired by multi-instance learning (MIL) works, we use a proportion of good samples selected from a bag to representing the bag, assuming that the training bags from different categories

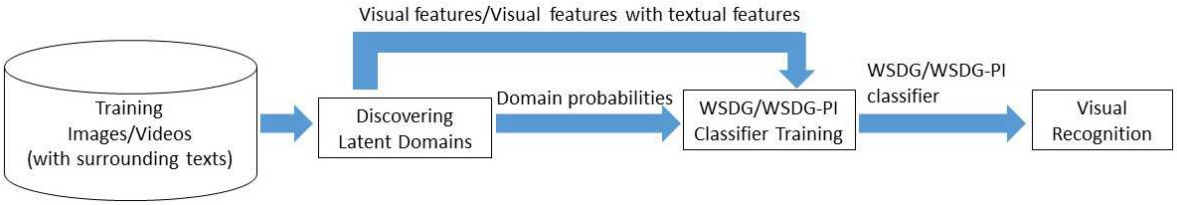


Figure 4.1: The flowchart of our visual recognition methods. The flowchart consists of an approach to discover the latent domains, which learns the probabilities that each training sample comes from each latent domain, and a classification method WSDG/WSDG-PI, which learns one classifier for each category and each hidden latent domain. For our WSDG method, only the visual features are required as the input, while for our WSDG-PI method, visual features together with textual features are required as the input.

can be well distinguished. We then unify learning robust classifiers and selecting good training samples for each bag in a multi-class multi-instance formulation.

On the other hand, inspired by the recent works [59, 69, 164], we conjecture the training web samples possibly form a set of hidden latent domains, each of which has a different data distribution. Thus, we apply the existing technology to discover multiple latent domains, and then learn one classifier for each category and each latent domain. Since the training samples for learning classifiers for each category and each latent domain are relatively more coherent, the integrated classifier obtained by fusing multiple classifiers from all categories is more robust to various data distributions. As a result, we expect the integrated classifier will have good generalization ability to arbitrary target domain. Note that for each training bag, we just use a proportion of training samples to learn the classifiers, and thus we propose to identify the training instances that have more distinctive data distributions by using a Maximum Mean Discrepancy (MMD) based regularizer.

In the testing stage, for each test sample, we select the classifier corresponding to the highest response among all classifiers from different latent domains in each category, which can be intuitively explained as we select the most matched latent domain for each test sample. As a result, the data distribution mismatch between training samples and test samples can be reduced.

Additionally, the web data are usually accompanied by additional textual information (*e.g.*, tags, descriptions, and captions), which can be used as privileged information [98, 150], though these textual features are not available for the test data. In Section 4.4, our WSDG method is extended by utilizing such privileged information, which is referred to as WSDG-PI. The flowchart of our WSDG and WSDG-PI methods is shown in Fig. 4.1. In Section 4.5, the extensive experimental results clearly show the effectiveness of our approaches.

4.2 Related Work

Multi-instance learning (MIL) is in the sense that we partition the training samples into clusters and use "bag" (*resp.*, "instances") to denote each cluster (*resp.*, the samples in each bag). A set of MIL approaches were developed in [4, 96, 99]. In mi-SVM [4], the SVM classifier is trained at each iteration based on the inferred instance labels from the previous iteration. In KI-SVM [99], the key instances inside each bag are used as the representatives of the bag. Nevertheless, these methods were proposed without taking the data distribution mismatch between two domains into consideration, so the learnt classifiers may not generalize well to arbitrary target domain.

Domain generalization is another relevant research topic. When we have target domain data in the training process, domain adaptation approaches can be used to reduce the domain distribution mismatch. The recent works on domain generalization and domain adaptation have been discussed in Section 2.4 and Section 2.3.

This chapter is also related to several recent approaches which can discover latent domains [59, 69, 160, 164]. In these works, latent domains are discovered based on a clustering approach (*i.e.*, [69]), the MMD criterion (*i.e.*, [59]), or mutual information (*i.e.*, [160]). After discovering the latent domains, these works train an SVM classifier or K-Nearest Neighbors (KNN) classifier for each hidden latent domain, and then all the classifiers learnt for different latent domains are integrated to predict the test samples. Unlike the above works, we jointly learn multiple classifiers for all latent domains and categories, which can be effectively integrated in a unified formulation.

The sub-categorization problem [67] is also related to this chapter since each category often consists of multiple subcategories. Recently, some works were proposed to integrate

multi-instance learning (MIL) with sub-categorization [158, 175, 176]. Nevertheless, the domain distribution mismatch between the training and the test data was not considered in these works, which is quite different from the domain generalization problem discussed in this chapter.

Finally, learning using privileged information (LUPI) [150] is also related to this chapter. In the LUPI paradigm, training samples are associated with additional features that are not available for the test data, which are referred to as privileged information. In some recent works [55, 98, 135, 163], privileged information was exploited for different computer vision tasks. In [135], rank SVM was proposed to rank web images based on privileged information. In [55, 163], privileged information is incorporated into distance metric learning. However, these works assume training data and test data are with the same data distribution while this assumption does not hold in our setting. In [98], a new method was proposed to simultaneously handle label noise, take advantage of privileged information, and reduce the domain distribution mismatch. However, the target domain data are required in [98], while they are assumed to be unseen in this chapter.

4.3 Weakly Supervised Domain Generalization

In this section, a novel weakly supervised domain generalization (WSDG) approach is proposed, which simultaneously identifies good samples and learns robust classifiers. Assuming there are N training samples from C categories in the source domain, the source domain data are denoted as $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i$ is the i-th training sample, with its corresponding category label $y_i \in \{1, \ldots, C\}$.

Next, we first provide a brief introduction on how to discover latent domains using the existing technology [59]. Then, we develop a multi-class multi-instance learning approach without considering the latent domain issues. Last, we integrate the latent domain discovery technique into our multi-class multi-instance learning formulation.

4.3.1 Discovering Latent Domains

In this chapter, the existing latent domain discovering technique in [59] is adopted, which relies on the Maximum Mean Discrepancy (MMD) criterion. We use $\pi_{i,m} \in \{0, 1\}$ to

indicate whether each sample belongs to each latent domain. Specifically, $\pi_{i,m} = 1$ if $\mathbf{x}_i$ comes from the m-th latent domain, and $\pi_{i,m} = 0$ otherwise. We denote $N_m = \sum_{i=1}^{N} \pi_{i,m}$ as the number of training samples from the m-th hidden latent domain. The approach in [59] aims to maximize the sum of MMDs between each pair of latent domains, expecting the discovered latent domains to be as distinctive as possible, *i.e.*,

$$\max_{\pi_{i,m}} \sum_{m \neq \tilde{m}} \| \frac{1}{N_m} \sum_{i=1}^{N} \pi_{i,m} \phi(\mathbf{x}_i) - \frac{1}{N_{\tilde{m}}} \sum_{i=1}^{N} \pi_{i,\tilde{m}} \phi(\mathbf{x}_i) \|^2, \tag{4.1}$$

where $\phi(\cdot)$ is the feature mapping function which is induced by a kernel $\mathbf{K} \in \mathbb{R}^{N \times N}$ on the training data (*i.e.*, $\mathbf{K} = [K_{i,j}]$ with $K_{i,j} = \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)$). Let $\beta_{i,m} = \frac{\pi_{i,m}}{N_m}$ and $\boldsymbol{\beta}_m = [\beta_{1,m}, \ldots, \beta_{N,m}]'$, we can relax the above problem according to [59] as,

$$\max_{\boldsymbol{\beta}} \quad \sum_{m \neq \tilde{m}} (\boldsymbol{\beta}_m - \boldsymbol{\beta}_{\tilde{m}})' \mathbf{K} (\boldsymbol{\beta}_m - \boldsymbol{\beta}_{\tilde{m}}) \tag{4.2}$$

$$\text{s.t.} \quad \frac{1}{N} \leq \sum\nolimits_{m=1}^{M} \beta_{i,m} \leq \frac{1}{C}, \quad \forall i, \tag{4.3}$$

$$\sum\nolimits_{i=1}^{N} \delta(y_i = c) \beta_{i,m} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \delta(y_i = c), \forall c, m, \tag{4.4}$$

$$\sum\nolimits_{i=1}^{N} \beta_{i,m} = 1, \beta_{i,m} \geq 0, \quad \forall i, m, \tag{4.5}$$

where the first constraint in (4.3) is to guarantee that at least one training sample is selected in each hidden latent domain per category, the second constraint in (4.4) is to ensure the class distribution in the whole source domain is consistent with that in each hidden latent domain, and the third constraint in (4.5) can be easily obtained based on the definitions of $\beta_{i,m}$ and $\pi_{i,m}$. Interested readers can refer to [59] for more technical details. Note the above quadratic programming problem is non-convex, which is not easy to be optimized. However, we can still utilize the existing solver in [1] to achieve satisfactory performance.

After latent domains are discovered by optimizing the objective function in (4.2), one classifier is learnt for each category and each hidden latent domain. Then, a set of classifiers for each category are integrated based on the learnt $\beta_{i,m}$'s. Next, we develop a novel multi-class multi-instance learning formulation to cope with the label noise, followed by extending our proposed formulation with the learnt $\beta_{i,m}$'s to make the learnt classifiers more capable of generalizing to arbitrary target domain.

4.3.2 Formulation

4.3.2.1 Learning with Weakly Supervised Information

In multi-instance learning (MIL), training samples are partitioned into a set of bags with explicit bag labels while the accurate labels of training instances in each bag are unknown. Inspired by MIL, the training samples within each category in our case are partitioned into training bags, *i.e.*, $\{(\mathcal{B}_l, Y_l)|l = 1, \ldots, L\}$. As the training samples are obtained by using category names as searching queries, the bag label $Y_l \in \{1, \ldots, C\}$ is the corresponding query name. Similarly to [96], each positive bag is assumed to have at least a certain portion of true positive instances. Thus, we use the ratio η to denote the proportion of true positive training instances in each bag. Note η can be estimated from some prior knowledge, similar to conventional MIL methods.

In order to learn robust classifiers effectively, we want to select good samples from each training bag by removing the outliers with inaccurate class labels. Particularly, we use a binary indicator $h_i \in \{0, 1\}$ to indicate whether each training sample $\mathbf{x}_i$ is selected. To be exact, $h_i = 0$ if $\mathbf{x}_i$ is not selected, and $h_i = 1$, otherwise. We define $\mathbf{h} = [h_1, \ldots, h_N]'$ as the indicator vector, and use $\mathcal{H} = \{\mathbf{h}|\sum_{i\in I_l} h_i = \eta|\mathcal{B}_l|, \forall l\}$ to represent the feasible set of $\mathbf{h}$, where I_l represents the set of instance indices in $\mathcal{B}_l$, and $|\mathcal{B}_l|$ denotes the cardinality of $\mathcal{B}_l$.

Based on the multi-class SVM [30], we propose our multi-class MIL formulation as follows. In particular, C classifiers $\{f_c(\mathbf{x})|c = 1, \ldots C\}$ are to be learnt, where each classifier[1] can be represented as $f_c(\mathbf{x}) = (\mathbf{w}_c)'\phi(\mathbf{x})$. Inspired by the MIL learning method KI-SVM [99] as well as multi-class SVM [30], we propose to jointly learn $\mathbf{h}$ and C classifiers as,

$$\min_{\substack{\mathbf{h}\in\mathcal{H}\\ \mathbf{w}_c,\xi_l}} \quad \frac{1}{2}\sum_{c=1}^{C}\|\mathbf{w}_c\|^2 + C_1\sum_{l=1}^{L}\xi_l \tag{4.6}$$

$$\text{s.t.} \quad \frac{1}{|\mathcal{B}_l|}\sum_{i\in I_l} h_i\left((\mathbf{w}_{Y_l})'\phi(\mathbf{x}_i) - (\mathbf{w}_{\tilde{c}})'\phi(\mathbf{x}_i)\right) \geq \eta - \xi_l, \quad \forall l, \tilde{c} \neq Y_l, \tag{4.7}$$

$$\xi_l \geq 0, \qquad \forall l, \tag{4.8}$$

[1]We omit the bias term here for better representation. Instead, the feature of each training sample is augmented with an extra element of 1.

where C_1 is a tradeoff parameter, and ξ_l's are slack variables. We enforce the total decision value of each bag $\mathcal{B}_l$ obtained based on the classifier corresponding to its category to be larger than those obtained by using the classifiers for the other categories by using the constraint in (4.7). Intuitively, we expect to identify good instances within each training bag to reduce the bag-level loss.

Note that multi-class SVM [30] is a special case of the problem in (4.6) with the bag size $|\mathcal{B}_l|$ being 1. Besides, when there are two categories, (4.6) becomes the MIL learning problem in KI-SVM [99] with slight modifications.

4.3.2.2 Weakly Supervised Domain Generalization

Now considering the training samples in the source domain come from M latent domains, we propose to enhance the generalization capability of the learnt classifiers by integrating the classifiers from all latent domains for each category.

To be exact, totally $C \times M$ classifiers $\{f_{c,m}(\mathbf{x})|c = 1, \dots C, \text{ and } m = 1, \dots, M\}$ are to be learnt, where $f_{c,m}(\mathbf{x}) = (\mathbf{w}_{c,m})'\phi(\mathbf{x})$ represents the classifier corresponding to the m-th hidden latent domain and the c-th category. Then, we can obtain the decision function on $\mathbf{x}_i$ for each category by integrating the learnt classifiers from multiple latent domains as $f_c(\mathbf{x}_i) = \sum_{m=1}^{M} \hat{\beta}_{i,m} f_{c,m}(\mathbf{x}_i)$, where $\hat{\beta}_{i,m}$ is the probability that the i-th training sample comes from the m-th hidden latent domain. $\hat{\beta}_{i,m}$ is defined as $\hat{\beta}_{i,m} = \frac{\beta_{i,m}}{\sum_{m=1}^{M} \beta_{i,m}}$, where $\beta_{i,m}$'s are precomputed by solving (4.2). In summary, we expect to learn $C \times M$ classifiers to make the integrated classifiers $f_c(\mathbf{x}_i)$'s as discriminative as possible.

Note that the latent domain discovery technique in [59] was proposed for clean training data without label noise. When dealing with the training data with noisy labels, while maximizing (4.2), we need to seek for an optimal $\mathbf{h}$ to remove outliers. With $\mathbf{B} = [\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_M] \in \mathbb{R}^{N \times M}$, the objective function in (4.2) can be written as $\rho(\mathbf{B}, \mathbf{K}) = \sum_{m \neq \tilde{m}} (\boldsymbol{\beta}_m - \boldsymbol{\beta}_{\tilde{m}})' \mathbf{K} (\boldsymbol{\beta}_m - \boldsymbol{\beta}_{\tilde{m}})$. In order to learn an optimal $\mathbf{h}$, we add a regularizer $\rho(\mathbf{B}, \mathbf{K} \circ (\mathbf{h}\mathbf{h}'))$ and derive the complete objective function of our proposed

WSDG approach as,

$$\min_{\substack{\mathbf{h}\in\mathcal{H}\\ \mathbf{w}_{c,m},\xi_l}} \quad \frac{1}{2}\sum_{c=1}^{C}\sum_{m=1}^{M}\|\mathbf{w}_{c,m}\|^2 + C_1\sum_{l=1}^{L}\xi_l - C_2\,\rho(\mathbf{B},\mathbf{K}\circ(\mathbf{h}\mathbf{h}')) \tag{4.9}$$

$$\text{s.t.} \quad \frac{1}{|\mathcal{B}_l|}\sum_{i\in I_l} h_i\left(\sum_{m=1}^{M}\hat{\beta}_{i,m}(\mathbf{w}_{Y_l,m})'\phi(\mathbf{x}_i) - (\mathbf{w}_{\tilde{c},\tilde{m}})'\phi(\mathbf{x}_i)\right)$$

$$\geq \eta - \xi_l, \qquad \forall l,\tilde{m},\tilde{c}\neq Y_l, \tag{4.10}$$

$$\xi_l \geq 0, \qquad \forall l, \tag{4.11}$$

where C_2 is a tradeoff parameter. The explanation for the constraint (4.10) is similar to that for (4.7) except that we replace $(\mathbf{w}_{Y_l})'\phi(\mathbf{x}_i)$ in (4.7) with $\sum_{m=1}^{M}\hat{\beta}_{i,m}(\mathbf{w}_{Y_l,m})'\phi(\mathbf{x}_i)$ and $(\mathbf{w}_{\tilde{c}})'\phi(\mathbf{x}_i)$ with $(\mathbf{w}_{\tilde{c},\tilde{m}})'\phi(\mathbf{x}_i)$.

Essentially, we train one classifier for each category and each hidden latent domain. This is mainly because the data distributions of the training samples from one category and one hidden latent domain are generally more similar [59], which makes learning a discriminative classifier easier. In the testing stage, given a test sample $\mathbf{x}$, we predict its label by,

$$\arg\max_{c}\left(\max_{m}\mathbf{w}_{c,m}{}'\phi(\mathbf{x})\right). \tag{4.12}$$

Namely, for each category, we attempt to seek for the most matched hidden latent domain for a given test sample, whose classifier achieves the largest decision value from all the latent domains. In this way, we conjecture the integrated classifiers have good generalization ability to the test data from arbitrary target domain.

4.3.3 Optimization

The non-convex mixed integer problem in (4.9) is nontrivial to solve. According to some recent works on MIL [96][99], the dual form of (4.9) can be relaxed as a multiple kernel learning (MKL) problem, which shares a similar solution as that in [82]. Next, we introduce how to relax the dual form of (4.9), and then discuss how to solve the relaxed problem in detail.

4.3.3.1 Reformulation in Dual Form

Proposition 1 *The dual form of (4.9) is,*

$$\begin{aligned} \min_{\mathbf{h}\in\mathcal{H}} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}^{\mathbf{h}}\boldsymbol{\alpha} + \boldsymbol{\zeta}'\boldsymbol{\alpha} - C_2\,\rho(\mathbf{B}, \mathbf{K}\circ(\mathbf{h}\mathbf{h}')) \\ s.t. \quad & \sum_{c,m}\alpha_{l,c,m} = C_1, \quad \forall l \\ & \alpha_{l,c,m} \geq 0, \quad \forall l, c, m, \end{aligned} \tag{4.13}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{\tilde{D}}$ *is a vector containing dual variables* $\alpha_{l,c,m}$, $\tilde{D} = L\cdot C\cdot M$, $\boldsymbol{\zeta} \in \mathbb{R}^{\tilde{D}}$ *is a vector, in which each entry* $\zeta_{l,c,m} = 0$ *if* $c = Y_l$ *and* $\zeta_{l,c,m} = \eta$ *otherwise. Each element in the matrix* $\mathbf{Q}^{\mathbf{h}} \in \mathbb{R}^{\tilde{D}\times\tilde{D}}$ *can be obtained based on* $Q^{\mathbf{h}}_{u,v} = \frac{1}{|\mathcal{B}_l||\mathcal{B}_{\tilde{l}}|}\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{I}_{\tilde{l}}} h_i h_j \phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)\gamma(i,j,c,\tilde{c},m,\tilde{m})$ $v = (\tilde{l}-1)\cdot C\cdot M + (\tilde{c}-1)\cdot M + \tilde{m}$ *and* $u = (l-1)\cdot C\cdot M + (c-1)\cdot M + m$ *are the indices, and* $\gamma(i,j,c,\tilde{c},m,\tilde{m}) = [1-\delta(c=y_i)][1-\delta(\tilde{c}=y_j)][\delta(y_i=y_j)\sum_{q=1}^{M}\hat{\beta}_{i,q}\hat{\beta}_{j,q} + \delta(c=\tilde{c})\delta(m=\tilde{m})] - [1-\delta(c=\tilde{c})]\{[1-\delta(c=y_i)]\delta(c=y_j)\hat{\beta}_{j,m} + [1-\delta(\tilde{c}=y_j)]\delta(\tilde{c}=y_i)\hat{\beta}_{i,\tilde{m}}\}$. *The proof of Proposition 1 can be found in Appendix B.*

The problem in (4.13) is a mixed integer programming problem, which is difficult to be solved. Inspired by [96][99], we use an alternative approach to find the optimal combination coefficients of $\mathbf{h}_t\mathbf{h}_t'$'s given all feasible $\mathbf{h}_t \in \mathcal{H}$, *i.e.*, $\sum_{\mathbf{h}_t\in\mathcal{H}} d_t\mathbf{h}_t\mathbf{h}_t'$ with d_t being the combination coefficient instead of directly optimizing over the indicator vector $\mathbf{h}$. For ease of presentation, we denote $T = |\mathcal{H}|$, $\mathbf{d} = [d_1,\ldots,d_T]'$, the feasible set of $\mathbf{d}$ as $\mathcal{D} = \{\mathbf{d}|\mathbf{d}'\mathbf{1} = 1, \mathbf{d}\geq 0\}$, and the feasible set of $\boldsymbol{\alpha}$ in (4.13) as $\mathcal{A}$. Then, we arrive at the following optimization problem:

$$\begin{aligned} \min_{\mathbf{d}\in\mathcal{D}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad & -\frac{1}{2}\sum_{t=1}^{T} d_t\boldsymbol{\alpha}'\mathbf{Q}^{\mathbf{h}_t}\boldsymbol{\alpha} + \boldsymbol{\zeta}'\boldsymbol{\alpha} \\ & -C_2\sum_{t=1}^{T} d_t\rho(\mathbf{B}, \mathbf{K}\circ(\mathbf{h}_t\mathbf{h}_t')), \end{aligned} \tag{4.14}$$

Note that we move the sum operator over d_t outside $\mathbf{Q}^{\mathbf{h}_t}$ and $\rho(\mathbf{B}, \mathbf{K}\circ(\mathbf{h}_t\mathbf{h}_t'))$, since both of them are linear terms of $\mathbf{h}_t\mathbf{h}_t'$. The above problem is similar to the MKL dual form when we treat each base kernel as $\mathbf{Q}^{\mathbf{h}_t}$. Therefore, we can solve it based on its following

primal form, which is a convex optimization problem:

$$\min_{\mathbf{d}\in\mathcal{D},\mathbf{w}_t,\xi_l} \quad \frac{1}{2}\sum_{t=1}^{T}\frac{\|\mathbf{w}_t\|^2}{d_t} + C_1\sum_{l=1}^{L}\xi_l - C_2\sum_{t=1}^{T} d_t\rho(\mathbf{B},\mathbf{K}\circ(\mathbf{h}_t\mathbf{h}_t')) \tag{4.15}$$

$$\text{s.t.} \quad \sum_{t=1}^{T}\mathbf{w}_t'\psi(\mathbf{h}_t,\mathcal{B}_l,c,m) \geq \zeta_{l,c,m} - \xi_l, \forall l,c,m, \tag{4.16}$$

where $\psi(\mathbf{h}_t,\mathcal{B}_l,c,m)$ is used to denote the feature mapping induced by $\mathbf{Q}^{\mathbf{h}_t}$, *i.e.*, $\psi(\mathbf{h}_t,\mathcal{B}_l,c,m)'$ $\psi(\mathbf{h}_t,\mathcal{B}_{\tilde{l}},\tilde{c},\tilde{m}) = Q^{\mathbf{h}_t}_{u,v}$, in which $v = (\tilde{l}-1)\cdot C\cdot M + (\tilde{c}-1)\cdot M + \tilde{m}$, $u = (l-1)\cdot C\cdot M + (c-1)\cdot M + m$.

4.3.3.2 The Solution to (4.15)

We solve the convex problem in (4.15) by updating $\mathbf{d}$ and $\{\mathbf{w}_t,\xi_l\}$ in an alternative way.
Update d: When fixing $\{\mathbf{w}_t,\xi_l\}$, in order to solve $\mathbf{d}$, we introduce a dual variable τ for the constraint $\mathbf{d}'\mathbf{1}=1$ and derive the Lagrangian form of (4.15) as,

$$\begin{aligned}\hat{\mathcal{L}} &= \frac{1}{2}\sum_{t=1}^{T}\frac{\|\mathbf{w}_t\|^2}{d_t} + C_1\sum_{l=1}^{L}\xi_l - C_2\sum_{t=1}^{T} d_t\rho(\mathbf{B},\mathbf{K}\circ(\mathbf{h}_t\mathbf{h}_t')) \\ &\quad - \sum_{l,c,m}\alpha_{l,c,m}\Big(\sum_{t=1}^{T}\mathbf{w}_t'\psi(\mathbf{h}_t,\mathcal{B}_l,c,m) - \zeta_{l,c,m} + \xi_l\Big) + \tau\Big(\sum_{t=1}^{T} d_t - 1\Big)\end{aligned} \tag{4.17}$$

By setting the derivative of (4.17) *w.r.t.* each d_t to zero, we have

$$\tau = \frac{\|\mathbf{w}_t\|^2}{2d_t^2} + C_2\rho(\mathbf{B},\mathbf{K}\circ(\mathbf{h}_t\mathbf{h}_t')),\ \forall t = 1,\ldots,T, \tag{4.18}$$

which can be rewritten as,

$$d_t = \frac{\|\mathbf{w}_t\|}{\sqrt{2\tau - 2C_2\rho(\mathbf{B},\mathbf{K}\circ(\mathbf{h}_t\mathbf{h}_t'))}},\ \forall t = 1,\ldots,T. \tag{4.19}$$

Since the function on the righthand side of (4.19) is monotonically decreasing *w.r.t.* τ and $\mathbf{d}'\mathbf{1}=1$, we first apply binary search to seek for the value τ which satisfies the constraint $\sum_{t=1}^{T} d_t = 1$, and then recover d_t's based on (4.19).
Update $\mathbf{w}_t$: When $\mathbf{d}$ is fixed, $\boldsymbol{\alpha}$ can be solved in the dual form (4.14) and $\mathbf{w}_t$ can be recovered. Particularly, we can solve the problem in (4.14), which is a quadratic programming problem *w.r.t.* $\boldsymbol{\alpha}$, by employing QP solvers. Nevertheless, it is very time-consuming to use the existing QP solvers which are not specifically designed for our problem with $L\cdot C\cdot M$ variables. Thus, we solve this QP problem by using an efficient Sequential Minimal Optimization (SMO) algorithm, based on [20] and [47].

4.3.3.3 Cutting-Plane Algorithm

When using the above alternating optimization algorithm, the major challenge is that there are too many base kernels. Inspired by the work on Infinite Kernel Learning (IKL) [57], we begin with a small number of base kernels and then add a new violating base kernel at each iteration iteratively, which is named the cutting-plane algorithm. Since the MKL subproblem we need to solve at each iteration only has a small set of $\mathbf{h}$, it becomes much more efficient to optimize the whole problem. Particularly, we replace $\rho(\mathbf{B}, \mathbf{K} \circ (\mathbf{h}_t\mathbf{h}_t'))$ in (4.17) equivalently by using $\mathbf{h}_t'\mathbf{P}\mathbf{h}_t$ with $\mathbf{P} = \sum_{m \neq \tilde{m}} \mathbf{K} \circ ((\boldsymbol{\beta}_m - \boldsymbol{\beta}_{\tilde{m}})(\boldsymbol{\beta}_m - \boldsymbol{\beta}_{\tilde{m}})')$. By setting the derivatives of (4.17) *w.r.t.* $\{\mathbf{w}_t, \xi_t, d_t\}$ as zeros followed by some derivations, we can rewrite (4.14) as,

$$\max_{\tau, \boldsymbol{\alpha} \in \mathcal{A}} \quad -\tau + \boldsymbol{\zeta}'\boldsymbol{\alpha}, \tag{4.20}$$

$$\text{s.t.} \quad \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}^{\mathbf{h}_t}\boldsymbol{\alpha} + C_2\mathbf{h}_t'\mathbf{P}\mathbf{h}_t \leq \tau, \ \forall t, \tag{4.21}$$

which has a large number of constraints.

To solve (4.20), we begin with only one constraint and add a new violated constraint at each iteration. Specifically, since each constraint is related to an $\mathbf{h}_t$, the most violated constraint can be obtained by optimizing

$$\max_{\mathbf{h}} \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}^{\mathbf{h}}\boldsymbol{\alpha} + C_2\mathbf{h}'\mathbf{P}\mathbf{h} \tag{4.22}$$

After a simple deduction, (4.22) can be rewritten as,

$$\max_{\mathbf{h}} \mathbf{h}'(\frac{1}{2}\hat{\mathbf{Q}} \circ (\hat{\boldsymbol{\alpha}}\hat{\boldsymbol{\alpha}}') + C_2\mathbf{P})\mathbf{h}, \tag{4.23}$$

where $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^N$ is the shrinked vector of $\boldsymbol{\alpha}$ with its element $\hat{\alpha}_i = \frac{1}{|\mathcal{B}_l|}\sum_{c,m} \alpha_{l,c,m}$ for each $i \in \mathcal{I}_l$, and $\hat{\mathbf{Q}} \in \mathbb{R}^{N \times N}$ is the shrinked matrix of $\mathbf{Q}$ with its element $\hat{Q}_{i,j} = \sum_{c,\tilde{c},m,\tilde{m}} \gamma(i,j,c,\tilde{c},m,\tilde{m})\phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)$. The problem in (4.23) can be solved approximately by enumerating the binary indicator vector $\mathbf{h}$ bag by bag in order to maximize the objective value of (4.23) until there is no change in $\mathbf{h}$.

The proposed WSDG method is summarized in Algorithm 5.

[2]We initialize $\mathbf{h}_1$ by assigning the entries corresponding to the top $\eta|\mathcal{B}_l|$ instances (*i.e.*, with highest decision values) in each bag $\mathcal{B}_l$ to 1, and other entries to 0. In particular, we assign the labels of all training instances as their corresponding bag labels to train SVM classifiers, and then get the decision values of all training instances based on the learnt SVM classifiers.

Algorithm 5 Weakly Supervised Domain Generalization (WSDG) Algorithm

Require: The training data $\{(\mathcal{B}_l, Y_l)|_{l=1}^{L}\}$.

1: Initialize $t = 1$ and[2] $\mathcal{C} = \{\mathbf{h}_1\}$.

2: **repeat**

3: Set $t \leftarrow t + 1$.

4: Based on $\mathcal{H} = \mathcal{C}$, obtain $(\mathbf{d}, \boldsymbol{\alpha})$ by optimizing the MKL subproblem in (4.14).

5: Solving (4.23) to find the violated $\mathbf{h}_t$, which is added to the violation set (*i.e.*, $\mathcal{C} \leftarrow \mathcal{C} \bigcup \mathbf{h}_t$).

6: **until** The objective of (4.14) converges.

Ensure: The learnt classifier $f(\mathbf{x})$.

4.4 Weakly Supervised Domain Generalization using Privileged Information (WSDG-PI)

The web data are generally accompanied by massive and informative contextual information (*e.g.*, surrounding texts, tags, and captions). Although the contextual information is not available for the test data, they can still be used as privileged information to improve the performance of the learnt classifiers [98, 150]. Based on the above idea, we extend our WSDG approach by further utilizing privileged information, *i.e.*, the textual features extracted from the textual descriptions of web images/videos, which leads to our WSDG-PI approach.

Let us denote the textual feature of the i-th training sample as $\mathbf{z}_i$. Inspired by the works in [98, 150], we define $\tilde{f}_{c,m}(\mathbf{z}_i) = (\tilde{\mathbf{w}}_{c,m})'\tilde{\phi}(\mathbf{z}_i)$ as the slack function, in which $\tilde{\phi}$ is the feature mapping function for $\mathbf{z}_i$. For ease of presentation, we define the lefthand side of (4.10) as $F(\mathcal{B}_l, \tilde{c}, \tilde{m}) = \frac{1}{|\mathcal{B}_l|}\sum_{i\in I_l} h_i(\sum_{m=1}^{M} \hat{\beta}_{i,m}(\mathbf{w}_{Y_l,m})'\phi(\mathbf{x}_i) - (\mathbf{w}_{\tilde{c},\tilde{m}})'\phi(\mathbf{x}_i))$, and also define $\tilde{F}(\mathcal{B}_l, \tilde{c}, \tilde{m}) = \frac{1}{|\mathcal{B}_l|}\sum_{i\in I_l} h_i(\sum_{m=1}^{M} \hat{\beta}_{i,m}(\tilde{\mathbf{w}}_{Y_l,m})'\tilde{\phi}(\mathbf{z}_i) - (\tilde{\mathbf{w}}_{\tilde{c},\tilde{m}})'\tilde{\phi}(\mathbf{z}_i))$. Then we

formulate our WSDG-PI approach as,

$$\begin{aligned}
\min_{\substack{\mathbf{h}\in\mathcal{H},\xi_l,\epsilon_l \\ \mathbf{w}_{c,m},\tilde{\mathbf{w}}_{c,m}}} \quad & \frac{1}{2}\sum_{c=1}^{C}\sum_{m=1}^{M}\left(\|\mathbf{w}_{c,m}\|^2+\lambda\|\tilde{\mathbf{w}}_{c,m}\|^2\right)+C_1\sum_{l=1}^{L}(\xi_l+\epsilon_l) \\
& -C_2\rho(B,\mathbf{K}\circ(\mathbf{h}\mathbf{h}'))+C_3\sum_{l=1}^{L}\sum_{\tilde{m}=1}^{M}\sum_{\tilde{c}\neq Y_l}\tilde{F}(\mathcal{B}_l,\tilde{c},\tilde{m}) && (4.24)\\
\text{s.t.} \quad & F(\mathcal{B}_l,\tilde{c},\tilde{m})\geq\eta-\tilde{F}(\mathcal{B}_l,\tilde{c},\tilde{m})-\xi_l,\ \forall l,\tilde{m},\tilde{c}\neq Y_l, && (4.25)\\
& \tilde{F}(\mathcal{B}_l,\tilde{c},\tilde{m})\geq\eta-\epsilon_l,\quad \forall l,\tilde{m},\tilde{c}\neq Y_l, && (4.26)\\
& \xi_l\geq 0,\quad \forall l, && (4.27)\\
& \epsilon_l\geq 0,\quad \forall l, && (4.28)
\end{aligned}$$

where C_1, C_2, C_3, and λ are the tradeoff parameters, and ϵ_l is the slack variable introduced for the slack function $\tilde{F}(\mathcal{B}_l,\tilde{c},\tilde{m})$. As discussed in [150], the slack function $\tilde{F}(\mathcal{B}_l,\tilde{c},\tilde{m})$ plays the role of teacher by providing the explanations to the students, so we expect $\tilde{F}(\mathcal{B}_l,\tilde{c},\tilde{m})$ can well adjust the prediction of $F(\mathcal{B}_l,\tilde{c},\tilde{m})$ for the samples which are difficult to be classified.

To derive the solution to the above problem, we write the dual form of (4.24) as,

$$\begin{aligned}
\min_{\mathbf{h}\in\mathcal{H}}\max_{\boldsymbol{\alpha},\boldsymbol{\varsigma}} \quad & -\frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}^{\mathbf{h}}\boldsymbol{\alpha}-\frac{1}{2\lambda}(\boldsymbol{\alpha}+\boldsymbol{\varsigma}-C_3\mathbf{1})'\tilde{\mathbf{Q}}^{\mathbf{h}}(\boldsymbol{\alpha}+\boldsymbol{\varsigma}-C_3\mathbf{1})+\boldsymbol{\zeta}'(\boldsymbol{\alpha}+\boldsymbol{\varsigma}) \\
& -C_2\rho(\mathbf{B},\mathbf{K}\circ(\mathbf{h}\mathbf{h}')) && (4.29)\\
\text{s.t.} \quad & \sum_{c,m}\alpha_{l,c,m}=C_1,\quad \forall l \\
& \alpha_{l,c,m}\geq 0,\quad \forall l,c,m, \\
& \sum_{c,m}\varsigma_{l,c,m}=C_1,\quad \forall l \\
& \varsigma_{l,c,m}\geq 0,\quad \forall l,c,m,
\end{aligned}$$

where $\boldsymbol{\alpha}\in\mathbb{R}^{\tilde{D}}$ is a vector containing the dual variables $\alpha_{l,c,m}$, $\tilde{D}=L\cdot C\cdot M$, $\boldsymbol{\varsigma}\in\mathbb{R}^{\tilde{D}}$ is a vector containing the dual variables $\varsigma_{l,c,m}$, $\mathbf{Q}^{\mathbf{h}}$ is defined in the paragraph after (4.13), and $\tilde{\mathbf{Q}}^{\mathbf{h}}$ is defined by replacing $\phi(\mathbf{x})$ in $\mathbf{Q}^{\mathbf{h}}$ with $\tilde{\phi}(\mathbf{z})$. We leave the details of deriving the dual form of (4.24) in Appendix B.

Similarly to solving (4.13), we optimize over the linear combination coefficients of $\mathbf{h}_t\mathbf{h}_t'$'s given all feasible $\mathbf{h}_t\in\mathcal{H}$, *i.e.*, $\sum_{\mathbf{h}_t\in\mathcal{H}}d_t\mathbf{h}_t\mathbf{h}_t'$, where d_t is the combination coefficient

for $\mathbf{h}_t\mathbf{h}_t'$. We denote $T = |\mathcal{H}|$, $\mathbf{d} = [d_1, \ldots, d_T]'$, $\mathcal{D} = \{\mathbf{d}|\mathbf{d}'\mathbf{1} = 1, \mathbf{d} \geq 0\}$ as the feasible set of $\mathbf{d}$, $\mathcal{A}$ as the feasible set of $\boldsymbol{\alpha}$ in (4.29), and $\mathcal{E}$ as the feasible set of $\boldsymbol{\varsigma}$ in (4.29). Then, we can arrive at the problem as follows,

$$\begin{aligned}\min_{\mathbf{d}\in\mathcal{D}}\max_{\substack{\boldsymbol{\alpha}\in\mathcal{A}\\ \boldsymbol{\varsigma}\in\mathcal{E}}} \quad & -\frac{1}{2}\sum_{t=1}^{T} d_t\boldsymbol{\alpha}'\mathbf{Q}^{\mathbf{h}_t}\boldsymbol{\alpha} + \boldsymbol{\zeta}'(\boldsymbol{\alpha}+\boldsymbol{\varsigma}) \\ & -\frac{1}{2\lambda}\sum_{t=1}^{T} d_t(\boldsymbol{\alpha}+\boldsymbol{\varsigma}-C_3\mathbf{1})'\tilde{\mathbf{Q}}^{\mathbf{h}_t}(\boldsymbol{\alpha}+\boldsymbol{\varsigma}-C_3\mathbf{1}) - C_2\sum_{t=1}^{T} d_t\rho(\mathbf{B}, \mathbf{K}\circ(\mathbf{h}_t\mathbf{h}_t')).\end{aligned} \tag{4.30}$$

We solve (4.30) based on its primal problem, which is a Multiple Kernel Learning (MKL) problem and we can solve it in a similar way as in [82],

$$\begin{aligned}\min_{\substack{\mathbf{d}\in\mathcal{D},\xi_l,\epsilon_l\\ \mathbf{w}_t,\tilde{\mathbf{w}}_t}} \quad & \frac{1}{2}\sum_{t=1}^{T}\frac{\|\mathbf{w}_t\|^2}{d_t} + \frac{\lambda}{2}\sum_{t=1}^{T}\frac{\|\tilde{\mathbf{w}}_t\|^2}{d_t} + C_1\sum_{l=1}^{L}(\xi_l+\epsilon_l) \\ & -C_2\sum_{t=1}^{T} d_t\rho(\mathbf{B}, \mathbf{K}\circ(\mathbf{h}_t\mathbf{h}_t')) + C_3\sum_{t=1}^{T}\sum_{l,\tilde{m},\tilde{c}\neq Y_l}\tilde{\mathbf{w}}_t'\tilde{\psi}(\mathbf{h}_t, \mathcal{B}_l, \tilde{c}, \tilde{m}) \\ \text{s.t.} \quad & \sum_{t=1}^{T}\mathbf{w}_t'\psi(\mathbf{h}_t, \mathcal{B}_l, \tilde{c}, \tilde{m}) \\ & \geq \zeta_{l,\tilde{c},\tilde{m}} - \sum_{t=1}^{T}\tilde{\mathbf{w}}_t'\tilde{\psi}(\mathbf{h}_t, \mathcal{B}_l, \tilde{c}, \tilde{m}) - \xi_l, \forall l, \tilde{c}, \tilde{m}, \\ & \sum_{t=1}^{T}\tilde{\mathbf{w}}_t'\tilde{\psi}(\mathbf{h}_t, \mathcal{B}_l, \tilde{c}, \tilde{m}) \geq \zeta_{l,\tilde{c},\tilde{m}} - \epsilon_l, \quad \forall l, \tilde{c}, \tilde{m}.\end{aligned} \tag{4.31}$$

where $\psi(\mathbf{h}_t, \mathcal{B}_l, c, m)$ is defined below (4.15), $\tilde{\psi}(\mathbf{h}_t, \mathcal{B}_l, c, m)$ is the feature mapping induced by $\tilde{\mathbf{Q}}^{\mathbf{h}_t}$, *i.e.*, $\tilde{\psi}(\mathbf{h}_t, \mathcal{B}_l, c, m)'\tilde{\psi}(\mathbf{h}_t, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m}) = \tilde{Q}^{\mathbf{h}_t}_{u,v}$, in which $v = (\tilde{l}-1)\cdot C\cdot M + (\tilde{c}-1)\cdot M + \tilde{m}$, $u = (l-1)\cdot C\cdot M + (c-1)\cdot M + m$. $\mathbf{w}_t$ and $\tilde{\mathbf{w}}_t$ are defined as

$$\mathbf{w}_t = d_t\sum_{l,c,m}\alpha_{l,c,m}\psi(\mathbf{h}_t, \mathcal{B}_l, c, m), \tag{4.32}$$

$$\tilde{\mathbf{w}}_t = d_t\sum_{l,c,m}(\alpha_{l,c,m} + \varsigma_{l,c,m} - C_3)\tilde{\psi}(\mathbf{h}_t, \mathcal{B}_l, c, m). \tag{4.33}$$

Similar as (4.15), the problem in (4.31) is also a convex problem, which can be solved by updating $\mathbf{d}$ and $\{\mathbf{w}_t, \tilde{\mathbf{w}}_t, \xi_l, \epsilon_l\}$ alternatively.

Update d: When $\{\mathbf{w}_t, \tilde{\mathbf{w}}_t, \xi_l, \epsilon_l\}$ is fixed, we first introduce a dual variable τ for the constraint $\mathbf{d}'\mathbf{1} = 1$ to obtain the Lagrangian form of (4.31) similarly as (4.17). When the derivative of the Lagrangian form *w.r.t.* each d_t is set to zero, we can have

$$\tau = \frac{\|\mathbf{w}_t\|^2}{2d_t^2} + \lambda\frac{\|\tilde{\mathbf{w}}_t\|^2}{2d_t^2} + C_2\rho(\mathbf{B}, \mathbf{K} \circ (\mathbf{h}_t\mathbf{h}_t')), \forall t, \tag{4.34}$$

which leads to

$$d_t = \sqrt{\frac{\|\mathbf{w}_t\|^2 + \lambda\|\tilde{\mathbf{w}}_t\|^2}{2\tau - 2C_2\rho(\mathbf{B}, \mathbf{K} \circ (\mathbf{h}_t\mathbf{h}_t'))}}, \forall t. \tag{4.35}$$

Similarly to (4.19), (4.35) is also monotonically decreasing *w.r.t.* τ. So we also use the binary search method to seek for τ which satisfies $\sum_{t=1}^{T} d_t = 1$, and calculate d_t's by using (4.35).

Update $\{\mathbf{w}_t, \tilde{\mathbf{w}}_t, \xi_l, \epsilon_l\}$: When $\mathbf{d}$ is fixed, we solve $\boldsymbol{\alpha}$ and $\boldsymbol{\varsigma}$ in (4.30). Specifically, we concatenate $\boldsymbol{\alpha}$ and $\boldsymbol{\varsigma}$ into a long vector $\boldsymbol{\vartheta}$, and thus (4.30) becomes a QP problem *w.r.t.* $\boldsymbol{\vartheta}$. Since there are too many variables in $\boldsymbol{\vartheta}$, it is very inefficient to be solved based on QP solvers. Similar to Section 4.3.3.2, we use the SMO algorithm to solve (4.30).

Again, there are too many $\mathbf{h}_t\mathbf{h}_t'$'s when using the above alternating optimization procedure. Similar to Section 4.3.3.3, we employ the cutting-plane algorithm. In each iteration, we seek for the most violating indicator $\mathbf{h}$ by solving the following problem similar to solving (4.22),

$$\max_{\mathbf{h}} \quad \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}^{\mathbf{h}}\boldsymbol{\alpha} + \frac{1}{2\lambda}(\boldsymbol{\alpha} + \boldsymbol{\varsigma} - C_3\mathbf{1})'\tilde{\mathbf{Q}}^{\mathbf{h}}(\boldsymbol{\alpha} + \boldsymbol{\varsigma} - C_3\mathbf{1}) + C_2\rho(\mathbf{B}, \mathbf{K} \circ (\mathbf{h}\mathbf{h}')). \tag{4.36}$$

The whole algorithm of weakly supervised domain generalization using privileged information (WSDG-PI) is summarized in Algorithm 6. The testing stage of our WSDG-PI method is similar to that of WSDG as discussed in Section 4.3.2.2. Note $\tilde{\mathbf{w}}_{c,m}$'s are not used in the testing phase because the privileged information (*i.e.*, textual features) is not available for the test samples.

Time Complexity Analysis: Our WSDG-PI method consists of two steps, in which we first discover latent domains by solving the QP problem in (4.1) and then learn classifiers by solving the problem in (4.31). In the first step, according to [1], the time complexity

[3] We adopt the same initialization method as in Algorithm 5.

Algorithm 6 Weakly Supervised Domain Generalization using Privileged Information (WSDG-PI) Algorithm

Require: The training data $\{(\mathcal{B}_l, Y_l)|_{l=1}^{L}\}$.

1: Initialize $t = 1$ and[3] $\mathcal{C} = \{\mathbf{h}_1\}$.

2: **repeat**

3: Set $t \leftarrow t + 1$.

4: Obtain $(\mathbf{d}, \boldsymbol{\alpha}, \boldsymbol{\varsigma})$ by optimizing the MKL problem in (4.30) based on $\mathcal{H} = \mathcal{C}$.

5: Solve (4.36) to find the violated $\mathbf{h}_t$, which is added to the violation set (*i.e.*, $\mathcal{C} \leftarrow \mathcal{C} \bigcup \mathbf{h}_t$).

6: **until** The objective of (4.30) converges.

Ensure: The learnt classifier $f(\mathbf{x})$.

for solving the non-convex QP problem in (4.1) is $O((NM)^3)$, in which N (*resp.*, M) is the number of training samples (*resp.*, latent domains).

In the second step, we solve the convex problem in (4.31) by employing the cutting-plane algorithm, in which we add the most violated label candidate and solve the MKL subproblem at each iteration. Since it is much more time-consuming to solve the MKL subproblems, the time complexity of the problem in (4.31) can be roughly estimated as $T \cdot O(\text{MKL})$, in which T is the number of iterations and $O(\text{MKL})$ is the time complexity of the MKL subproblem.

Nevertheless, no previous work has studied the time complexity of MKL theoretically. When solving the MKL problem in (4.30), the most time-consuming step is to solve the convex QP problem *w.r.t.* $\boldsymbol{\alpha}$ and $\boldsymbol{\varsigma}$ when fixing $\mathbf{d}$, which is solved by using our sequential minimal optimization (SMO) solver. According to [127], the time complexity of SMO is between $O(LCM)$ and $O((LCM)^{2.3})$, in which M is the number of latent domains, and L and C are the number of bags and categories, respectively. So the time complexity of MKL (*i.e.*, $O(\text{MKL})$) is between $t \cdot O(LCM)$ and $t \cdot O((LCM)^{2.3})$, where t is the number of iterations in MKL. Since our WSDG method also employs the cutting-plane algorithm and solves an MKL subproblem by using our SMO solver at each iteration, its time complexity can be analysed similarly.

Table 4.1: Accuracies (%) of baselines and our WSDG method including two special cases for the image classification and video event recognition tasks. We denote the best results in boldface.

Method	Testing Dataset			
	Kodak	CCV	Caltech (CNN)	Caltech (Classeme)
SVM [29]	40.00	45.80	70.93	33.87
sMIL [17]	46.15	50.52	71.33	34.67
mi-SVM [4]	43.59	51.31	71.47	35.07
MIL-CPB [96]	46.67	51.76	71.60	34.93
KI-SVM [99]	46.15	46.36	71.20	35.20
DICA [110]	45.12	50.80	70.80	35.60
LRESVM [164]	49.74	54.69	72.93	36.93
[69] (Match)	41.03	50.18	71.07	36.53
[69] (Ensemble)	42.05	49.96	70.08	36.80
[59] (Match)	45.13	50.78	71.47	36.40
[59](Ensemble)	46.15	52.20	72.40	37.07
Sub-Cate [67]	45.13	53.17	72.27	36.40
MMDL [158]	47.69	54.70	72.80	37.47
WSDG_sim1	48.21	52.02	71.87	35.47
WSDG_sim2	50.26	55.37	74.00	38.13
WSDG	**51.28**	**56.83**	**75.20**	**38.80**

4.5 Experiments

In this section, the effectiveness of our weakly supervised domain generalization (WSDG) approach is demonstrated for image classification and video event recognition by comprehensive experiments on three benchmark datasets. We also analyze why we can learn a better classifier and discover more distinctive latent domains by removing outliers in our WSDG method. Moreover, we extend our WSDG method to WSDG-PI and the experimental results indicate the benefit of utilizing privileged information (*i.e.*, additional textual features).

4.5.1 Weakly Supervised Domain Generalization

Experimental Settings: Our WSDG method is evaluated by utilizing the videos and images crawled from web to train classifiers for video event recognition and image classification tasks, respectively. In this chapter, we use multi-class classification accuracy for performance evaluation, as suggested in [69].

For the video event recognition task, we employ two benchmark datasets Kodak [107] and CCV [79]. The Kodak dataset contains 195 consumer videos distributed over 6 event categories. The CCV dataset [79] contains 4659 and 4658 videos distributed over 20 categories for training and testing, respectively. Strictly following the experimental setting in [44], only the videos belonging to the related event categories are used by merging the categories sharing similar semantic meanings, which finally leads to 2440 videos from five event classes.

In order to collect the training set for video event recognition from the Internet, web videos are crawled from *Flickr.com* by querying based on the 6 (*resp.*, 5) event category names for the Kodak (*resp.*, CCV) test set. For each query, 100 relevant web videos are downloaded and partitioned uniformly according to their ranks to construct 20 bags with 5 instances in each bag.

For the visual features used for video event recognition, we firstly extract Improved Dense Trajectory (IDT) descriptors which include 100-dim trajectory, 96-dim HOG, 108-dim HOF, and 192-dim MBH by using the source code provided in [156]. Then, following the Fisher vector encoding method in [156], we train 256 Gaussian Mixture Models (GMMs) by using the IDT descriptors from the videos in the Flickr training dataset and generate the 128,000-dim Fisher vector for each video on both training and test datasets. Finally, following [42], we use the Aligned Space-Time Pyramid Matching (ASTPM) method to obtain the video clip distances based on Fisher vectors. When employing the ASTPM method, we set the volume size as $1/2^l$ ($l = 1, \ldots, L$) of the original video in height, width, and temporal dimension, in which L is set as 2 as suggested in [42]. Based on the obtained distance matrices, we calculate the average of RBF kernel matrices from different pyramid levels, which are used in the training or testing procedure.

For the image classification task, the BING dataset [9] is used as the source domain while the Caltech-256 dataset is used as the test set. Strictly following the experimental

setting in [69], we only utilize the images belonging to the first 30 categories in the BING and Caltech-256 dataset. Following [69], 20 training images and 25 test images are used per category, which leads to totally 600 (*resp.*, 750) training (*resp.*, test) samples. Similar to video event recognition, we uniformly partition the training images based on the given indices to construct training bags with 5 instances in each bag. We employ both traditional and deep learning features on the Bing-Caltech dataset. Specifically, for traditional features, we use the $2,625$-dim classeme feature provided in [9]. For deep learning features, we the DeCAF features [38] (*i.e.*, the 6th layer outputs), which leads to $4,096$-dim DeCAF_6 features.

As web data are not associated with explicit domain labels and even the number of latent domains is not given, we follow [69] to assume there are 2 latent domains for all methods on all datasets. We empirically fix $C_1 = C_2 = 1$, $\eta = 0.8$ (*resp.*, 0.2) for our WSDG approach for image classification (*resp.*, video event recognition). For fair comparison, the optimal parameters are selected for baseline methods based on their best performances on the test dataset.

Baselines: Our WSDG approach is compared with three sets of baselines: the multi-instance learning (MIL) baselines, the domain generalization baselines, and the latent domain discovering baselines. The MIL methods can be categorized into the instance-level methods including mi-SVM [4] and MIL-CPB [96] and the bag-level methods including sMIL [17] and KI-SVM [99]. The domain generalization methods contain the low-rank exemplar SVM (LRESVM) method [164] and the domain-invariant component analysis (DICA) method [110]. Note that the approach in [80] cannot be directly applied to our tasks since the training web data are not associated with domain labels. For the two latent domain discovering methods [59, 69], we employ two strategies named "Match" and "Ensemble" following the suggestion in [164].

Furthermore, as the MMDL method in [158] and the discriminative sub-categorization method [67] are related to our approach, our method is also compared with them.

To demonstrate the benefits of discovering latent domains and validate our MMD-based regularizer in (4.9), the performances of two simplified versions of our WSDG approach are additionally reported. We refer to them as WSDG_sim1 and WSDG_sim2 respectively. Specifically, in WSDG_sim2, we set $C_2 = 0$ to remove the MMD-based

regularizer $\rho(\mathbf{B}, \mathbf{K} \circ (\mathbf{h}\mathbf{h}'))$ in our WSDG approach. Based on WSDG_sim2, the latent domain issues are further ignored by setting the number of latent domains to one (*i.e.*, $M = 1$) and we refer to this case as WSDG_sim1, in which our objective in (4.9) can be reduced to that in (4.6).

Experimental Results: The experimental results are reported in Table 4.1, from which we can see that the sub-categorization baselines MMDL and Sub-Cate, the domain generalization baselines LRESVM and DICA, and the latent domain discovering baselines [69] and [59] generally outperform SVM. These results show that exploiting additional information such as subcategories, low-rank structure, or hidden latent domains in the training samples is helpful.

Another observation is that the MIL baselines (*e.g.*, mi-SVM, MIL-CPB, sMIL, and KI-SVM) outperform SVM on all three datasets, although various MIL assumptions are used in these methods. We also observe that MMDL outperforms both the Sub-Cate method and MIL baselines, possibly because it simultaneously exploits subcategories and utilizes the MIL technique to cope with the label noise in web data.

The performances of MIL baselines are worse than that of our special case WSDG_sim1, which might because the classifiers for different categories are jointly learnt. WSDG_sim1 is worse than WSDG_sim2 on all three datasets, which demonstrates the advantage of integrating multiple classifiers from different latent domains. Moreover, our WSDG approach achieves better performances than WSDG_sim2 on all three datasets, which proves our MMD-based regularizer in (4.9) is valid. Another observation is that WSDG and WSDG_sim2 are better than all the MIL baselines [4, 17, 96, 99] and the domain generalization baselines LRESVM and DICA, which shows the advantage of handling label noise and exploiting latent domains in the web images/videos at the same time.

Finally, the best results are achieved by our WSDG method on all datasets and the results clearly show that our WSDG method is effective for the image classification and video event recognition tasks by utilizing the web data.

4.5.2 Experimental Analysis on WSDG

Recall that in our WSDG method, we tend to identify a subset of non-outliers from the training samples and simultaneously expect the selected samples coming from more distinctive latent domains by using the indicator $\mathbf{h}$ in (4.9). Let us take image classification

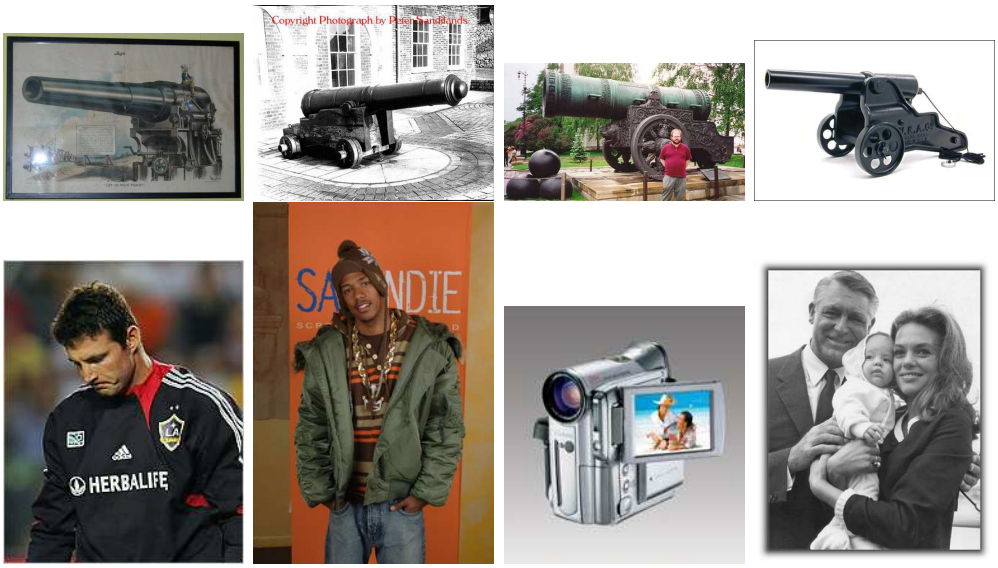


Figure 4.2: The top and bottom rows show the most and least confident images for the category "cannon" on the Bing dataset, respectively.

task (*i.e.*, the training and test sets are the Bing and Caltech-256 datasets, respectively) as an example to show the benefits by introducing $\mathbf{h}$ for removing outliers and discovering more distinctive latent domains.

We firstly demonstrate the effectiveness of our WSDG method for removing the outliers. Note that the problem for solving a binary indicator $\mathbf{h}$ is relaxed to seeking for a linear combination of feasible $\mathbf{h}_t$'s, so we calculate $\tilde{\mathbf{h}} = \sum_{t=1}^{T} d_t \mathbf{h}_t$ as the approximation of $\mathbf{h}$, where d_t and $\mathbf{h}_t$ are learnt by solving (4.15). Intuitively, for each element $\tilde{h}_i$ in the vector $\tilde{\mathbf{h}}$, the higher value $\tilde{h}_i$ indicates it is more confident that the corresponding training image is a true positive instance. We show the most and least confident images from the category "cannon" in the Bing dataset and their corresponding values $\tilde{h}_i$'s in Figure 4.2. We can observe that the images with the highest values $\tilde{h}_i$'s are all true positive instances (see the top row), while the images with the lowest values $\tilde{h}_i$'s are the outliers (see the bottom row). This indicates that our WSDG method is able to remove the outliers from the training samples, and thus we can learn more robust classifiers for the domain generalization problem.

In order to demonstrate the effectiveness of our WSDG approach for constructing more distinctive latent domains, we calculate the sum of MMDs (SMMDs) between each

Table 4.2: The sum of MMDs (SMMDs) between each pair of latent domains by using different methods.

Method	[69]	[59]	WSDG
SMMDs	24.46	27.08	31.56

pair of latent domains to measure the distinctiveness of latent domains. We also compare our WSDG method with the latent domain discovering methods in [69] and [59]. For [69], we denote the binary latent domain indictor as $\bar{\pi}_{i,m}$'s, where $\bar{\pi}_{i,m}$ indicates whether the i-th training sample comes from the m-th hidden latent domain, and then we calculate the sum of MMDs between each pair of latent domains as $\sum_{m\neq\tilde{m}} \|\frac{1}{N_m}\sum_{i=1}^{N}\bar{\pi}_{i,m}\phi(\mathbf{x}_i) - \frac{1}{N_{\tilde{m}}}\sum_{i=1}^{N}\bar{\pi}_{i,\tilde{m}}\phi(\mathbf{x}_i)\|^2$. For [59], we calculate the sum of MMDs between each pair of latent domains based on the soft assignment coefficients $\boldsymbol{\beta}_m$'s as $\sum_{m\neq\tilde{m}}(\boldsymbol{\beta}_m - \boldsymbol{\beta}_{\tilde{m}})'\mathbf{K}(\boldsymbol{\beta}_m - \boldsymbol{\beta}_{\tilde{m}})$ (see (4.2)). For our method, we first calculate $\bar{\boldsymbol{\beta}}_m = \frac{\tilde{\mathbf{h}}\circ\boldsymbol{\beta}_m}{\|\tilde{\mathbf{h}}\circ\boldsymbol{\beta}_m\|_1}$, and then obtain the sum of MMDs between each pair of latent domains by using $\sum_{m\neq\tilde{m}}(\bar{\boldsymbol{\beta}}_m - \bar{\boldsymbol{\beta}}_{\tilde{m}})'\mathbf{K}(\bar{\boldsymbol{\beta}}_m - \bar{\boldsymbol{\beta}}_{\tilde{m}})$.

In Table 4.2, the image classification task is taken as an example to report the sum of MMDs between each pair of latent domains from different methods. It can be seen from Table 4.2 that SMMDs of [59] is larger than that of [69], possibly because [59] is specifically designed to maximize the sum of MMDs between each pair of latent domains. We also observe that SMMDs of our WSDG approach is larger than that of [59], which demonstrates our WSDG method can construct more distinctive latent domains by removing the outliers. So our WSDG method has better generalization ability than [69] and [59].

4.5.3 Weakly Supervised Domain Generalization using Privileged Information

Experimental Settings: Our proposed WSDG-PI approach is evaluated using the Flickr web video dataset (*resp.*, the CCV and Kodak datasets) as the training set (*resp.*, the test sets). Note that the Bing dataset provided in [9] is not associated with textual information, so our WSDG-PI method cannot be evaluated on the Caltech-256 dataset.

Table 4.3: Accuracies (%) of the baselines and our methods for the video event recognition task. We denote the best results in boldface.

Method	Testing Dataset	
	Kodak	CCV
SVM [29]	40.00	45.80
SVM-2K [49]	46.15	51.33
KCCA [65]	45.64	51.05
Classeme [146]	44.62	47.67
RT [135]	43.59	49.22
SVM+ [150]	47.69	52.69
sMIL-PI [98]	49.23	54.88
WSDG	51.28	56.83
WSDG-PI	**55.38**	**58.15**

We crawl the surrounding tags of each Flickr video and extract a $2,000$-dim term frequency (TF) feature based on the associated tags for each video. The vocabulary when extracting TF features is constructed by using $2,000$ most frequent words after removing the stop-words. These textual features of training data are considered as privileged information. All other settings are identical to those in Section 4.5.1. WSDG-PI has two more parameters C_3 and λ, compared with WSDG. We empirically fix C_3 as 0.1 and λ as 10 on both datasets. For the baselines, the optimal parameters are selected based on their best performances on the test dataset.

Baselines: Our method is compared with RankTransfer (RT) [135] and SVM+ [150]. Moreover, we additionally include Classeme [146] as well as two multi-view learning methods SVM-2K [49] and KCCA [65] as the baselines because they can also utilize both textual features and visual features of training samples.

- *Classeme* [146]: For each word in the $2,000$-dim textual features, we learn a classeme classifier based on the relevant and irrelevant samples. For each sample from both training set and test set, the visual features are augmented with

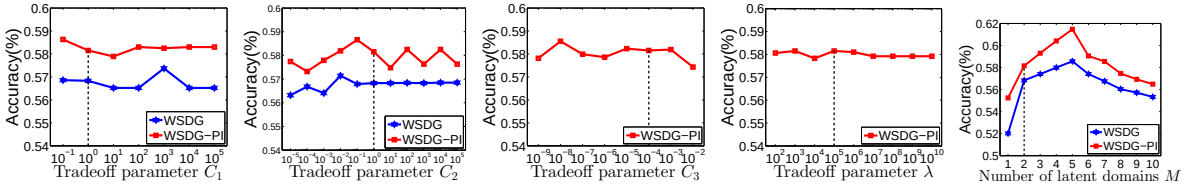


Figure 4.3: Accuracies of our WSDG and WSDG-PI methods on the CCV dataset when using different trade-off parameters. The vertical dash lines indicate the default parameters.

the 2,000 decision values which are obtained by using 2,000 pre-learnt classeme classifiers. Finally, we use the the augmented features to train the SVM classifiers and predict the test samples.

- *SVM-2K* [49]: SVM-2K classifiers are trained by utilizing both visual features and textual features of training data. Then, the classifier based on visual features is used to classify the test samples.

- *Kernel Canonical Correlation Analysis (KCCA)* [65]: KCCA is employed on the visual features and textual features of training data. Then, we use the projected visual features to train SVM classifiers and classify the test samples.

We also compare our WSDG-PI method with sMIL-PI [98], which can simultaneously cope with label noise and take advantage of privileged information (*i.e.*, textual features). We additionally include SVM and WSDG for comparison.

Experimental Results: The experimental results are reported in Table 4.3, from which we observe that learning using privileged information methods SVM+ and RT outperform SVM on both datasets, which indicates the advantage of utilizing privileged information (*i.e.*, additional textual features). Besides, multi-view approaches SVM-2K and KCCA also outperform SVM on both datasets after employing both visual features and textual features. We also observe that Classeme outperforms SVM on both datasets. One possible explanation is that it is helpful to augment the visual features with the decision values obtained by using classeme classifiers. Moreover, sMIL-PI and our WSDG-PI method are better than sMIL reported in Table 4.1 and WSDG respectively on both datasets, which again demonstrates the benefits of utilizing the textual features as privileged information.

Table 4.4: Training time (s) of the baselines without using privileged information and our WSDG approach on the Bing and CCV dataset.

Method	KI-SVM	[69]	[59]	DICA	LRESVM	Sub-Cate	MMDL	WSDG
Bing	94.89	213.64	19.54	126.61	2986.57	436.18	195.01	102.15
CCV	20.50	189.91	17.49	83.85	2484.31	77.62	46.43	39.54

Table 4.5: Training time (s) of the baselines using privileged information and our WSDG-PI approach on the CCV dataset.

Method	SVM-2K	KCCA	Classeme	RT	SVM+	sMIL-PI	WSDG-PI
CCV	31.67	41.32	1526.12	89.90	37.88	29.16	72.40

Finally, our method WSDG-PI outperforms all the baselines on both datasets, which indicates the benefits of simultaneously handling label noise, exploiting privileged information, and learning robust classifiers for better generalization ability.

4.5.4 Sensitivity of Our Approaches *w.r.t.* Parameters

We take the CCV dataset as an example to study the performance variation of our WSDG and WSDG-PI methods by varying one parameter when fixing all other parameters as their default values. Note that C_1, C_2, and M (*i.e.*, the number of latent domains) are the common parameters shared by our WSDG and WSDG-PI methods, while C_3 and γ are the additional parameters of WSDG-PI method. From Figure 4.3, we observe that our methods are relatively robust when the trade-off parameters C_1, C_2, C_3, and γ are varied in certain ranges. We also observe that the results of our methods are improved when M increases but less than 5. If M increases over 5, the results of our methods decrease. One possible explanation is that the training set is considerably diverse, so it contains more than two latent domains. On the other hand, the total number of training samples is limited (only 600 training images/videos on the Bing/Flickr dataset), so the results of our methods will decrease if we use too many latent domains.

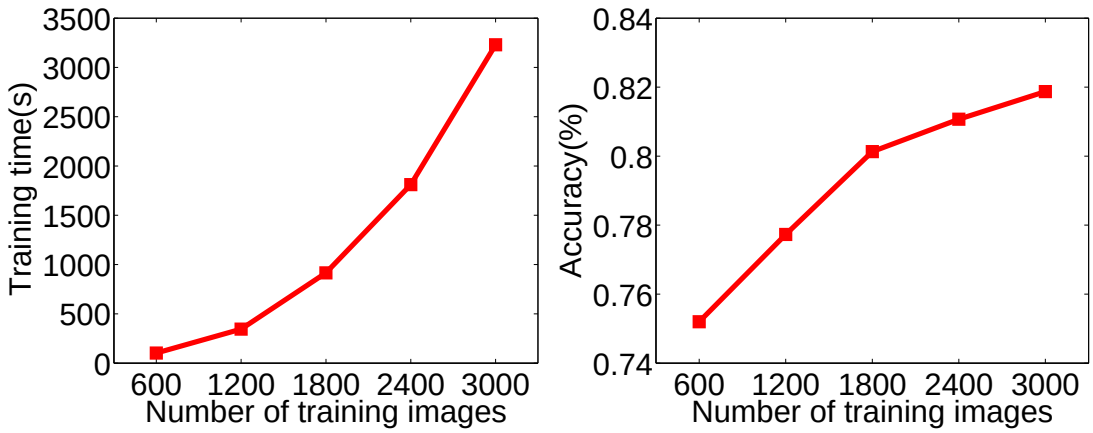


Figure 4.4: The training time and accuracies of our WSDG method with respect to the number of training images on the Bing dataset.

4.5.5 Comparison of Training Time

We compare the training time of our WSDG and WSDG-PI methods with other baseline methods. All the experiments are conducted on a server machine with 18 GB RAM and Intel Xeon 3.33 GHz CPUs using a single thread. Let us take the Bing and CCV datasets as two examples. In Table 4.4, we report the training time of our WSDG method and other baselines without using privileged information. We observe that our WSDG method is more efficient than other baselines except [59] and KI-SVM. WSDG is slower than [59] because we need to solve (4.9) instead of directly using SVM after employing the latent domain discovery technique in [59]. KI-SVM is also faster than WSDG. One possible explanation is that we need to solve a more complex subproblem in each iteration.

In Table 4.5, we report the training time of our WSDG-PI method and other baseline methods using privileged information. Note that the images in the Bing dataset do not have additional textual information, so we only report the training time on the CCV dataset in Table 4.5. The training time of WSDG-PI is longer than that of WSDG reported in Table 4.4, because we need to solve a larger scale QP problem at each iteration in our WSDG-PI method. Our WSDG-PI method is still reasonably efficient when compared with other baseline methods.

4.5.6 Time Complexity and Scalability of Our Approach

Let us take the image classification task with Bing as the training set and Caltech-256 as the test set as an example to demonstrate the scalability of our WSDG method. As the Bing dataset and its associated training indices *w.r.t.* various numbers of training samples per category are provided in [9], we use various numbers of training samples for each category (*i.e.*, [20, 40, 60, 80, 100]) to construct the training set in order to evaluate the performance and the scalability of our algorithms. Since we use 30 categories on the Bing dataset and n training samples per category, we have totally $30n$ training samples. The accuracies and the training time with various numbers of training samples are reported in Figure 4.4, from which we observe that both the accuracy and the training time increase as the number of training samples increases.

4.6 Summary

In this chapter, a novel weakly supervised domain generalization approach WSDG has been proposed for visual recognition tasks by utilizing loosely labeled web images/videos as training data. Our WSDG method is able to handle the label noise in training web data and has good generalization ability to arbitrary target domain. Additionally, we have extended our WSDG approach to WSDG-PI by utilizing textual descriptions of training web data as privileged information. The effectiveness of our WSDG and WSDG-PI methods has also been demonstrated by the comprehensive experiments.

Chapter 5

An Exemplar-based Multi-view Domain Generalization Framework for Visual Recognition

In this chapter, we propose a new exemplar-based multi-view domain generalization (EMVDG) framework for visual recognition, by learning robust classifiers which are able to generalize well to arbitrary target domain based on the training samples with multiple types of features (*i.e.*, multi-view features). In this framework, we aim to address two issues simultaneously: 1) the distribution of training samples (*i.e.*, the source domain) is often considerably different from that of test samples (*i.e.*, the target domain), so the performance of the classifiers learnt on the source domain may drop significantly on the target domain. Moreover, the test data are often unseen during the training procedure; 2) when the training data are associated with multi-view features, the recognition performance can be further improved by exploiting the relation among multiple types of features. To address the first issue, considering that it has been shown that fusing multiple SVM classifiers can enhance the domain generalization ability, we build our EMVDG framework upon exemplar SVMs, in which a set of exemplar SVM classifiers are learnt with each one trained based on one positive training sample and all the negative training samples. When the source domain contains multiple latent domains, the learnt exemplar SVM classifiers are expected to be grouped into multiple clusters. To address the second issue, we propose two approaches under the EMVDG framework based on the consensus principle and the complementary principle, respectively. Specifically, we propose an EMVDG_CO method by adding a co-regularizer to enforce the cluster structures of exemplar SVM classifiers on different views to be consistent based on the consensus principle. Inspired by multiple kernel learning (MKL), we also propose another

EMVDG_MK method by fusing the exemplar SVM classifiers from different views based on the complementary principle. In addition, we further extend our EMVDG framework to exemplar-based multi-view domain adaptation (EMVDA) framework when the unlabeled target domain data are available during the training procedure. The effectiveness of our EMVDG and EMVDA frameworks for visual recognition is clearly demonstrated by comprehensive experiments on three benchmark datasets.

5.1 Introduction

In the field of visual recognition, the data distributions of the training data and the test data are usually quite different, in which the training set (*resp.*, the test set) are referred to as the source domain (*resp.*, the target domain). Recently, abundant domain adaptation approaches [7, 16, 25, 39, 53, 60, 62, 71, 115]were proposed to reduce the data distribution mismatch between the source domain and the target domain explicitly. Nevertheless, the target domain samples are often unavailable during the training procedure and this problem is named as domain generalization [110]. In comparison with domain adaptation, domain generalization aims to learn robust classifiers that can generalize well to arbitrary target domain. More recently, several domain generalization approaches [110, 114, 117, 164] were also developed to enhance the generalization capability of the classifiers learnt on the source domain. For more details about domain generalization and adaptation, please refer to Section 2.4 and Section 2.3.

Most of the existing approaches for domain generalization or domain adaptation only utilize one type of feature in the training and test stage. In fact, when the training and test data are associated with multiple types of features, the recognition performance can be enhanced by exploiting the relation among multiple types of features (see Section 5.2 for the details). Some recently proposed domain adaptation approaches [11, 39, 166, 174] are based on multiple types of features, which aim to tackle with the data distribution mismatch and simultaneously exploit the relation among multiple types of features. In [11], Blitzer *et al.* use Canonical Correlation Analysis (CCA) to learn the projection matrices, based on which the classifiers learnt on the source domain are adapted to the target domain. In [174], different weights are assigned to the training samples based

on the Maximum Mean Discrepancy (MMD), while the prediction scores obtained on multiple views are expected to be consistent. In [166], Yang *et al.* incorporate an MMD based regularizer into the CCA framework. The approach in [39] can be used to learn the kernel weights to cope with the domain distribution mismatch by treating each view as a kernel. However, the above multi-view approaches [11, 39, 166, 174] require the target domain samples in the training stage, which are not required by our domain generalization methods.

To this end, we propose an exemplar-based multi-view domain generalization (EMVDG) framework by utilizing multi-view source domain data to learn robust classifiers which are able to generalize well to arbitrary target domain. On one hand, our approach is inspired by the recent work [164] which demonstrates that fusing multiple SVM classifiers can enhance the domain generalization capability. In particular, our EMVDG framework builds upon exemplar SVMs [108] with each SVM classifier learnt based on one positive training sample together with all the negative training samples. According to the assumptions in [59, 69, 164], the source domain may contain multiple hidden latent domains. Thus, the exemplar SVM classifiers, which correspond to the positive samples belonging to the same hidden latent domain, are expected to be similar. Therefore, the exemplar SVM classifiers can be grouped into multiple clusters, which can be achieved by using low-rank techniques (*e.g.*, nuclear norm based regularizer or low rank representation (LRR)).

On the other hand, in order to take full advantage of multi-view features, we propose two methods under the EMVDG framework based on the consensus principle and the complementary principle [161], respectively. Without loss of generality, the consensus principle expects the information of multiple views to be consistent while the complementary principle assumes that each view may contain some information which are missing in the other views so that multiple views can be jointly used to make the data representation more comprehensive. In this chapter, for the consensus principle, we enforce the consistency of inherent cluster structures on different views by adding a co-regularizer, which uses low-rank representation (LRR) [103] based on the the weight vectors of exemplar SVM classifiers. This method is named as EMVDG_CO. For the complementary principle, we linearly combine multiple kernels on different views as in multiple kernel learning (MKL) [6], and simultaneously enforce the dual matrix, which consists of the

dual vectors of exemplar SVM classifiers, to be low-rank by adding a nuclear norm based regularizer. We refer to this approach as EMVDG_MK. For both methods, alternating optimization algorithms are developed to solve the nontrivial optimization problems.

5.2 Related Work

This chapter is related to the domain generalization methods [110, 164]. Among the exisiting domain generalization methods, this chapter is more related to [164], which builds upon exemplar SVMs [108] to explore the low-rank structure in positive source domain samples. However, the above approaches [110, 164] only focus on one type of feature, while this chapter focuses on domain generalization in the multi-view scenario.

This chapter is also related to the latent domain discovering methods [59, 69]. However, the above methods require the number of hidden latent domains and these methods do not discuss how to employ multiple types of features effectively.

In this chapter, our EMVDG framework is also extended to EMVDA for domain adaptation. As mentioned in Section 5.1, some domain adaptation approaches [11, 39, 166, 174] can be used in the multi-view scenario. However, their methods require the target domain samples in the training stage, which are not available for domain generalization.

Finally, this chapter is related to the multi-view learning approaches [6, 34, 49, 65, 73]. Generally, the existing multi-view learning methods mainly rely on either the consensus principle or the complementary principle [161]. For the consensus principle, the approach in [65] first uses Kernel Canonical Correlation Analysis (KCCA) to transform the training and test features, and then learns SVM classifiers by using the transformed features, while the work in [49] formulates this two-stage approach as one unified optimization problem. In [34], Ding *et al.* proposed to learn a common low-rank subspace among multiple types of features. For the complementary principle, the linear combination of multiple kernels on different types of features is used to improve the performance in the multiple kernel learning methods [6, 88]. In addition, some multi-view semi-supervised learning approaches [12, 141] have also been proposed. For manifold based approaches, a semi-supervised Laplacian regularizer is incorporated into KCCA in [10] while the average matrix of multiple Laplacian matrices based on multi-view features is used in

a semi-supervised learning method in [141]. In co-training [12], Blum *et al.* select the confident unlabeled training samples by utilizing the classifier learnt on one view and add these confident samples to the labeled training set for learning the classifier on the other view in an iterative way. For more details about multi-view learning, please refer to the recent survey [161]. However, all the above multi-view learning methods assume the data distribution of the training data and the test data are the same, while our frameworks do not have this assumption.

5.3 Exemplar-based Multi-view Domain Generalization

In this section, an exemplar-based multi-view domain generalization (EMVDG) framework is proposed. In the following, we first introduce domain generalization with exemplar SVMs briefly in Section 5.3.1, and then introduce our two methods under the EMVDG framework: EMVDG_CO in Section 5.3.2 and EMVDG_MK in Section 5.3.3.

In this chapter, we explore the multi-view domain generalization problem in the binary classification scenario. Suppose the source domain contains n positive training samples and m negative training samples, in which each sample is associated with V types of features, then each positive training sample can be denoted as $\mathbf{x}_i^+ = (\mathbf{x}_i^{1+}, \ldots, \mathbf{x}_i^{V+})$, $i = 1, \ldots, n$, and each negative training sample can be denoted as $\mathbf{x}_j^- = (\mathbf{x}_j^{1-}, \ldots, \mathbf{x}_j^{V-})$, $j = 1, \ldots, m$.

5.3.1 Domain Generalization with Exemplar SVMs

Domain generalization targets at learning robust classifiers that are able to generalize well to arbitrary target domain by utilizing the source domain samples, which can be achieved by fusing multiple SVM classifiers as discussed in Section 5.2. Specifically, when the source domain data are assumed to be sampled from multiple latent domains, the latent domain labels are given (*i.e.*, in [110]) or obtained by using latent domain discovering methods [59, 69]. Then, the classifiers learnt based on each latent domain are integrated to predict the target domain data. Since the training samples within each latent domain are with more coherent data distribution, each classifier corresponding to

each latent domain should be more discriminative and the integrated classifier should be more robust to the various data distribution of the unseen target domain.

However, in the real world scenario, the variance of training samples is likely to be affected by complicated hidden factors that often overlap and interact with each other. Considering that it is a very challenging task to explicitly discover the hidden latent domains, low-rank exemplar SVM (LRESVM) was proposed in [164], which utilizes the low-rank structure of positive source domain data. It is worth noting that this approach builds upon the exemplar SVMs [108] with each SVM classifier learnt based on one positive source domain sample and all negative source domain samples. Exemplar SVM targets at capturing the specific feature of individual positive training sample, which has been widely used in many computer vision tasks such as object detection [108], image retrieval [139], and feature encoding [173]. By using $f_i(\mathbf{x}) = \mathbf{w}_i'\mathbf{x}$ to denote the exemplar SVM classifier learnt based on the i-th positive sample $\mathbf{x}_i^+$ and all the negative samples[1] $\{\mathbf{x}_j^-|_{j=1}^m\}$ (we only focus on single-view learning in this section, so the superscript v is omitted for ease of presentation), the formulation for learning n exemplar SVMs can be written as,

$$\begin{aligned} \min_{\mathbf{w}_i,\xi_i,\epsilon_{ij}} \quad & \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{w}_i\|^2 + C\sum_{i=1}^{n}\xi_i + C\sum_{i=1}^{n}\sum_{j=1}^{m}\epsilon_{ij} \\ \text{s.t.} \quad & \mathbf{w}_i{}'\mathbf{x}_i^+ \geq 1-\xi_i, \quad \xi_i \geq 0, \quad \forall i, \\ & \mathbf{w}_i{}'\mathbf{x}_j^- \leq -1+\epsilon_{ij}, \quad \epsilon_{ij} \geq 0, \quad \forall i, \forall j, \end{aligned} \tag{5.1}$$

where C is a trade-off parameter, ξ_i's and ϵ_{ij}'s are the slack variables, and $\|\mathbf{w}_i\|^2$ is the regularizer to control the complexity of $\mathbf{w}_i$.

As the positive samples belonging to the same hidden latent domain should be similar, the work in [164] enforces the prediction score matrix $\bar{\mathbf{G}} \in \mathcal{R}^{n\times n}$, in which $\bar{G}_{ij}$ is the prediction score by using the j-th exemplar SVM classifier on the i-th positive training sample, to be low-rank by employing a nuclear norm based regularizer. However, this approach only considers the training data with one type of feature. When the training data are associated with multiple types of features, we demonstrate that it is useful to exploit the relation among multiple types of features based on the consensus principle in Section 5.3.2 or the complementary principle in Section 5.3.3 .

[1]We do not employ the bias term explicitly. Instead, we augment each feature vector with an extra element of 1.

5.3.2 Exemplar-based Multi-view Domain Generalization with Co-regularizer

In this section, we propose our EMVDG_CO method by taking advantage of multi-view features based on the consensus principle, in which an exemplar SVM classifier is learnt for each positive sample on each view. Specifically, we use $f_i^v(\mathbf{x}^v) = \mathbf{w}_i^{v\prime}\mathbf{x}^v$ to denote the exemplar SVM classifier learnt based on $\mathbf{x}_i^{v+}$ and $\{\mathbf{x}_j^{v-}|_{j=1}^m\}$ on the v-th view. We also use $\mathbf{W}^v = [\mathbf{w}_1^v, \ldots, \mathbf{w}_n^v]$ to denote the weight matrix consisting of all the exemplar SVM classifiers learnt on the v-th view.

5.3.2.1 Formulation

Since the positive samples belonging to the same hidden latent domain should be similar, so their corresponding exemplar SVM classifiers should also be similar to each other, and thus the weight vectors $\mathbf{w}_i^v$'s on each view can be grouped into multiple clusters. In this chapter, such cluster structure is exploited by utilizing the low-rank representation (LRR) [103] technique. According to LRR [103], the weight matrix on each view can be reconstructed by using itself as a dictionary, *i.e.*, $\mathbf{W}^v = \mathbf{W}^v\mathbf{Z}^v + \mathbf{E}^v$, in which $\mathbf{Z}^v \in \mathbb{R}^{n\times n}$ is the representation matrix and $\mathbf{E}^v$ is the reconstruction error. Note that the representation matrix $\mathbf{Z}^v$ encodes the cluster structure of exemplar SVM classifiers [103], in which the between-cluster (*resp.*, within-cluster) entries of $\mathbf{Z}^v$ are generally sparse (*resp.*, dense).

On one hand, in LRR, the representation matrices $\mathbf{Z}^v$'s are expected to be low-rank. Moreover, by jointly learning $\mathbf{W}^v$ and low-rank matrix $\mathbf{Z}^v$ using $\mathbf{W}^v = \mathbf{W}^v\mathbf{Z}^v + \mathbf{E}^v$, $\mathbf{W}^v$ is also expected to be low-rank when the error term $\mathbf{E}^v$ is close to zero. In such a case, the weight vectors $\mathbf{w}_i^v$'s corresponding to the positive training samples belonging to the same hidden latent domain are expected to be similar, which is consistent with our motivation.

On the other hand, when the training data are associated with multiple types of features, the cluster structures of $\mathbf{W}^v$'s on different views are expected to be consistent according to the consensus principle. Based on our low-rank representation (LRR), in which the cluster structure of $\mathbf{W}^v$ is encoded in $\mathbf{Z}^v$, such consistency can be easily

introduced by enforcing $\mathbf{Z}^v$'s on multiple views to be close to each other based on our new co-regularizer $\sum_{v,\tilde{v}:v\neq\tilde{v}} \|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2$.

To this end, we formulate our EMVDG_CO method as,

$$\begin{aligned}
\min_{\substack{\mathbf{Z}^v,\mathbf{W}^v,\mathbf{E}^v \\ \xi_i^v,\epsilon_{ij}^v}} \quad & \sum_{v=1}^{V}\left(\frac{1}{2}\|\mathbf{W}^v\|_F^2 + C\sum_{i=1}^{n}\xi_i^v + C\sum_{i=1}^{n}\sum_{j=1}^{m}\epsilon_{ij}^v\right) \\
& + \sum_{v=1}^{V}\left(\lambda_1\|\mathbf{E}^v\|_F^2 + \lambda_2\|\mathbf{Z}^v\|_*\right) + \frac{\gamma}{2}\sum_{v,\tilde{v}:v\neq\tilde{v}}\|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2 \qquad (5.2) \\
\text{s.t.} \quad & \mathbf{w}_i^{v\prime}\mathbf{x}_i^{v+} \geq 1 - \xi_i^v, \quad \xi_i^v \geq 0, \quad \forall v, \forall i, \qquad (5.3) \\
& \mathbf{w}_i^{v\prime}\mathbf{x}_j^{v-} \leq -1 + \epsilon_{ij}^v, \quad \epsilon_{ij}^v \geq 0, \quad \forall v, \forall i, \forall j, \qquad (5.4) \\
& \mathbf{W}^v = \mathbf{W}^v\mathbf{Z}^v + \mathbf{E}^v, \quad \forall v, \qquad (5.5)
\end{aligned}$$

where ξ_i^v, ϵ_{ij}^v are the slack variables, $\|\mathbf{W}^v\|_F^2$ is the regularizer for controlling the complexity of exemplar SVM classifiers, and C, λ_1, λ_2, and γ are the trade-off parameters. The nuclear norm based regularizer $\|\mathbf{Z}^v\|_*$ is used to enforce $\mathbf{Z}^v$ to be low-rank, and the regularizer $\|\mathbf{E}^v\|_F^2$ is employed to enforce the reconstruction error $\mathbf{E}^v$ to approach zeros. Note hat $\mathbf{Z}^v$ cannot be an identity matrix, otherwise $\mathbf{Z}^v$ will be full-rank instead of low-rank, which is against our motivation.

5.3.2.2 Optimization

For better optimizing the problem in (5.2), an intermediate variable $\mathbf{G}^v$ is introduced for each $\mathbf{W}^v$. Instead of employing LRR on $\mathbf{W}^v$ as in (5.2), we employ LRR on $\mathbf{G}^v$ while enforcing $\mathbf{G}^v$ to be close to $\mathbf{W}^v$ by adding the regularizer $\|\mathbf{W}^v - \mathbf{G}^v\|_F^2$. In particular,

we reach the following formulation:

$$\begin{aligned}
\min_{\substack{\mathbf{Z}^v,\mathbf{W}^v,\mathbf{G}^v\\ \mathbf{E}^v,\xi_i^v,\epsilon_{ij}^v}} \quad & \sum_{v=1}^{V}\left(\frac{1}{2}\|\mathbf{W}^v\|_F^2 + C\sum_{i=1}^{n}\xi_i^v + C\sum_{i=1}^{n}\sum_{j=1}^{m}\epsilon_{ij}^v\right) \\
& + \sum_{v=1}^{V}\left(\lambda_1\|\mathbf{W}^v - \mathbf{G}^v\|_F^2 + \lambda_1\|\mathbf{E}^v\|_F^2 + \lambda_2\|\mathbf{Z}^v\|_*\right) \\
& + \frac{\gamma}{2}\sum_{v,\tilde{v}:v\neq\tilde{v}}\|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2 \qquad (5.6)\\
\text{s.t.} \quad & \mathbf{w}_i^{v\prime}\mathbf{x}_i^{v+} \geq 1 - \xi_i^v, \quad \xi_i^v \geq 0, \quad \forall v, \forall i, \qquad (5.7)\\
& \mathbf{w}_i^{v\prime}\mathbf{x}_j^{v-} \leq -1 + \epsilon_{ij}^v, \quad \epsilon_{ij}^v \geq 0, \quad \forall v, \forall i, \forall j, \qquad (5.8)\\
& \mathbf{G}^v = \mathbf{G}^v\mathbf{Z}^v + \mathbf{E}^v, \quad \forall v, \qquad (5.9)
\end{aligned}$$

in which λ_1 is a trade-off parameter. It is obvious that the problem in (5.6) can reduce to the problem in (5.2) when λ_1 approaches $+\infty$. The problem in (5.6) can be solved by an alternative approach, in which two sets of variables $\{\mathbf{Z}^v, \mathbf{E}^v\}$ and $\{\mathbf{W}^v, \mathbf{G}^v, \xi_i^v, \epsilon_{ij}^v\}$ are updated alternatively until the objective value of (5.6) converges.

Update $\mathbf{Z}^v$ and $\mathbf{E}^v$: When $\mathbf{W}^v$, $\mathbf{G}^v$, ξ_i^v, and ϵ_{ij}^v are fixed, the problem in (5.6) becomes the following problem:

$$\begin{aligned}
\min_{\mathbf{Z}^v,\mathbf{E}^v} \quad & \sum_{v=1}^{V}(\lambda_1\|\mathbf{E}^v\|_F^2 + \lambda_2\|\mathbf{Z}^v\|_*) + \frac{\gamma}{2}\sum_{v,\tilde{v}:v\neq\tilde{v}}\|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2 \qquad (5.10)\\
\text{s.t.} \quad & \mathbf{G}^v = \mathbf{G}^v\mathbf{Z}^v + \mathbf{E}^v, \quad \forall v, \qquad (5.11)
\end{aligned}$$

which can be solved by utilizing inexact augmented Lagrange Multiplier (ALM) method [15]. In particular, we introduce the auxiliary variable $\mathbf{P}^v$ (*resp.*, $\mathbf{Q}^v$) to replace $\mathbf{Z}^v$ in $\|\mathbf{Z}^v\|_*$ (*resp.*, $\mathbf{Z}^v$ in the constraint (5.11)), and arrive at the augmented Lagrangian

function as,

$$\begin{aligned}\mathcal{L} = \quad & \sum_{v=1}^{V}(\lambda_1\|\mathbf{E}^v\|_F^2 + \lambda_2\|\mathbf{P}^v\|_*) + \frac{\gamma}{2}\sum_{v,\tilde{v}:v\neq\tilde{v}}\|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2 \\ & + \sum_{v=1}^{V}\langle\mathbf{S}^v, \mathbf{Z}^v - \mathbf{P}^v\rangle + \sum_{v=1}^{V}\langle\mathbf{T}^v, \mathbf{Z}^v - \mathbf{Q}^v\rangle \\ & + \sum_{v=1}^{V}\langle\mathbf{R}^v, \mathbf{G}^v - \mathbf{G}^v\mathbf{Q}^v - \mathbf{E}^v\rangle + \frac{\mu}{2}\sum_{v=1}^{V}\|\mathbf{Z}^v - \mathbf{P}^v\|_F^2 \\ & + \frac{\mu}{2}\sum_{v=1}^{V}\|\mathbf{Z}^v - \mathbf{Q}^v\|_F^2 + \frac{\mu}{2}\sum_{v=1}^{V}\|\mathbf{G}^v - \mathbf{G}^v\mathbf{Q}^v - \mathbf{E}^v\|_F^2,\end{aligned} \tag{5.12}$$

where $\mu > 0$ is a penalty parameter, $\mathbf{S}^v$, $\mathbf{T}^v$, and $\mathbf{R}^v$ are the Lagrangian multipliers. The objective function in (5.12) can be minimized by using the inexact ALM approach, *i.e.*, updating the variables $\{\mathbf{P}^v, \mathbf{Q}^v, \mathbf{Z}^v, \mathbf{E}^v\}$'s, the Lagrangian multipliers $\{\mathbf{S}^v, \mathbf{T}^v, \mathbf{R}^v\}$'s, and the penalty parameter μ in the augmented Lagrangian function (5.12) iteratively until the termination criterion is met. In the following, we will describe how to update $\mathbf{P}^v$, $\mathbf{Q}^v$, $\mathbf{Z}^v$, and $\mathbf{E}^v$ when fixing other variables one by one while the methods for updating $\mathbf{S}^v$, $\mathbf{T}^v$, $\mathbf{R}^v$, and μ are trivial and can be directly found in Algorithm 7.

When fixing the other variables, the subproblem for updating $\{\mathbf{P}^v|_{v=1}^V\}$ is independent *w.r.t.* each $\mathbf{P}^v$, so we solve each $\mathbf{P}^v$ separately. After omitting and adding some constants, we reach the objective function *w.r.t.* $\mathbf{P}^v$ as $\mathbf{P}^v = \arg\min_{\mathbf{P}^v} \lambda_2\|\mathbf{P}^v\|_* + \frac{\mu}{2}\|\mathbf{P}^v - (\mathbf{Z}^v + \frac{\mathbf{S}^v}{\mu})\|_F^2$, which can be solved by employing the Singular Value Threshold (SVT) algorithm [18].

When fixing the other variables, the subproblem for updating $\{\mathbf{Q}^v|_{v=1}^V\}$ is independent *w.r.t.* each $\mathbf{Q}^v$, so we solve each $\mathbf{Q}^v$ separately. By setting the derivative of the subproblem *w.r.t.* $\mathbf{Q}^v$ to zeros, we can easily obtain the solution to $\mathbf{Q}^v$ as $\mathbf{Q}^v = (\mathbf{I} + \mathbf{G}^{v\prime}\mathbf{G}^v)^{-1}(\mathbf{G}^{v\prime}(\mathbf{G}^v - \mathbf{E}^v + \frac{\mathbf{R}^v}{\mu}) + \mathbf{Z}^v + \frac{\mathbf{T}^v}{\mu})$.

When fixing the other variables, the subproblem for updating $\{\mathbf{Z}^v|_{v=1}^V\}$ can be rewritten as $\min_{\mathbf{Z}^v} \frac{\gamma}{2}\sum_{v,\tilde{v}:v\neq\tilde{v}}\|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2 + \sum_{v=1}^V \mu\|\mathbf{Z}^v - \mathbf{H}^v\|_F^2$ with $\mathbf{H}^v = \frac{1}{2}(\mathbf{P}^v + \mathbf{Q}^v - \frac{1}{\mu}(\mathbf{S}^v + \mathbf{T}^v))$, which has a close-form solution based on the vectorization of $\mathbf{Z}^v$.

When fixing the other variables, the subproblem for updating $\{\mathbf{E}^v|_{v=1}^V\}$ is independent *w.r.t.* each $\mathbf{E}^v$, so we solve each $\mathbf{E}^v$ separately. By setting the derivative of the subproblem *w.r.t.* $\mathbf{E}^v$ to zeros, the solution to $\mathbf{E}^v$ can be easily obtained as $\mathbf{E}^v = \frac{\mu(\mathbf{G}^v - \mathbf{G}^v\mathbf{Q}^v) + \mathbf{R}^v}{2\lambda_1 + \mu}$. The steps to solve (5.12) are summarized in Algorithm 7.

Update $\mathbf{W}^v$, $\mathbf{G}^v$, ξ_i^v, ϵ_{ij}^v: When fixing $\mathbf{Z}^v$, we equivalently replace $\mathbf{E}^v$ by $\mathbf{G}^v - \mathbf{G}^v\mathbf{Z}^v$ and rewrite the problem in (5.6) as,

$$\begin{aligned}
\min_{\substack{\mathbf{W}^v,\mathbf{G}^v\\ \xi_i^v,\epsilon_{ij}^v}} \quad & \sum_{v=1}^{V}(\frac{1}{2}\|\mathbf{W}^v\|_F^2 + C\sum_{i=1}^{n}\xi_i^v + C\sum_{i=1}^{n}\sum_{j=1}^{m}\epsilon_{ij}^v \qquad (5.13)\\
& +\lambda_1\|\mathbf{W}^v - \mathbf{G}^v\|_F^2 + \lambda_1\|\mathbf{G}^v - \mathbf{G}^v\mathbf{Z}^v\|_F^2)\\
\text{s.t.} \quad & \mathbf{w}_i^{v\prime}\mathbf{x}_i^{v+} \geq 1 - \xi_i^v, \quad \xi_i^v \geq 0, \quad \forall v, \forall i, \qquad (5.14)\\
& \mathbf{w}_i^{v\prime}\mathbf{x}_j^{v-} \leq -1 + \epsilon_{ij}^v, \quad \epsilon_{ij}^v \geq 0, \quad \forall v, \forall i, \forall j. \qquad (5.15)
\end{aligned}$$

It can be observed that the above problem contains V independent subproblems corresponding to V views. So we solve each subproblem by alternatively updating two sets of variables $\{\mathbf{W}^v, \xi_i^v, \epsilon_{ij}^v\}$ and $\mathbf{G}^v$ until the objective value of (5.13) converges. In particular, when fixing $\mathbf{G}^v$, the problem *w.r.t.* $\mathbf{W}^v$, ξ_i^v, and ϵ_{ij}^v can be separated into n independent subproblems with each related to one exemplar SVM classifier. Thus, we have the following subproblem *w.r.t.* the i-th exemplar SVM classifier:

$$\begin{aligned}
\min_{\mathbf{w}_i^v,\xi_i^v,\epsilon_{ij}^v} \quad & \frac{1}{2}\|\mathbf{w}_i^v\|^2 + C(\xi_i^v + \sum_{j=1}^{m}\epsilon_{ij}^v) + \lambda_1\|\mathbf{w}_i^v - \mathbf{g}_i^v\|^2 \qquad (5.16)\\
\text{s.t.} \quad & \mathbf{w}_i^{v\prime}\mathbf{x}_i^{v+} \geq 1 - \xi_i^v, \quad \xi_i^v \geq 0, \qquad (5.17)\\
& \mathbf{w}_i^{v\prime}\mathbf{x}_j^{v-} \leq -1 + \epsilon_{ij}^v, \quad \epsilon_{ij}^v \geq 0, \quad \forall j, \qquad (5.18)
\end{aligned}$$

where $\mathbf{g}_i^v$ is the i-th column vector of $\mathbf{G}^v$. We introduce the dual variables $\{\hat{\alpha}^+, \hat{\beta}^+\}$ and $\{\hat{\alpha}_j^-, \hat{\beta}_j^-\}$'s for the constraints in (5.17) and (5.18) respectively, and obtain the dual form of (5.16) as,

$$\begin{aligned}
\min_{\hat{\boldsymbol{\alpha}}} \quad & \hat{\boldsymbol{\alpha}}'\frac{\mathbf{K}_i^v \circ (\mathbf{y}\mathbf{y}')}{2(1+2\lambda_1)}\hat{\boldsymbol{\alpha}} + [\frac{2\lambda_1(\mathbf{X}_i^{v\prime}\mathbf{g}_i^v)\circ\mathbf{y}}{1+2\lambda_1} - \mathbf{1}]'\hat{\boldsymbol{\alpha}} \qquad (5.19)\\
\text{s.t.} \quad & \mathbf{0} \leq \hat{\boldsymbol{\alpha}} \leq C\mathbf{1},
\end{aligned}$$

where $\mathbf{X}_i^v = [\mathbf{x}_i^{v+}, \mathbf{x}_1^{v-}, \ldots, \mathbf{x}_m^{v-}]$, $\mathbf{K}_i^v = \mathbf{X}_i^{v\prime}\mathbf{X}_i^v$, $\hat{\boldsymbol{\alpha}} = [\hat{\alpha}^+, \hat{\alpha}_1^-, \ldots, \hat{\alpha}_m^-]'$, and $\mathbf{y} = [1, -\mathbf{1}_m']'$. The problem in (5.19) is a quadratic programming (QP) problem, which can be solved efficiently by using the SMO algorithm [126], *i.e.*, updating one selected dual variable in each iteration. With obtained $\hat{\boldsymbol{\alpha}}$, $\mathbf{w}_i^v$ can be recovered by using the following equation:

$$\mathbf{w}_i^v = \frac{1}{1+2\lambda_1}(2\lambda_1\mathbf{g}_i^v + \mathbf{X}_i^v(\mathbf{y}\circ\hat{\boldsymbol{\alpha}})). \qquad (5.20)$$

Algorithm 7 Solving (5.12) with inexact ALM

Input: $\mathbf{G}^v, \lambda_1, \lambda_2, \gamma$

Initialize $\mathbf{Z}^v = \mathbf{E}^v = \mathbf{S}^v = \mathbf{T}^v = \mathbf{R}^v = \mathbf{O}$, $\rho = 0.1$, $\mu = 0.1$, $\mu_{max} = 10^6$, $\nu = 10^{-5}$, $N_{iter} = 10^6$.

for $t = 1 : N_{iter}$ **do**

  $\forall v$, update $\mathbf{P}^v$ by solving $\mathbf{P}^v = \arg\min_{\mathbf{P}^v} \lambda_2 \|\mathbf{P}^v\|_* + \frac{\mu}{2}\|\mathbf{P}^v - (\mathbf{Z}^v + \frac{\mathbf{S}^v}{\mu})\|_F^2$.

  $\forall v$, update $\mathbf{Q}^v$ by $\mathbf{Q}^v = (\mathbf{I} + \mathbf{G}^{v\prime}\mathbf{G}^v)^{-1}(\mathbf{G}^{v\prime}(\mathbf{G}^v - \mathbf{E}^v + \frac{\mathbf{R}^v}{\mu}) + \mathbf{Z}^v + \frac{\mathbf{T}^v}{\mu})$.

  $\forall v$, update $\mathbf{Z}^v$ by solving $\min_{\mathbf{Z}^v} \frac{\gamma}{2}\sum_{v,\tilde{v}:v\neq\tilde{v}} \|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2 + \sum_{v=1}^V \mu\|\mathbf{Z}^v - \mathbf{H}^v\|_F^2$, where $\mathbf{H}^v = \frac{1}{2}(\mathbf{P}^v + \mathbf{Q}^v - \frac{1}{\mu}(\mathbf{S}^v + \mathbf{T}^v))$.

  $\forall v$, update $\mathbf{E}^v$ by $\mathbf{E}^v = \frac{\mu(\mathbf{G}^v - \mathbf{G}^v\mathbf{Q}^v) + \mathbf{R}^v}{2\lambda_1 + \mu}$.

  $\forall v$, update $\mathbf{S}^v$, $\mathbf{T}^v$, and $\mathbf{R}^v$ by $\mathbf{S}^v = \mathbf{S}^v + \mu(\mathbf{Z}^v - \mathbf{P}^v)$, $\mathbf{T}^v = \mathbf{T}^v + \mu(\mathbf{Z}^v - \mathbf{Q}^v)$, $\mathbf{R}^v = \mathbf{R}^v + \mu(\mathbf{G}^v - \mathbf{G}^v\mathbf{Q}^v - \mathbf{E}^v)$.

  Update the parameter μ by $\mu = \min(\mu_{max}, (1+\rho)\mu)$.

  Break if $\|\mathbf{G}^v - \mathbf{G}^v\mathbf{Q}^v - \mathbf{E}^v\|_\infty < \nu$, $\|\mathbf{Z}^v - \mathbf{P}^v\|_\infty < \nu$, $\|\mathbf{Z}^v - \mathbf{Q}^v\|_\infty < \nu$, $\quad \forall v$.

end for

Output: $\mathbf{Z}^v$.

When $\mathbf{W}^v$, ξ_i^v, and ϵ_{ij}^v are fixed, we have a close-form solution for updating $\mathbf{G}^v$. In particular, by setting the derivative of (5.13) *w.r.t.* $\mathbf{G}^v$ to zeros, we can easily obtain the updating equation of $\mathbf{G}^v$ as,

$$\mathbf{G}^v = \lambda_1 \mathbf{W}^v \left(\lambda_1(\mathbf{I} - \mathbf{Z}^v)(\mathbf{I} - \mathbf{Z}^v)' + \lambda_1 \mathbf{I}\right)^{-1}. \tag{5.21}$$

The whole algorithm is listed in Algorithm 8.

Time Complexity Analysis: In Algorithm 8, the most time-consuming steps are updating $\mathbf{P}^v$'s and $\mathbf{Q}^v$'s. When updating $\mathbf{P}^v$ or $\mathbf{Q}^v$ on each view, the time complexity is $O(n^3)$ with n being the number of positive training samples due to SVD or matrix inverse operation. Assume inexact ALM converges in T iterations, then the time complexity for the whole algorithm is $O(TVn^3)$.

During the testing procedure, inspired by the prediction method in [164], given a test sample, we average the higher prediction scores of this sample obtained by using the

Algorithm 8 Exemplar-based Multi-view Domain Generalization with Co-regularizer

Require: Training data $\{\mathbf{x}_i^{v+}|_{i=1}^n\}$ and $\{\mathbf{x}_j^{v-}|_{j=1}^m\}$ with V views.

1: Initialize[2] $\mathbf{G}^v$'s.
2: **repeat**
3: Use Algorithm 7 to update $\mathbf{Z}^v$'s.
4: **repeat**
5: Solve n independent subproblems in the dual form (5.19) and then recover $\mathbf{W}^v$ using (5.20) on each view.
6: Update $\mathbf{G}^v$ by using (5.21) on each view.
7: **until** The objective function of (5.13) converges.
8: **until** The objective function of (5.6) converges.

Ensure: The learnt classifier $\mathbf{W}^v$'s.

exemplar classifiers on each view. By representing each test sample as $\mathbf{u} = (\mathbf{u}^1, \ldots, \mathbf{u}^V)$ with $\mathbf{u}^v$ being the v-th view feature, we formulate the final prediction score of $\mathbf{u}$ as,

$$f(\mathbf{u}) = \frac{1}{V}\sum_{v=1}^{V}\frac{1}{|\Gamma(\mathbf{u}^v)|}\sum_{i:i\in\Gamma(\mathbf{u}^v)} f_i^v(\mathbf{u}^v), \tag{5.22}$$

where $f_i^v(\mathbf{u}^v)$ is the prediction score of $\mathbf{u}^v$ by using the i-th exemplar SVM classifier $\mathbf{w}_i^v$, and $\Gamma(\mathbf{u}^v)$ is the index set of exemplar SVM classifiers which obtain the top prediction scores on $\mathbf{u}^v$. Following [164], the cardinality of $\Gamma(\mathbf{u}^v)$ (*i.e.*, $|\Gamma(\mathbf{u}^v)|$) is set as 5 in our experiments. By using this prediction method, we conjecture that this test sample is predicted by the exemplar SVM classifiers learnt based on the positive training samples which may come from the most relevant hidden latent domain. Consequently, the integrated classifier $f(\mathbf{u})$ in (5.22) is expected to generalize well to arbitrary target domain.

[2]We initialize $\mathbf{G}^v$ by using the weight vector of the exemplar classifier learnt based on the i-th positive sample and all the negative samples on the v-th view as its i-th column vector.

5.3.3 Exemplar-based Multi-view Domain Generalization Based on MKL

Inspired by multiple kernel learning (MKL) [6], in this section we propose our EMVDG_MK approach by exploiting multi-view features based on the complementary principle, . Specifically, in our multi-view scenario, multiple types of features may have complementary information, and thus it is beneficial to fuse the classifiers learnt on different views. By treating each view as a kernel, our problem can be considered as a multiple kernel learning (MKL) problem.

5.3.3.1 Formulation

Inspired by [6], we first write the primal form of multiple kernel learning (MKL) based on hard-margin[3] SVM with V-view features as,

$$\min_{\mathbf{d},\mathbf{w}} \quad \sum_{v=1}^{V} \frac{\|\mathbf{w}^v\|^2}{d_v} \tag{5.23}$$

$$\text{s.t.} \quad \tilde{y}_i \sum_{v=1}^{V} \mathbf{w}^{v'} \mathbf{x}_i^v \geq 1, \quad \forall i, \tag{5.24}$$

$$\mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0},$$

where $\mathbf{d} = [d_1, \ldots, d_V]'$, $\tilde{y}_i$ is the label of the i-th training sample, $\mathbf{x}_i^v$ is the v-th type of feature of the i-th training sample, and $\mathbf{w}^v$ is the SVM classifier on the v-th view. From (5.23) we can observe that the SVM classifiers $\mathbf{w}^v$'s on different views are integrated based on the complementary principle.

By introducing dual variables α_i's for the constraints in (5.24) and setting the derivative of the Lagrangian form *w.r.t.* each $\mathbf{w}^v$ to zeros, we can easily obtain the following equation:

$$\mathbf{w}^v = d_v \mathbf{X}^v (\boldsymbol{\alpha} \circ \tilde{\mathbf{y}}), \quad \forall v, \tag{5.25}$$

[3]Our formulation can be similarly derived when using soft-margin SVM. Here we use parameter-free hard-margin SVM for simplicity. Moreover, we do not employ the bias term explicitly. Instead, we augment each feature vector with an extra element of 1.

where $\mathbf{X}^v = [\mathbf{x}_1^v, \ldots, \mathbf{x}_{\tilde{n}}^v]$ with $\tilde{n}$ being the number of training samples, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{\tilde{n}}]'$, and $\tilde{\mathbf{y}} = [\tilde{y}_1, \ldots, \tilde{y}_{\tilde{n}}]'$. By substituting (5.25) back into the Lagrangian form of (5.23), we can obtain the dual form of (5.23) as the following min-max optimization problem:

$$\begin{aligned} \min_{\mathbf{d}} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}'(\mathbf{K}^v \circ (\tilde{\mathbf{y}}\tilde{\mathbf{y}}'))\boldsymbol{\alpha} + \mathbf{1}'\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{\alpha} \geq \mathbf{0}, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}, \end{aligned} \tag{5.26}$$

where $\mathbf{K}^v = \mathbf{X}^{v'}\mathbf{X}^v$ is the kernel matrix on the v-th view. From the dual form in (5.26), we can observe that multiple kernels on different views are linearly combined with the coefficient $\mathbf{d}$ based on the complementary principle.

In (5.25), V SVM classifiers share the same dual vector $\boldsymbol{\alpha}$. So in this chapter, n dual vectors should be used because we need to train n exemplar SVM classifiers on each view and the exemplar SVM classifiers corresponding to the same positive sample on different views share the same dual vector. By using $\boldsymbol{\alpha}_i$ to denote the dual vector of the exemplar SVM classifiers corresponding to the i-th positive training sample, we can formulate our MKL problem with V views as,

$$\begin{aligned} \min_{\mathbf{d}} \max_{\boldsymbol{\alpha}_i} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i'\mathbf{M}_i^v\boldsymbol{\alpha}_i + \sum_{i=1}^{n}\mathbf{1}'\boldsymbol{\alpha}_i \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}, \end{aligned} \tag{5.27}$$

in which $\mathbf{d}$ is the same as defined in the paragraph below (5.23), and $\mathbf{M}_i^v = \mathbf{K}_i^v \circ (\mathbf{y}\mathbf{y}')$ with $\mathbf{y}$ and $\mathbf{K}_i^v$ being the same as defined in the paragraph below (5.19).

Recall that the positive training samples are likely to come from multiple hidden latent domains. When the j-th positive training sample and the k-th training sample come from the same latent domain, $\mathbf{X}_j^v$ and $\mathbf{X}_k^v$ should be similar, and the weight vectors of their corresponding exemplar SVM classifiers (*i.e.*, $\mathbf{w}_j^v$ and $\mathbf{w}_k^v$) should also be similar as discussed in Section 5.3.2. Moreover, similar as (5.25), we can easily derive that $\mathbf{w}_i^v = d_v\mathbf{X}_i^v(\boldsymbol{\alpha}_i \circ \mathbf{y})$, based on which we can infer that the dual vectors $\boldsymbol{\alpha}_j$ and $\boldsymbol{\alpha}_k$ should be similar when $\mathbf{w}_j^v$ is similar to $\mathbf{w}_k^v$ and $\mathbf{X}_j^v$ is similar to $\mathbf{X}_k^v$. Based on the

Algorithm 9 Exemplar-based Multi-view Domain Generalization Based on MKL

Require: Training data $\{\mathbf{x}_i^{v+}|_{i=1}^n\}$ and $\{\mathbf{x}_j^{v-}|_{j=1}^m\}$ with V views.

1: Initialize[4] $\mathbf{A}$, $\mathbf{d} = \frac{1}{V}\mathbf{1}$.

2: **repeat**

3: Update $\mathbf{B}$ by solving the problem in (5.31).

4: **repeat**

5: Update $\boldsymbol{\alpha}_i$'s by solving n independent subproblems in the inner problem of (5.32) and then recover $\mathbf{w}_i^v$'s by using (7.33) on each view.

6: Update $\mathbf{d}$ by using (5.37).

7: **until** The objective function of (5.32) converges.

8: **until** The objective function of (5.30) converges.

Ensure: The learnt classifier $\mathbf{W}^v$'s.

above discussions, the dual vectors $\boldsymbol{\alpha}_i$'s can be organized into multiple hidden clusters. By denoting the dual matrix as $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n] \in \mathcal{R}^{(m+1)\times n}$, we add a nuclear norm based regularizer $\|\mathbf{A}\|_*$ to (5.27) to enforce $\mathbf{A}$ to be low-rank, and arrive at our final formulation:

$$\begin{aligned} \min_{\mathbf{d}} \max_{\mathbf{A}} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i' \mathbf{M}_i^v \boldsymbol{\alpha}_i + \sum_{i=1}^{n} \mathbf{1}'\boldsymbol{\alpha}_i - \zeta\|\mathbf{A}\|_* \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}, \end{aligned} \tag{5.28}$$

in which ζ is a trade-off parameter.

5.3.3.2 Optimization

The problem in (5.28) is not easy to be optimized due to the regularizer $\|\mathbf{A}\|_*$, so we introduce an intermediate variable $\mathbf{B}$ and apply the low-rank regularizer on $\mathbf{B}$ instead of

[4]We initialize $\mathbf{A}$ with its i-th column vector being the dual vector of exemplar classifiers learnt based on the averaged kernel from V views, which are obtained based on the i-th positive sample and all the negative samples.

$\mathbf{A}$ and enforce $\mathbf{B}$ to be close to $\mathbf{A}$. Then, we reach the following formulation,

$$\begin{aligned} \min_{\mathbf{d}} \max_{\mathbf{A},\mathbf{B}} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i' \mathbf{M}_i^v \boldsymbol{\alpha}_i + \sum_{i=1}^{n} \mathbf{1}'\boldsymbol{\alpha}_i - \zeta_1 \|\mathbf{B}\|_* - \frac{\zeta_2}{2}\|\mathbf{A}-\mathbf{B}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}, \end{aligned} \tag{5.29}$$

in which ζ_1 and ζ_2 are two trade-off parameters. It is obvious that the problem in (5.29) can reduce to the problem in (5.28) when ζ_2 approaches $+\infty$. Since the objective function in (5.29) is concave *w.r.t.* $\mathbf{B}$ and convex *w.r.t.* $\mathbf{d}$, so $\min_{\mathbf{d}}$ and $\max_{\mathbf{A}}$ can be exchanged [100]. Then, we can rewrite (5.29) as,

$$\begin{aligned} \max_{\mathbf{B}} \min_{\mathbf{d}} \max_{\mathbf{A}} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i' \mathbf{M}_i^v \boldsymbol{\alpha}_i + \sum_{i=1}^{n} \mathbf{1}'\boldsymbol{\alpha}_i - \zeta_1 \|\mathbf{B}\|_* - \frac{\zeta_2}{2}\|\mathbf{A}-\mathbf{B}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}. \end{aligned} \tag{5.30}$$

Note that the inner problem of (5.30) *w.r.t.* $\mathbf{d}$ and $\mathbf{A}$ can be reformulated as a convex problem, and will be discussed in the proof of Proposition 2. So we solve (5.30) by using an alternating optimization approach. Particularly, we alternatively update two sets of variables $\mathbf{B}$ and $\{\mathbf{A}, \mathbf{d}\}$ until the objective of (5.30) converges.

Update B: When fixing $\mathbf{A}$ and $\mathbf{d}$, the problem in (5.30) reduces to the following problem,

$$\min_{\mathbf{B}} \zeta_1 \|\mathbf{B}\|_* + \frac{\zeta_2}{2}\|\mathbf{A}-\mathbf{B}\|_F^2, \tag{5.31}$$

which can be solved by employing the Singular Value Threshold (SVT) algorithm [18].

Update A, d: When fixing $\mathbf{B}$, the problem in (5.30) can reduce to the following problem,

$$\begin{aligned} \min_{\mathbf{d}} \max_{\boldsymbol{\alpha}_i} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i' \mathbf{M}_i^v \boldsymbol{\alpha}_i + \sum_{i=1}^{n} 1'\boldsymbol{\alpha}_i - \frac{\zeta_2}{2}\sum_{i=1}^{n}\|\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\|^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}, \end{aligned} \tag{5.32}$$

in which β_i is the i-th column of $\mathbf{B}$.

Interestingly, the primal form of (5.32) is closely related to the primal form of MKL in (5.23), which is described below.

Proposition 2 *The primal form of (5.32) can be written as,*

$$\min_{\substack{\mathbf{d},\mathbf{w}_i^v \\ \tilde{\xi}_i,\tilde{\epsilon}_{ij}}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V}\frac{\|\mathbf{w}_i^v\|^2}{d_v} + \frac{1}{2\zeta_2}(\sum_{i=1}^{n}\tilde{\xi}_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{\epsilon}_{ij}^2) \tag{5.33}$$

$$s.t. \quad \sum_{v=1}^{V}\mathbf{w}_i^{v'}\mathbf{x}_i^{v+} \geq (1+\zeta_2\beta_i^+) - \tilde{\xi}_i, \quad \forall i, \tag{5.34}$$

$$\sum_{v=1}^{V}\mathbf{w}_i^{v'}\mathbf{x}_j^{v-} \leq -(1+\zeta_2\beta_{ij}^-) + \tilde{\epsilon}_{ij}, \quad \forall i, \forall j, \tag{5.35}$$

$$\mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}, \tag{5.36}$$

where β_i^+'s and β_{ij}^-'s are newly introduced variables, and $\tilde{\xi}_i$'s and $\tilde{\epsilon}_{ij}$'s are the slack variables. The detailed proof can be found in Appendix C.

The problem in (5.33) is jointly convex *w.r.t.* $\mathbf{d}$, $\mathbf{w}_i^v$'s, $\tilde{\xi}_i$'s, and $\tilde{\epsilon}_{ij}$'s, so the global optimum can be achieved by using an alternative optimization approach. Specifically, when $\mathbf{d}$ is fixed, we solve $\boldsymbol{\alpha}_i$ in the dual form in (5.32) and then recover $\mathbf{w}_i^v$ by using (7.33). The subproblems *w.r.t.* each $\boldsymbol{\alpha}_i$ are independent with each subproblem being a quadratic programming (QP) problem, which can be solved efficiently by using the SMO algorithm [126]. When $\mathbf{w}_i^v$'s are fixed, we introduce a dual variable τ for the constraint $\mathbf{1}'\mathbf{d} = 1$ in (5.36) and set the derivative of the Lagrangian form *w.r.t.* d_v to zero, which leads to $d_v = \sqrt{\frac{\sum_{i=1}^{n}\|\mathbf{w}_i^v\|^2}{2\tau}}$. Considering $\mathbf{1}'\mathbf{d} = 1$ and the equation in (7.33), we can easily obtain the close-form solution for d_v as,

$$d_v = \frac{\sqrt{\sum_{i=1}^{n}\|\mathbf{w}_i^v\|^2}}{\sum_{v=1}^{V}\sqrt{\sum_{i=1}^{n}\|\mathbf{w}_i^v\|^2}} = \frac{\sqrt{\sum_{i=1}^{n}d_v^2\boldsymbol{\alpha}_i'\mathbf{M}_i^v\boldsymbol{\alpha}_i}}{\sum_{v=1}^{V}\sqrt{\sum_{i=1}^{n}d_v^2\boldsymbol{\alpha}_i'\mathbf{M}_i^v\boldsymbol{\alpha}_i}}. \tag{5.37}$$

The whole algorithm of EMVDG_MK is listed in Algorithm 9.

Time Complexity Analysis: In Algorithm 9, the most time-consuming step is to update $\boldsymbol{\alpha}_i$'s. When solving each independent subproblem in the inner problem of (5.32), we use SMO algorithm to solve $\boldsymbol{\alpha}_i$. According to [127], the time complexity of SMO is between $O(m)$ and $O(m^{2.3})$, in which m is the number of negative training samples. Because we have in total n independent subproblems with n being the number of positive training samples, the time complexity of each inner iteration is between $O(nm)$ and $O(nm^{2.3})$. Assume the whole algorithm takes T outer iterations and the average number

of inner iterations in each outer iteration is t, then the time complexity of the whole algorithm is between $O(Ttnm)$ and $O(Ttnm^{2.3})$.

In the testing stage, we use the same prediction method as for EMVDG_CO (see (5.22)) in Section 5.3.2.

5.4 Extending our EMVDG Framework for Domain Adaptation

When we have unlabeled target domain samples in the training stage, our EMVDG framework can be extended to exemplar-based multi-view domain adaptation (EMVDA) by utilizing the unlabeled data for domain adaptation. Specifically, we further add a Laplacian regularizer, such that the prediction scores of target domain samples obtained by using the learnt exemplar SVM classifiers should satisfy the smoothness constraint. This regularizer has proved to be effective for domain adaptation [37]. To be exact, when two target domain samples are similar, their prediction scores obtained by using the same set of exemplar SVM classifiers should be close to each other. We extend our EMVDG_CO and EMVDG_MK methods to EMVDA_CO and EMVDA_MK, respectively.

5.4.1 Exemplar-based Multi-view Domain Adaptation with Co-regularizer

We add a Laplacian regularizer to the objective function of our EMVDG_CO method (*i.e.*, (5.6)) and formulate the objective function of our EMVDA_CO approach as,

$$\begin{aligned}
\min_{\substack{\mathbf{Z}^v,\mathbf{W}^v,\mathbf{G}^v\\ \mathbf{E}^v,\xi_i^v,\epsilon_{ij}^v}} \quad & \sum_{v=1}^{V}\Big(\frac{1}{2}\|\mathbf{W}^v\|_F^2 + C\sum_{i=1}^{n}\xi_i^v + C\sum_{i=1}^{n}\sum_{j=1}^{m}\epsilon_{ij}^v \\
& +\lambda_1\|\mathbf{W}^v-\mathbf{G}^v\|_F^2 + \lambda_1\|\mathbf{E}^v\|_F^2 + \lambda_2\|\mathbf{Z}^v\|_*\Big) \\
& +\frac{\gamma}{2}\sum_{v,\tilde{v}:v\neq\tilde{v}}\|\mathbf{Z}^v-\mathbf{Z}^{\tilde{v}}\|_F^2 + \theta\sum_{v=1}^{V}\Omega(\mathbf{W}^v,\mathbf{L}^v,\mathbf{U}^v) \qquad (5.38)\\
\text{s.t.} \quad & \mathbf{w}_i^{v\prime}\mathbf{x}_i^{v+} \geq 1-\xi_i^v, \quad \xi_i^v \geq 0, \quad \forall v,\forall i, \qquad (5.39)\\
& \mathbf{w}_i^{v\prime}\mathbf{x}_j^{v-} \leq -1+\epsilon_{ij}^v, \quad \epsilon_{ij}^v \geq 0, \quad \forall v,\forall i,\forall j, \qquad (5.40)\\
& \mathbf{G}^v = \mathbf{G}^v\mathbf{Z}^v + \mathbf{E}^v, \quad \forall v, \qquad (5.41)
\end{aligned}$$

where θ is a trade-off parameter, $\Omega(\mathbf{W}^v, \mathbf{L}^v, \mathbf{U}^v) = \mathrm{tr}(\mathbf{W}^{v\prime}\mathbf{U}^v\mathbf{L}^v\mathbf{U}^{v\prime}\mathbf{W}^v)$ is the Laplacian regularizer, in which $\mathbf{U}^v = [\mathbf{u}_1^v, \ldots, \mathbf{u}_N^v]$ is the target domain samples with N being the total number of unlabeled target domain samples and $\mathbf{u}_i^v$ being the v-th type of feature of the i-th target domain sample, $\mathbf{L}^v$ is the Laplacian matrix constructed based on the target domain samples on the v-th view. Note that we use the nearest neighbor graph to construct the Laplacian matrices $\mathbf{L}^v$'s based on cosine similarity as suggested in [167].

We can solve the problem in (5.38) similarly to that for solving (5.6). The only difference lies in that when updating $\mathbf{W}^v$ on the v-th view, compared with (5.16), the subproblem *w.r.t.* the i-th exemplar classifier has an additional Laplacian regularizer, which is written as,

$$\begin{aligned}
\min_{\mathbf{w}_i^v, \xi_i^v, \epsilon_{ij}^v} \quad & \frac{1}{2}\|\mathbf{w}_i^v\|^2 + C(\xi_i^v + \sum_{j=1}^{m} \epsilon_{ij}^v) + \lambda_1 \|\mathbf{w}_i^v - \mathbf{g}_i^v\|^2 \\
& + \theta \mathbf{w}_i^{v\prime}\mathbf{U}^v\mathbf{L}^v\mathbf{U}^{v\prime}\mathbf{w}_i^v \qquad (5.42) \\
\text{s.t.} \quad & \mathbf{w}_i^{v\prime}\mathbf{x}_i^{v+} \geq 1 - \xi_i^v, \quad \xi_i^v \geq 0, . \qquad (5.43) \\
& \mathbf{w}_i^{v\prime}\mathbf{x}_j^{v-} \leq -1 + \epsilon_{ij}^v, \quad \epsilon_{ij}^v \geq 0, \quad \forall j, \qquad (5.44)
\end{aligned}$$

which can also be solved in the dual form by using the SMO algorithm [126].

5.4.2 Exemplar-based Multi-view Domain Adaptation Based on MKL

Similar to Section 5.4.1, we also add a Laplacian regularizer to the objective function of our EMVDG_MK method (*i.e.*, (5.28)). Recall that $\mathbf{w}_i^v = d_v \mathbf{X}_i^v(\boldsymbol{\alpha}_i \circ \mathbf{y})$ (see (7.33)), so we can derive the Laplacian regularizer $\Omega(\mathbf{W}^v, \mathbf{L}^v, \mathbf{U}^v) = \mathrm{tr}(\mathbf{W}^{v\prime}\mathbf{U}^v\mathbf{L}^v\mathbf{U}^{v\prime}\mathbf{W}^v) = d_v^2 \sum_{i=1}^n \boldsymbol{\alpha}_i'(\mathbf{X}_i^{v\prime}\mathbf{U}^v\mathbf{L}^v\mathbf{U}^{v\prime}\mathbf{X}_i^v \circ (\mathbf{y}\mathbf{y}'))\boldsymbol{\alpha}_i$. Similar to the regularizer $\|\mathbf{w}_i^v\|^2$ in (5.33), we assign the weight $\frac{1}{d_v}$ to the Laplacian regularizer on the v-th view. After denoting $\hat{\mathbf{K}}_i^v = \mathbf{X}_i^{v\prime}\mathbf{U}^v$ and adding the weighted Laplacian regularizer to (5.28), we formulate

our EMVDA_MK method as,

$$\begin{aligned}\min_{\mathbf{d}}\max_{\mathbf{A}} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i' \mathbf{M}_i^v \boldsymbol{\alpha}_i + \sum_{i=1}^{n} \mathbf{1}'\boldsymbol{\alpha}_i - \zeta\|\mathbf{A}\|_* \\ & -\frac{\vartheta}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i' (\hat{\mathbf{K}}_i^v \mathbf{L}^v \hat{\mathbf{K}}_i^{v'} \circ (\mathbf{y}\mathbf{y}'))\boldsymbol{\alpha}_i \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0},\end{aligned} \tag{5.45}$$

where ϑ is a trade-off parameter. After denoting $\hat{\mathbf{M}}_i^v = (\mathbf{K}_i^v + \vartheta \hat{\mathbf{K}}_i^v \mathbf{L}^v \hat{\mathbf{K}}_i^{v'}) \circ (\mathbf{y}\mathbf{y}')$, we can simplify (5.45) as,

$$\begin{aligned}\min_{\mathbf{d}}\max_{\mathbf{A}} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i' \hat{\mathbf{M}}_i^v \boldsymbol{\alpha}_i + \sum_{i=1}^{n} \mathbf{1}'\boldsymbol{\alpha}_i - \zeta\|\mathbf{A}\|_* \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0},\end{aligned} \tag{5.46}$$

which shares a similar form with (5.28) except that we replace $\mathbf{M}_i^v$ by $\hat{\mathbf{M}}_i^v$. So the algorithm for solving (5.46) is similar to that for solving (5.28). The only difference lies in that when fixing $\mathbf{B}$ and updating $\{\mathbf{A}, \mathbf{d}\}$, the primal form of the subproblem can be written as,

$$\begin{aligned}\min_{\substack{\mathbf{d},\mathbf{w}_i^v \\ \tilde{\xi}_i,\tilde{\epsilon}_{ij}}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V}\frac{\|\mathbf{w}_i^v\|^2}{d_v} + \frac{1}{2\zeta_2}(\sum_{i=1}^{n}\tilde{\xi}_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{\epsilon}_{ij}^2) \\ \text{s.t.} \quad & \sum_{v=1}^{V}\mathbf{w}_i^{v'}\psi(\mathbf{x}_i^{v+}) \geq (1+\zeta_2\beta_i^+) - \tilde{\xi}_i, \quad \forall i, \\ & \sum_{v=1}^{V}\mathbf{w}_i^{v'}\psi(\mathbf{x}_j^{v-}) \leq -(1+\zeta_2\beta_{ij}^-) + \tilde{\epsilon}_{ij}, \quad \forall i, \forall j, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0},\end{aligned} \tag{5.47}$$

where $\psi(\cdot)$ is the feature mapping function induced by the kernel $(\mathbf{K}_i^v + \vartheta \hat{\mathbf{K}}_i^v \mathbf{L}^v \hat{\mathbf{K}}_i^{v'})$. The problem in (5.47) shares a similar form with (5.33) except that we apply the feature mapping function $\psi(\cdot)$ on $\mathbf{x}_i^{v+}$'s and $\mathbf{x}_j^{v-}$'s. So when updating $\mathbf{A}$ (*resp.*, $\mathbf{d}$), we replace $\mathbf{M}_i^v$ in (5.32) (*resp.*, (5.37)) by $\hat{\mathbf{M}}_i^v$.

5.5 Experiments

In this section, the effectiveness of our EMVDG and EMVDA frameworks for human action recognition and object recognition is demonstrated by extensive experiments on three benchmark datasets. In particular, we show that our EMVDG (*resp.*, EMVDA) framework outperforms all the state-of-the-art baselines in Section 5.5.1 (*resp.*, Section 5.5.2). We also provide the insightful analysis on why our two methods under the EMVDG framework are effective. Moreover, we take the Office-Caltech dataset as an example to show that the performance can be further improved by using more types of features in Section 5.5.3.

5.5.1 Domain Generalization

Experimental Settings: All methods are evaluated for the human action recognition task on two benchmark datasets: ACT4^2 [26] and Online RGBD Action Dataset (ORGBD) [168].

The ACT4^2 dataset consists of 2648 RGB-D videos from 14 action categories, which are captured from 4 camera viewpoints. As suggested in [26], the samples captured from each camera viewpoint are treated as one domain. Then, the videos from 2 domains and the remaining 2 domains are merged as the source domain and the target domain respectively, which leads to in total 6 settings.

The Online RGBD Action Dataset (ORGBD) [168] contains the RGB-D videos from 7 action categories. This dataset has 3 sets with each set containing 112 videos, in which Set 3 is captured in one environment while Set 1 and Set 2 are captured in another environment. In order to evaluate all methods for cross-environment human action recognition, two sets captured in different environments are merged as the source domain and the remaining one is treated as the target domain. Thus, we have a total of 2 settings, that is, Set 1 and 3 (*resp.*, Set 2 and 3) for training and Set 2 (*resp.*, Set 1) for testing.

For human action recognition on the ACT4^2 and ORGBD datasets, two types of features (*i.e.*, RGB and depth) are used in the experiments. In particular, for each pair of RGB and depth videos in both ACT4^2 and ORGBD datasets, we extract the improved dense trajectory (IDT) descriptors [156]. Compared with the preliminary conference

version of this paper [116], we use Fisher vector encoding method instead of Bag-of-Word (BOW) to encode the IDT descriptors. Specifically, following [2], we train 256 Gaussian Mixture Models (GMMs) based on the IDT descriptors from the training videos, and then extract a 109,056-dim Fisher vector for each training and test video. Finally, we perform PCA to reduce the dimension of Fisher vectors to 10000.

Moreover, all methods are also evaluated for the object recognition task on the benchmark dataset Office-Caltech [60]. The images in the Office-Caltech dataset are from 4 domains, that is, Caltech-256 (C), Amazon (A), Webcam (W), and Digital SLR (D). Following the experiemtal setting in [60], the 10 common categories among the 4 domains are used, which consists of a total of 2533 images. As suggested in [59, 164], we mix D and W (*resp.*, C, D, and W; A and C) as the source domain and the remaining domains are used as the target domain, which leads to 3 experimental settings in total. For each image, we extract the 4096-dim DeCAF_6 feature [38] and the 4096-dim Caffe_6 [77] feature as two-view features.

Baselines: We compare EMVDG_CO and EMVDG_MK methods with two basic baselines, *i.e.*, SVM [29] and exemplar SVM (ESVM) [108], as well as three sets of baseline methods: the multi-view learning approaches, the domain generalization approaches, and the latent domain discovering approaches. For SVM, the classifiers are trained on each view, and then we fuse the prediction scores from two views for the final prediction. For ESVM, one exemplar SVM classifier is trained for each positive training sample on each view, and then we use the same prediction method as in (5.22).

The multi-view learning baseline methods contain SVM-2K [49], kernel canonical correlation analysis (KCCA) [65], low-rank common subspace (LRCS) [34], and multiple kernel learning (MKL) [6] by utilizing two types of features, *i.e.*, RGB/DeCAF_6 features and depth/Caffe_6 features.

The domain generalization baseline methods include low-rank exemplar SVM (LRESVM) [164] and domain-invariant component analysis (DICA) [110]. Under the multi-view setting, LRESVM and DICA are employed on each view, and then we fuse the prediction scores from multiple views.

The latent domain discovering methods contain [69] and [59]. We learn the SVM classifiers for each discovered latent domain, followed by employing two prediction strategies

named "ensemble"and "match" as in [164]. employ two strategies to fuse the learnt classiers as suggested in [8, 9], which are referred as the ensemble strategy and the match strategy, respectively. The ensemble strategy is to re-weight the decision values from dierent SVM clas-siers by using the domain probabilities learnt with the method in [69]. In the match strategy, we rst select the most relevant domain based on the MMD criterion, and then use the SVM classier from this domain to predict the test samples. Similar to latent domains discovering algorithms, sub-categorization methods aim at discovering subcategories within each category, which can also be applied to our task. So we include the discriminative sub-categorization (Sub-Cate) method [67] as a baseline. For all the above methods, we employ them on each view, and then average the prediction scores from two views.

Moreover, in order to validate the co-regularizer in (5.2), we additionally report the results of a simplified version of our EMVDG_CO method, which is named EMVDG_CO_sim, in which the co-regularizer $\sum_{v,\tilde{v}:v\neq\tilde{v}} \|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2$ is removed by setting γ to 0.

For performance evaluation, the recognition accuracy is used for all approaches. For our EMVDG_CO method, the parameters are empirically fixed as $C = 0.1$, $\lambda_1 = 100$, $\lambda_1 = 10$, $\lambda_2 = 0.1$, $\gamma = 100$ for all settings on all datasets. For our EMVDG_MK method, the parameters are empirically fixed as $\zeta_1 = 10$, $\zeta_2 = 10000$ for all settings on all datasets. For the baselines, the optimal parameters are chosen based on their best performance on the test set. Due to the space limitation, only the average accuracy over the 3 (*resp.*, 6, 2) settings for the Office-Caltech (*resp.*, ACT4^2, ORGBD) dataset is reported.

Results: We summarize the experimental results in Table 5.1, from which we observe that ESVM outperforms SVM, which indicates the effectiveness of fusing multiple exemplar SVM classifiers to enhance the domain generalization ability.

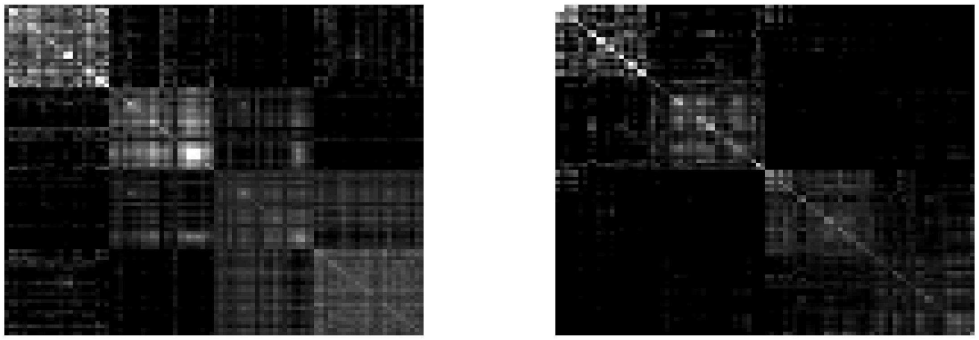
Multi-view learning approaches LRCS, SVM-2K, KCCA, and MKL acheive better results than SVM because they exploit the relation among multiple types of features. LRESVM, DICA, and Sub-Cate are all better than SVM, which indicates that it is useful to exploit the intrinsic structure when the training data are sampled from multiple latent domains. The latent domain discovering approaches [59, 69] using the "match" or "ensemble" strategy generally outperform SVM, which shows the effectiveness of discovering the latent domains.

Table 5.1: Average accuracies (%) over multiple settings of different approaches on each dataset without using the target domain samples during the training procedure. We denote the best results in boldface.

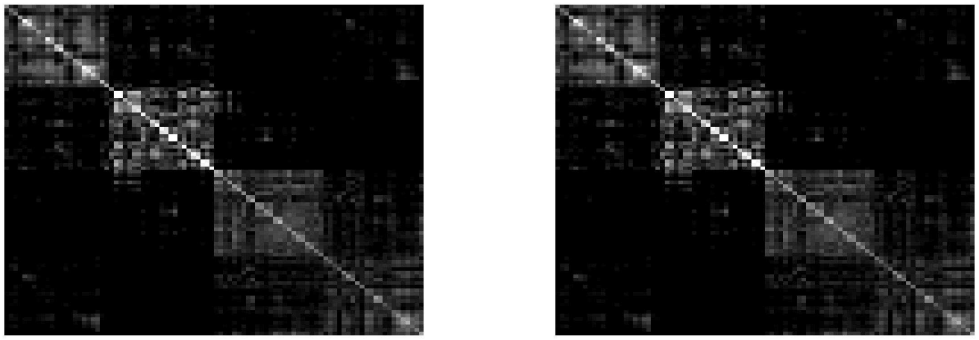
Dataset	ACT4[2]	ORGBD	Office-Caltech
SVM [29]	68.10	62.05	84.52
ESVM [108]	69.11	62.95	86.14
LRCS [34]	70.81	66.07	85.28
SVM-2K [49]	70.34	65.63	86.10
KCCA [65]	69.56	63.84	86.33
MKL [6]	69.98	65.18	86.50
DICA [110]	69.53	66.52	86.12
LRESVM [164]	71.18	67.42	87.04
[59](match)	70.05	65.63	86.47
[59](ensemble)	69.28	66.52	86.06
[69](match)	68.60	61.16	85.75
[69](ensemble)	68.66	65.63	84.81
Sub-Cate [67]	69.90	64.74	86.64
EMVDG_CO_sim	72.10	67.86	87.72
EMVDG_CO	**74.22**	69.20	88.13
EMVDG_MK	73.08	**70.54**	**88.33**

Another observation is that EMVDG_CO_sim is better than ESVM on all datasets. Since ESVM can be treated as a special case of our EMVDG_CO_sim method without employing low rank representation (LRR), the results indicate the benefits of exploiting the low-rank structure of positive training samples for domain generalization. Our EMVDG_CO method achieves better results than its simplified version EMVDG_CO_sim, which shows that our new co-regularizer $\sum_{v,\tilde{v}:v\neq\tilde{v}} \|\mathbf{Z}^v - \mathbf{Z}^{\tilde{v}}\|_F^2$ is effective. So it is useful to jointly exploit the cluster structures from multiple views.

Our EMVDG_CO and EMVDG_MK methods outperform all the baseline methods on all three datasets, which indicates that our EMVDG framework can improve the domain generalization ability and utilize multiple types of features effectively. Note that there is no consistent winner in our EMVDG framework. In particular, our EMVDG_CO



(a) $\mathbf{Z}$ on RGB view w/o co-reg (b) $\mathbf{Z}$ on depth view w/o co-reg



(c) $\mathbf{Z}$ on RGB view with co-reg (d) $\mathbf{Z}$ on depth view with co-reg

Figure 5.1: Illustration of the learnt representation matrices on two views for the action "Put On" on the ACT42 dataset when treating the camera viewpoint 1 and 4 (*resp.*, 2 and 3) as the source (*resp.*, target) domain.

method achieves the best result on the ACT4[2] dataset while our EMVDG_MK method outperforms EMVDG_CO on the ORGBD dataset and achieves the comparable result on the Office-Caltech dataset.

Analysis on the learnt representation matrices using EMVDG_CO_sim and EMVDG_CO: In order to demonstrate how our EMVDG_CO method exploits the latent domains of positive training samples in an intuitive way, we take the ACT4[2] dataset as an example to compare the representation matrices $\mathbf{Z}^v$'s (*i.e.*, $\mathbf{Z}^{\text{RGB}}$ and $\mathbf{Z}^{\text{depth}}$) learnt by using our EMVDG_CO method and its simplified version EMVDG_CO_sim in Fig-

ure 5.1, which correspond to MVDG and MVDG (w/o co-reg) in the preliminary conference version respectively. Recall that the representation matrix $\mathbf{Z}^v$ encodes the cluster structure of exemplar classifiers, in which the between-cluster (*resp.*, within-cluster) entries are generally sparse (*resp.*, dense). So in ideal cases, $\mathbf{Z}^v$ should be block-diagonal with each block representing a latent domain. From Figure 5.1, we observe that all 4 representation matrices exhibit block-diagonal structure, which indicates that it is effective to discover hidden latent domains by employing LRR on each view. It is worth noting that although only two domains (i.e., camera viewpoint 1 and camera viewpoint 4) are merged as the source domain, there are in fact 4 blocks in Figure 5.1, which means that totally 4 latent domains are discovered. We conjecture that actors are likely to put on clothes from two opposite directions with each direction leading to a latent domain. Thus, the videos captured from each camera viewpoint actually contain two latent domains. Another observation is that with our newly proposed co-regularizer, the two representation matrices learnt by using our EMVDG_CO method are more consistent and also exhibit relatively clearer block-diagonal structure than those learnt by using the simplified version EMVDG_CO_sim. This result demonstrates the benefits of using our co-regularizer. We have similar observations for the other scenarios.

Analysis on the learnt kernel weights using EMVDG_MK: From Table 5.1, we observe that our EMVDG_MK method outperforms our EMVDG_CO method on the ORGBD dataset, which could be explained as follows. Since two types of features (*i.e.*, RGB and depth) are used on the ORGBD dataset, we conjecture that one of them (*i.e.*, RGB or depth) is more discriminative so that assigning higher weight for more discriminative features will help better exploit the latent domain structure and learn more robust classifiers, which leads to better performance. To this end, we analyze the learnt kernel weights $\mathbf{d}$ in (5.28) by taking the two settings on the ORGBD dataset as examples.

To capture the relation between the kernels constructed from different types of features and the learnt kernel weights, we additionally report the accuracies of SVM by using only RGB or depth features. When the performance of SVM obtained based on one feature is higher than the other one, the corresponding kernel is expected to be more

Figure 5.2: Illustration of the kernel combination weights and the accuracies of SVM based on each kernel corresponding to RGB and depth features on the two settings on the ORGBD dataset.

discriminative and the weight assigned to this kernel is expected to be higher. With regards to the kernel weights, we have a set of learnt kernel combination weights for each category, as our EMVDG_MK method is under the binary classification setting. For better representation, we report the average of the learnt kernel weights over all categories. To this end, we illustrate the accuracies of SVM and the learnt kernel weights in Fig. 5.2, from which we observe that the SVM classifiers trained based on the RGB features achieve better performance on both settings. Moreover, higher weights are correctly assigned to the RGB kernel by EMVDG_MK on both settings, which demonstrates that our EMVDG_MK method can select more discriminative kernels.

Parameter Study: Note that for our EMVDG_CO and EMVDG_MK methods, there exist several trade-off parameters, which are empirically fixed in our experiments. Actually, our methods are insensitive to the parameters when varying one parameter in certain range while fixing other parameters. By taking our EMVDG_MK method on the ACT4^2 dataset as an example, when setting $\zeta_1 \in [10^{-2}, 10^2]$, the accuracy of EMVDG_MK is in the range of $[73.04\%, 73.80\%]$. Similarly, the accuracy of EMVDG_MK is in the range of $[72.66\%, 73.16\%]$ when setting $\zeta_2 \in [10^1, 10^5]$.

5.5.2 Domain Adaptation

Experimental Settings: We use the same experimental settings as in Section 5.5.1 except that we additionally use the unlabeled target domain data in the training process.

Table 5.2: Average accuracies (%) over multiple settings of different approaches on each dataset after utilizing the target domain samples during the training procedure. The best results are denoted in boldface.

Dataset	ACT4[2]	ORGBD	Office-Caltech
SVM [29]	68.10	62.05	84.52
DASVM [16]	70.61	65.63	85.60
KMM [71]	71.74	65.18	86.34
TCA [119]	70.29	64.74	85.79
SA [53]	71.85	67.42	86.79
DIP [7]	71.49	66.07	86.58
GFK [60]	70.31	65.63	86.22
SGF [62]	71.31	62.50	85.78
Co-training [12]	73.36	67.86	87.96
Co-LapSVM [141]	73.76	68.31	88.20
Coupled [11]	73.16	68.31	86.48
MVTL_LM [174]	73.80	62.95	87.76
MDT [166]	72.66	67.42	86.87
DTMKL [39]	73.57	68.31	88.07
LRCS [34]	73.04	67.42	86.12
EMVDA_CO	**76.18**	70.09	**91.04**
EMVDA_MK	75.08	**71.88**	89.84

Baselines: We compare our EMVDA framework including EMVDA_CO and EMVDA_MK methods with three sets of baseline methods: the domain adaptation approaches and the multi-view semi-supervised learning approaches as well as the existing multi-view domain adaptation approaches.

The domain adaptation baselines are kernel mean matching (KMM) [71], domain adaptive SVM (DASVM) [16], domain invariant projection (DIP) [7], subspace alignment (SA) [53], transfer component analysis (TCA) [119], sampling geodesic flow (SGF) [62], and geodesic flow kernel (GFK) [60]. The above domain adaptation approaches are employed on each view, followed by fusing the prediction scores from two views using the

late fusion strategy.

Our EMVDA framework is also compared with multi-view semi-supervised learning approaches Co-LapSVM [141] and Co-training [12], together with the multi-view domain adaptation approaches including multi-view transfer learning (MVTL_LM) [174], Coupled [11], multi-view discriminant transfer (MDT) [166], and domain transfer multiple kernel learning (DTMKL) [39], which exploit the relation among multiple types of features and simultaneously cope with the domain distribution mismatch. Additionally, we compare our EMVDA framework with low-rank common subspace (LRCS) [34] by using the target domain samples as the dictionary as suggested in [34].

Compared with EMVDG_CO, our EMVDA_CO method has an extra parameter θ, which is empirically set as 10^{-5} for all settings on all datasets. Similarly, compared with EMVDG_MK, our EMVDA_MK method has an extra parameter ϑ, which is empirically set as 10^{-7} for all settings on all datasets. For the baselines, the optimal parameters are chosen based on their best results on the test set. Due to the space limitation, only the average accuracy over the 3 (*resp.*, 6, 2) settings for the Office-Caltech (*resp.*, ACT4^2, ORGBD) dataset is reported.

Results: We summarize the experimental results in Table 5.2. The results of SVM from Table 5.1 are also included for comparison. It can be observed that the domain adaptation approaches DASVM, KMM, SA, DIP, GFK, TCA, and SGF outperform SVM, which indicates the advantage of reducing the domain distribution mismatch between the source domain and the target domain.

We also observe that the multi-view semi-supervised learning approaches Co-LapSVM and Co-training as well as the multi-view domain adaptation approaches Coupled, MVTL_LM, MDT, and DTMKL are generally better than the multi-view learning approaches reported in Table 5.1, which demonstrates the effectiveness of utilizing the unlabeled target domain samples. Another observation is that the multi-view domain adaptation approaches are generally better than or comparable to other domain adaptation approaches, which shows the advantage of additionally exploiting the relation among multiple views. LRCS also achieves better results by using the target domain data as the dictionary, when compared with its corresponding results without using the target domain data.

Our EMVDA_CO (*resp.*, EMVDA_MK) method outperforms our EMVDG_CO (*resp.*, EMVDG_MK) method reported in Table 5.1, which indicates the benefits of utilizing the

Table 5.3: Average training time (s) of our EMVDG and EMVDA frameworks on the Office-Caltech dataset by employing 2-view or 4-view features.

Method	EMVDG_CO	EMVDG_MK	EMVDA_CO	EMVDA_MK
2 views	2795.2013	399.4078	3245.8470	570.6233
4 views	4815.8333	439.0771	7630.5167	643.8519

Table 5.4: Average accuracies (%) of our EMVDG and EMVDA frameworks on the Office-Caltech dataset by employing 2-view or 4-view features.

Method	EMVDG_CO	EMVDG_MK	EMVDA_CO	EMVDA_MK
2 views	88.13	88.33	91.04	89.84
4 views	91.54	90.63	93.26	92.20

unlabeled target domain data during the training procedure. Moreover, our EMVDA_CO and EMVDA_MK methods outperform all the baselines on all datasets. Our EMVDA_CO method achieves the best results on the ACT4[2] and Office-Caltech dataset while our EMVDA_MK achieves the best result on the ORGBD dataset.

5.5.3 Utilizing Multiple Types of Features

Although we only use two types of features (*i.e.*, RGB/depth features for human action recognition and Decaf_6/Caffe_6 for object recognition) in Section 5.5.1 and Section 5.5.2, our EMVDG and EMVDA frameworks can be readily used for multiple types of features. When employing more types of features, our EMVDG_MK and EMVDA_MK methods are much more efficient than our EMVDG_CO and EMVDA_CO methods, which can be explained as follows. For our EMVDG_CO method, we need to update $\mathbf{W}^v$'s and $\mathbf{G}^v$'s on each view as indicated in Algorithm 8, and update $\mathbf{Z}^v$'s by solving the subproblems on each view as indicated in Algorithm 7. So the training time of our EMVDG_CO method increases linearly with the number of views. In contrast, for our EMVDG_MK method, the most time-consuming steps are to solve the problem in (5.31) and the inner problem of (5.32), and their time complexity is irrelevant to the number of views, as indicated in Algorithm 9. So the extra training time of our EMVDG_MK method with multiple types of features is much less than that of EMVDG_CO. The analysis of the time complexity

for EMVDA_CO and EMVDA_MK is similar to that for EMVDG_CO and EMVDG_MK, respectively.

To compare EMVDG_MK (*resp.*, EMVDA_MK) with EMVDG_CO (*resp.*, EMVDA_CO) in terms of the training time and accuracy when using different numbers of views, we take the Office-Caltech dataset as an example to conduct experiments on a server machine with Intel Xeon 3.2 GHz CPUs and 16 GB RAM using a single thread. Besides the Decaf_6 and Caffe_6 features, we additionally use Decaf_7 and Caffe_7 features, which leads to four types of features in total. The average training time over three settings of our four methods is reported in Table 5.3, from which we can observe that the training time of EMVDG_CO and EMVDA_CO approximately increases linearly as the number of views increases while the training time of EMVDG_MK and EMVDA_MK increases much less. We also report the average accuracies over three settings of our four methods in Table 5.4, from which we observe that the performances of all four methods are improved after employing two more types of features. When using four types of features, EMVDG_CO (*resp.*, EMVDA_CO) achieves better result than EMVDG_MK (*resp.*, EMVDA_MK). However, our EMVDG_MK and EMVDA_MK methods are much more efficient.

5.6 Summary

In this chapter, an exemplar-based multi-view domain generalization (EMVDG) framework has been proposed for visual recognition. Our framework can enhance the domain generalization capability to arbitrary target domain and simultaneously exploit the relation among multiple types of features. Moreover, our EMVDG framework has been further extended to a new domain adaptation framework named EMVDA by additionally using the unlabeled target domain samples in the training process. The effectiveness of our EMVDG and EMVDA frameworks has been demonstrated by extensive experiments for visual recognition on three benchmark datasets.

Chapter 6

Domain Adaptation Based on Fisher Vector for Visual Recognition

Recently, Fisher vector is widely used and has achieved excellent performance in various visual recognition tasks. In this chapter, we consider Fisher vector in the context of domain adaptation, which has rarely been discussed by the existing domain adaptation methods. Particularly, in many real scenarios, the distributions of Fisher vectors of the training samples (*i.e.*, source domain) and test samples (*i.e.*, target domain) are considerably different, which may degrade the classification performance on the target domain by using the classifiers/regressors learnt based on the training samples from the source domain. To address the domain shift issue, we propose a Domain Adaptive method based on Fisher Vector (DAFV), which learns a transformation matrix to select the domain invariant components of Fisher vectors and simultaneously solves a regression problem for visual recognition tasks based on the transformed features. Specifically, we employ a group lasso based regularizer on the transformation matrix to select the components of Fisher vectors, and use a regularizer based on the Maximum Mean Discrepancy (MMD) criterion to reduce the data distribution mismatch of transformed features between the source domain and the target domain. Comprehensive experiments demonstrate the effectiveness of our DAFV method on two benchmark datasets.

6.1 Introduction

Constructing global feature representations based on local descriptors of images/videos is a common approach in a multitude of visual recognition tasks. In particular, for each image/video, we extract a set of local descriptors and encode them into a high dimensional

vector based on different encoding methods. As a commonly used encoding method, Fisher vector [74] encodes both first and second order statistical information of local descriptors *w.r.t.* the generative model (*e.g.*, Gaussian Mixture Model (GMM)) trained based on them, and one Gaussian model in the GMM corresponds to one component in the extracted Fisher vector. Recently, Fisher vector achieves excellent performance for object recognition by using traditional hand-craft features (*e.g.*, SIFT) as local descriptors [125, 140] or CNN features of proposals as local descriptors [149, 159]. Moreover, Fisher vector also achieves competitive results for human action recognition by using Improved Dense Trajectory (IDT) features as local descriptors [123, 155]. To extract Fisher vector, we generally train a GMM based on the local descriptors of training samples and extract Fisher vectors for both training and test samples based on the pre-trained GMM. However, the GMM trained on the training samples does not consider the data distribution of test samples properly and thus lacks the generalization ability [36] on the test samples, leading to unsatisfactory recognition performance on the test datasets.

When the target domain data are available in the training stage, we can train GMMs based on the mixture of local descriptors from both source domain and target domain. However, even in this case, the generated Fisher vectors of source domain samples and target domain samples may be still considerably different in terms of statistical properties, which is referred to as dataset bias [144]. Instead of training GMMs based on the data from both domains, another approach is to adapt the GMM trained based on the source domain to the target domain, or interpolate two GMMs which are trained based on the source domain and the target domain separately. In particular, some Bayesian model adaptation methods such as [36] can be used to adapt the background GMM to new samples and a recent work [81] can be used to interpolate a set of GMMs. However, these methods did not explicitly consider the domain distribution mismatch between the source domain and the target domain. So they cannot guarantee the extracted Fisher vectors based on the adapted or interpolated GMMs are domain invariant.

In recent works, many domain adaptation approaches [3, 7, 16, 53, 60, 62, 71, 119, 133, 136] have been proposed to tackle the domain shift issue between the source domain and the target domain (see Section 6.2 for details). However, none of them is specifically designed for Fisher Vector, since they did not take the generative models (*i.e.*, GMMs)

into consideration. Therefore, the excellent performance of Fisher vector for visual recognition [123, 149] and the lack of effective domain adaptation methods for Fisher vector motivate our work. By noticing that each Gaussian model in the GMM characterizes the data distribution of a cluster of local descriptors, and some Gaussian models are more likely to capture the common data distribution between the source domain and the target domain, we come out the idea of identifying the common Gaussian models via selecting the corresponding components of Fisher vectors that are more likely to be domain invariant.

Let us take the object recognition and human action recognition tasks as two examples to provide more explanations for domain invariant components. For object recognition, the appearance of images within the same category may be quite different between the source domain and the target domain, which is usually referred to as intra-class difference, while some specific object regions within the category may be relatively consistent. Considering extracting the CNN features of object proposals as local descriptors and encoding them into Fisher vectors based on the pre-trained GMM, we expect to select the components of Fisher vectors corresponding to the Gaussian models from the object proposals which are more consistent across the source domain and the target domain. To validate this point, we present a detailed showcase associated with more discussions in Section 6.5.1. For human action recognition, sometimes the videos in the source domain are captured from the front view while the videos in the target domain are captured from the back view. When using the popular Improved Dense Trajectory (IDT) features as local descriptors in videos, each trajectory represents a local movement of human body, some of which can be observed from both front view and back view while the others can only be observed from one view. After encoding the IDT descriptors in videos into Fisher vectors based on the pre-trained GMM, we want to select the components of Fisher vectors corresponding to the Gaussian models from the trajectories which can be observed from both views.

To this end, we propose our Domain Adaptation method based on Fisher Vector (DAFV). Specifically, we learn a transformation matrix to project the Fisher vectors into a lower dimensional latent subspace and consider visual recognition task as a regression problem based on the transformed features. A group lasso based regularizer [170] is

employed on the transformation matrix to enforce the components of the transformation matrix corresponding to the selected (*resp.*, unselected) components of Fisher vectors to be associated with large (*resp.*, small) weights. At the same time, we apply the criterion of minimizing the Maximum Mean Discrepancy (MMD) of the transformed features between the source domain and the target domain by using an MMD-based regularizer. In Section 6.3, we briefly provide the background knowledge of Fisher vector. In Section 6.4, we introduce our Domain Adaptation method based on Fisher Vector (DAFV) in detail and also present a novel solution to the nontrivial optimization problem. In section 6.5, we conduct extensive experiments on two benchmark datasets Bing-Caltech256 and ACT4[2] to demonstrate the effectiveness of our proposed method.

6.2 Related Work

This chapter is related to using Fisher vector for visual recognition tasks. Fisher vector was first used for image classification in [124] and further improved in [125] with power normalization and L_2 normalization. In [140], Simonyan *et al.* developed a two-layer deep network based on Fisher vector for large-scale image classification. More recently, with the breakthrough in image representation by using Convolutional Neural Networks (CNN), CNN features of local regions have been used as local descriptors for Fisher vector [61, 104, 149, 159]. Fisher vector was also applied to video action and event recognition [118, 155]. Similar to the idea in [140] for image classification, Peng *et al.* proposed stacked Fisher vectors for human action recognition in [123]. All these methods assume the training samples and test samples are with the same data distribution while this assumption does not hold in domain adaptation scenarios.

This chapter is related to domain adaptation. The existing domain adaptation methods can be classified into feature-based methods [3, 7, 19, 53, 60, 62, 86, 119], SVM-based methods [16, 39, 41], instance-reweighting methods [71], dictionary learning methods [136], and low-rank based methods [76, 133]. However, all the above methods are not specifically designed for Fisher vector. Among them, our method is more related to [7] and [119] which also learn a transformation matrix. However, [7, 119] are only feature learning methods without considering the property of Fisher vector while our method

can select the domain invariant components of Fisher vectors and simultaneously learn the regression matrix.

Finally, this chapter is also related to adapted or interpolated GMMs. Recently, Bayesian model adaptation has attracted much attention and several approaches have been proposed to adapt the background GMM to each image [178] or each category with very few examples [51]. Then, a more general formulation of Bayesian adaptation was proposed in [36] for image classification. Note that these methods [36, 51, 178] focus on adapting the background GMM to either a new image or a new category instead of considering the difference between two domains. So the motivation of their methods is intrinsically different from ours. More recently, Kim *et al.* proposed to interpolate a set of GMMs on the manifold in [81], which can be used to learn the interpolation between two GMMs from two domains. Nevertheless, all the above works did not explicitly address the domain shift issue. In contrast, our method explicitly reduces the domain distribution mismatch between two domains. Moreover, the Fisher vectors based on the GMMs learnt by their methods can be readily used to replace the original Fisher vectors in our method to further improve the performance.

6.3 Fisher Vector

Fisher vector is a commonly used encoding method to construct global feature representations from local descriptors. As a combination of generative and discriminative approaches, on one hand, the generation procedure of a set of local descriptors $\mathbf{X} = \{\mathbf{x}_i|_{i=1}^N\}$ (N is the number of local descriptors) is assumed to obey a probability density function $p(\mathbf{X}; \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$. On the other hand, the gradients of the log-likelihood *w.r.t.* the model parameters, which describe the contribution of model parameters to the generation procedure of $\mathbf{X}$ [74], can be used as input features for discriminative methods such as classifiers and regressors. Since each image/video can be treated as a set of local descriptors $\{\mathbf{x}_i|_{i=1}^N\}$, its Fisher vector can be represented as,

$$G_{\boldsymbol{\theta}}^{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i; \boldsymbol{\theta}). \tag{6.1}$$

For visual recognition tasks, the probability density function $p(\mathbf{X};\boldsymbol{\theta})$ is usually modeled by Gaussian Mixture Model (GMM) [118, 124]. Suppose K is the number of Gaussian models in the GMM, we use model parameters $\boldsymbol{\theta} = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1; \ldots; \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\sigma}_K\}$ to denote the mixture weights, means, and diagonal covariances of GMM, respectively. Based on the definition of Fisher vector (6.1), the gradients of the log-likelihood *w.r.t.* the model parameters (*i.e.*, means and diagonal covariances) of the k-th Gaussian model can be written as (refer to [124] for the derivation details),

$$\mathcal{G}^{\mathbf{X}}_{\boldsymbol{\mu},k} = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} \gamma_i(k)\left(\frac{\mathbf{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}\right), \tag{6.2}$$

$$\mathcal{G}^{\mathbf{X}}_{\boldsymbol{\sigma},k} = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} \gamma_i(k)\left[\frac{(\mathbf{x}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1\right], \tag{6.3}$$

where $\gamma_i(k)$ is the probability that the i-th local descriptor $\mathbf{x}_i$ belongs to the k-th Gaussian model, which is defined as,

$$\gamma_i(k) = \frac{\pi_k \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}, \tag{6.4}$$

in which $\mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$ is the probability of $\mathbf{x}_i$ based on the Gaussian distribution of the k-th Gaussian model. Assuming that the dimension of local descriptors is d, then the dimension of the k-th component of Fisher vectors corresponding to the k-th Gaussian model is $2d$ by concatenating (6.2) and (6.3). So the final Fisher vector is a $2Kd$-dim vector *w.r.t.* a K-component GMM.

6.4 Domain Adaptation based on Fisher Vector

In this section, we introduce our Domain Adaptation method based on Fisher Vector (DAFV), in which we select the domain invariant components of Fisher vectors by simultaneously learning a transformation matrix and a regression matrix for visual recognition tasks. In order to make the proposed formulation easier to be optimized, we introduce an intermediate variable and relax our formulation, and then develop an effective algorithm to solve the optimization problem.

6.4.1 Formulation

Suppose we have n_s source domain samples and n_t target domain samples from C categories. Each sample is represented by a $2Kd$-dim Fisher vector, in which d is the dimension of local descriptors and K is the number of Gaussian models in the GMM. Let us denote $\mathbf{X}^s \in \mathcal{R}^{2Kd \times n_s}$ and $\mathbf{X}^t \in \mathcal{R}^{2Kd \times n_t}$ as the features of source domain samples and target domain samples, and $\mathbf{Y} \in \mathcal{Z}^{C \times n_s}$ as the binary label matrix for the source domain samples. In order to select domain invariant components and simultaneously keep discriminative information, we use the transformation matrix $\mathbf{R} \in \mathcal{R}^{m \times 2Kd}$ to project the original Fisher vector to lower dimensional subspace with m being the dimension of transformed features. We employ the group lasso based regularizer [170] $\|\tilde{\mathbf{R}}\|_{2,1}$ to enforce each column of $\tilde{\mathbf{R}}$ to have either all zero weights or multiple nonzero weights, in which $\tilde{\mathbf{R}} \in \mathcal{R}^{2d \times Km}$ is a reshaped matrix of $\mathbf{R}$ by setting each group of $2d$ entries in each row of $\mathbf{R}$ corresponding to one component in the Fisher vector as one column in $\tilde{\mathbf{R}}$. To be exact, we expect to assign nonzero weights to the selected domain invariant components of Fisher vectors and zero weights to the remaining ones.

To ensure the selected components are domain invariant, we tend to minimize the Maximum Mean Discrepancy (MMD) of transformed features between the source domain and the target domain by using an MMD-based [71] regularizer $\|\frac{1}{n_s}\mathbf{R}\mathbf{X}^s\mathbf{1} - \frac{1}{n_t}\mathbf{R}\mathbf{X}^t\mathbf{1}\|^2$, in which $\frac{1}{n_s}\mathbf{R}\mathbf{X}^s\mathbf{1}$ (*resp.*, $\frac{1}{n_t}\mathbf{R}\mathbf{X}^t\mathbf{1}$) is the mean of transformed features from the source (*resp.*, target) domain, so that the data distribution mismatch between two domains can be reduced. Additionally, inspired by [119], we add a constraint $\mathbf{R}\mathbf{X}\mathbf{H}\mathbf{X}'\mathbf{R}' = \mathbf{I}$ to maximally preserve the data variance, where $\mathbf{X} = [\mathbf{X}^s, \mathbf{X}^t]$ and $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}'$ with $n = n_s + n_t$.

By denoting $\mathbf{W} \in \mathcal{R}^{C \times m}$ as the regression matrix, we formulate our method by solving the following regression problem:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{R}} \quad & \frac{1}{2}\|\mathbf{W}\mathbf{R}\mathbf{X}^s - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \lambda\|\tilde{\mathbf{R}}\|_{2,1} \\ & + \frac{1}{2}\|\frac{1}{n_s}\mathbf{R}\mathbf{X}^s\mathbf{1} - \frac{1}{n_t}\mathbf{R}\mathbf{X}^t\mathbf{1}\|^2 \qquad (6.5) \\ \text{s.t.} \quad & \mathbf{R}\mathbf{X}\mathbf{H}\mathbf{X}'\mathbf{R}' = \mathbf{I}, \qquad (6.6) \end{aligned}$$

in which $\|\mathbf{W}\mathbf{R}\mathbf{X}^s - \mathbf{Y}\|_F^2$ is the regression error, $\|\mathbf{W}\|_F^2$ is the weight decay regularizer to control the complexity of $\mathbf{W}$, γ and λ are two trade-off parameters.

The problem in (6.5) is not easy to solve due to the constraint in (6.6). For ease of optimization, we introduce an intermediate variable $\mathbf{S}$ and promote the coherence between $\mathbf{R}$ and $\mathbf{S}$ by adding a coherent regularizer $\|\mathbf{RS}'\|_F^2$ [130]. With larger $\|\mathbf{RS}'\|_F^2$, $\mathbf{R}$ is more coherent to $\mathbf{S}$. As a result, the proposed formulation after introducing $\mathbf{S}$ becomes,

$$\begin{aligned} \min_{\mathbf{W},\mathbf{R},\mathbf{S}} \quad & \frac{1}{2}\|\mathbf{WSX}^s - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \lambda\|\tilde{\mathbf{S}}\|_{2,1} \\ & + \frac{1}{2}\|\frac{1}{n_s}\mathbf{RX}^s\mathbf{1} - \frac{1}{n_t}\mathbf{RX}^t\mathbf{1}\|^2 - \frac{1}{2}\|\mathbf{RS}'\|_F^2 \qquad (6.7) \\ \text{s.t.} \quad & \mathbf{RXHX}'\mathbf{R}' = \mathbf{I}. \qquad (6.8) \end{aligned}$$

By replacing $\mathbf{R}$ in $\|\mathbf{WRX}^s - \mathbf{Y}\|_F^2$ and $\|\tilde{\mathbf{R}}\|_{2,1}$ in (6.5) by $\mathbf{S}$, the subproblem *w.r.t.* $\mathbf{R}$ in (6.7) can be easily solved by using eigen decomposition, which will be discussed in detail in the next section.

Another problem is that the dimension of Fisher vector is usually very high. Considering high time-complexity operations such as eigen decomposition, the algorithm will become very time-consuming. To accelerate the algorithm and simultaneously capture the semantic information within each category, we partition each Fisher vector into C uncorrelated parts by training a category-specific GMM with a smaller number of Gaussian models based on the training samples within each category. Then, a set of $\mathbf{W}_c$, $\mathbf{R}_c$, and $\mathbf{S}_c$ is learnt for the components of each Fisher vector corresponding to the c-th GMM. As a result, we have totally C sets of $\mathbf{W}_c \in \mathcal{R}^{C\times\bar{m}}$, $\mathbf{R}_c \in \mathcal{R}^{\bar{m}\times 2\bar{K}d}$, and $\mathbf{S}_c \in \mathcal{R}^{\bar{m}\times 2\bar{K}d}$ for $c = 1, \ldots, C$, in which we denote the number of Gaussian models in each category-specific GMM as $\bar{K}$ ($\bar{K} << K$) and the dimension of the transformed features corresponding to each category-specific GMM as $\bar{m}$ ($\bar{m} << m$). Correspondingly, we partition the training (*resp.*, test) features $\mathbf{X}^s$ (*resp.*, $\mathbf{X}^t$) into $\mathbf{X}_c^s \in \mathcal{R}^{2\bar{K}d\times n_s}$'s (*resp.*, $\mathbf{X}_c^t \in \mathcal{R}^{2\bar{K}d\times n_t}$'s) with each obtained based on the c-th GMM, and denote $\mathbf{X}_c = [\mathbf{X}_c^s, \mathbf{X}_c^t]$. In fact, supervised learning for GMM (*i.e.*, train one GMM per category) has been studied in [50] and proved to be able to preserve the useful discriminative information. To this end, we can relax the problem in (6.7) as,

$$\min_{\mathbf{W}_c,\mathbf{R}_c,\mathbf{S}_c} \quad \frac{1}{2}\|\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{X}_c^s - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\sum_{c=1}^{C}\|\mathbf{W}_c\|_F^2 + \lambda\sum_{c=1}^{C}\|\tilde{\mathbf{S}}_c\|_{2,1}$$
$$+\frac{1}{2}\sum_{c=1}^{C}\|\frac{1}{n_s}\mathbf{R}_c\mathbf{X}_c^s\mathbf{1} - \frac{1}{n_t}\mathbf{R}_c\mathbf{X}_c^t\mathbf{1}\|^2 - \frac{1}{2}\sum_{c=1}^{C}\|\mathbf{R}_c\mathbf{S}_c'\|_F^2 \tag{6.9}$$
$$\text{s.t.} \quad \mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c' = \mathbf{I}, \quad \forall c. \tag{6.10}$$

By partitioning a Fisher vector into C uncorrelated parts, we can solve C small-scale subproblems instead of a large-scale problem, which is more efficient. Considering the tradeoff between efficiency and effectiveness, we set $\bar{K}$ as 8 and $\bar{m}$ as 1000 in our experiments. Moreover, another benefit of replacing $\|\tilde{\mathbf{S}}\|_{2,1}$ with $\|\tilde{\mathbf{S}}_c\|_{2,1}$ is that we can guarantee at least one Gaussian model is selected from each category-specific GMM, which ensures capturing the semantic information over all categories. Next, we will discuss how to solve the problem in (6.9).

6.4.2 Optimization

We solve the problem in (6.9) by using an alternative optimization approach. Specifically, we alteratively update three sets of variables $\mathbf{W}_c$'s, $\mathbf{S}_c$'s, and $\mathbf{R}_c$'s until the objective value of (6.9) converges.

Update $\mathbf{W}_c$ when fixing $\mathbf{R}_c$ and $\mathbf{S}_c$: When fixing $\mathbf{R}_c$'s and $\mathbf{S}_c$'s, the problem in (6.9) reduces to:

$$\min_{\mathbf{W}_c} \frac{1}{2}\|\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{X}_c^s - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\sum_{c=1}^{C}\|\mathbf{W}_c\|_F^2 \tag{6.11}$$

By setting the derivative of (6.11) *w.r.t.* each $\mathbf{W}_c$ to $\mathbf{0}$, we can derive the close-form solution for each $\mathbf{W}_c$ as,

$$\mathbf{W}_c = (\mathbf{Y} - \sum_{\tilde{c}=1,\tilde{c}\neq c}^{C}\mathbf{W}_{\tilde{c}}\mathbf{S}_{\tilde{c}}\mathbf{X}_{\tilde{c}}^s)\mathbf{X}_c^{s\prime}\mathbf{S}_c'(\mathbf{S}_c\mathbf{X}_c^s\mathbf{X}_c^{s\prime}\mathbf{S}_c' + \gamma\mathbf{I})^{-1}. \tag{6.12}$$

We calculate each $\mathbf{W}_c$ when fixing all the other $\mathbf{W}_{\tilde{c}}$ for $\tilde{c} \neq c$ and repeat this process iteratively until the objective value of (6.11) converges.

Update $\mathbf{R}_c$ when fixing $\mathbf{W}_c$ and $\mathbf{S}_c$: When fixing $\mathbf{W}_c$'s and $\mathbf{S}_c$'s, the problem in (6.9) can be separated into C independent subproblems with one for each $\mathbf{R}_c$. For ease of optimization, we rewrite the subproblem *w.r.t.* each $\mathbf{R}_c$ by using trace norm as follows,

$$\min_{\mathbf{R}_c} \quad \frac{1}{2}\text{tr}(\mathbf{R}_c\mathbf{X}_c\mathbf{L}\mathbf{X}_c'\mathbf{R}_c') - \frac{1}{2}\text{tr}(\mathbf{R}_c\mathbf{S}_c'\mathbf{S}_c\mathbf{R}_c') \tag{6.13}$$

$$\text{s.t.} \quad \mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c' = \mathbf{I}, \tag{6.14}$$

where $\mathbf{L}$ is an indicator matrix, in which $L_{ij} = \frac{1}{n_s^2}$ if $i \leq n_s$ and $j \leq n_s$; else $L_{ij} = \frac{1}{n_t^2}$ if $i > n_s$ and $j > n_s$; otherwise, $L_{ij} = -\frac{1}{n_s n_t}$.

By introducing a symmetric matrix $\mathbf{Z}_c$ containing the Lagrangian multipliers for the constraints in (6.14), we obtain the Lagrangian form of (6.13) as,

$$\mathcal{L}_{\mathbf{R}_c,\mathbf{Z}_c} = \text{tr}(\mathbf{R}_c(\frac{1}{2}\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \frac{1}{2}\mathbf{S}_c'\mathbf{S}_c)\mathbf{R}_c') - \text{tr}((\mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c' - \mathbf{I})\mathbf{Z}_c). \tag{6.15}$$

By setting the derivative of (6.15) *w.r.t.* $\mathbf{R}_c$ to $\mathbf{0}$, we arrive at

$$\mathbf{R}_c(\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \mathbf{S}_c'\mathbf{S}_c) = 2\mathbf{Z}_c\mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'. \tag{6.16}$$

Multiplying both sides on the right by $\mathbf{R}_c'$, we obtain the solution *w.r.t.* $\mathbf{Z}_c$ as follows,

$$\mathbf{Z}_c = \frac{1}{2}(\mathbf{R}_c(\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \mathbf{S}_c'\mathbf{S}_c)\mathbf{R}_c')(\mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c')^{-1}. \tag{6.17}$$

By substituting (6.17) back into (6.15) followed by some simplifications, we reach an equivalent problem of (6.13) as

$$\max_{\mathbf{R}_c} \frac{1}{2}\text{tr}((\mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c')(\mathbf{R}_c(\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \mathbf{S}_c'\mathbf{S}_c)\mathbf{R}_c')^{-1}) \tag{6.18}$$

Similar to kernel Fisher discriminant analysis [111], the problem in (6.18) can be solved by eigen decomposition and the rows of $\mathbf{R}_c$ are the $\bar{m}$ leading eigen vectors of $(\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \mathbf{S}_c'\mathbf{S}_c)^{-1}(\mathbf{X}_c\mathbf{H}\mathbf{X}_c')$.

Update $\mathbf{S}_c$ when fixing $\mathbf{R}_c$ and $\mathbf{W}_c$: When fixing $\mathbf{R}_c$'s and $\mathbf{W}_c$'s, the problem in (6.9) reduces to the following problem:

$$\min_{\mathbf{S}_c} \frac{1}{2}\|\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{X}_c^s - \mathbf{Y}\|_F^2 + \lambda\sum_{c=1}^{C}\|\tilde{\mathbf{S}}_c\|_{2,1} - \frac{1}{2}\sum_{c=1}^{C}\|\mathbf{R}_c\mathbf{S}_c'\|_F^2 \tag{6.19}$$

The optimization problem in (6.19) is non-convex and thus only local optimum can be reached by using gradient descent algorithm. First, we derive the derivative of each term in (6.19) *w.r.t.* each $\mathbf{S}_c$ separately.

$$\mathbf{J}_1 = \frac{\partial \frac{1}{2}\| \sum_{c=1}^{C} \mathbf{W}_c \mathbf{S}_c \mathbf{X}_c^s - \mathbf{Y}\|_F^2}{\partial \mathbf{S}_c} = \mathbf{W}_c'(\sum_{c=1}^{C} \mathbf{W}_c \mathbf{S}_c \mathbf{X}_c^s - \mathbf{Y})\mathbf{X}_c^{s\prime}, \tag{6.20}$$

$$\mathbf{J}_2 = \frac{\partial \lambda \|\tilde{\mathbf{S}}_c\|_{2,1}}{\partial \mathbf{S}_c} = \lambda \mathbf{S}_c \circ \mathbf{D}_c, \tag{6.21}$$

where $\mathbf{D}_c \in \mathcal{R}^{\bar{m} \times 2\bar{K}d}$ is a matrix, in which each entry D_c^{ij} is set as $\frac{1}{\|\mathbf{S}_c^{ik}\|_2}$ if j belongs to the k-th component, with $\mathbf{S}_c^{i,k}$ denoting the k-th component in the i-th row of $\mathbf{S}_c$.

$$\mathbf{J}_3 = \frac{\partial -\frac{1}{2}\|\mathbf{R}_c \mathbf{S}_c'\|_F^2}{\partial \mathbf{S}_c} = -\mathbf{S}_c \mathbf{R}_c' \mathbf{R}_c. \tag{6.22}$$

In each iteration, we update each $\mathbf{S}_c$ when fixing all the other $\mathbf{S}_{\tilde{c}}$'s for $\tilde{c} \neq c$ by using the following equation:

$$\mathbf{S}_c \leftarrow \mathbf{S}_c - \eta(\mathbf{J}_1 + \mathbf{J}_2 + \mathbf{J}_3), \tag{6.23}$$

where η is the learning rate, which is empirically fixed as 0.0001 in our experiments. We repeat this process iteratively until the objective value of (6.19) converges. The whole algorithm is summarized in Algorithm 10.

In the testing stage, for each test sample $\mathbf{x}^t$ which contains the features $\mathbf{x}_c^t$'s obtained based on each category-specific GMM, we use $\sum_{c=1}^{C} \mathbf{W}_c \mathbf{S}_c \mathbf{x}_c^t$ to obtain the regression values and assign this test sample to the category corresponding to the maximum regression value.

6.5 Experiments

In this section, we demonstrate the effectiveness of our Domain Adaptive approach based on Fisher Vector (DAFV) for object recognition and human action recognition by conducting extensive experiments on two benchmark datasets Bing-Caltech256 and ACT4[2].

Algorithm 10 Domain Adaptation based on Fisher Vector (DAFV) Algorithm

Input: $\mathbf{X}_c^s, \mathbf{X}_c^t, \mathbf{Y}, \lambda, \gamma$
Initialize $\mathbf{S}_c$ as the PCA projection matrix on $\mathbf{X}_c$.
repeat
  repeat
    For c=1,...,C, update $\mathbf{W}_c$ using (6.12).
  until The objective of (6.11) converges.
  For c=1,...,C, update $\mathbf{R}_c$ by solving (6.18).
  repeat
    For c=1,...,C, update $\mathbf{S}_c$ using (6.23).
  until The objective of (6.19) converges.
until The objective of (6.9) converges.
Output: $\mathbf{W}_c$, $\mathbf{S}_c$.

We compare our method with state-of-the-art baselines, and also compare it with its two special cases to validate our MMD-based regularizer and group lasso based regularizer. Moreover, we take the object recognition task as an example to visually illustrate the domain invariant components of Fisher vectors selected by using our method.

6.5.1 Object Recognition

Experimental Settings: We use Bing-Caltech256 [9] dataset, which is commonly used to evaluate domain adaption methods for object recognition. Bing-Caltech256 dataset consists of the images from Caltech256 dataset and the images from Bing search engine distributed in 256 categories. Generally, Bing is treated as the source domain and Caltech-256 is treated as the target domain, because Bing images are collected by the search engine without having ground-truth labels and thus not appropriate for being used as test set. Following the setting in [68], we use the first 20 categories and set the number of source (*resp.*, target) domain examples per category to be 50 (*resp.*, 25) based on the train/test split provided in [9].

In order to generate local descriptors for each image, we first use selection search [148]

to generate object proposals. Then, we use the output of the 6-th layer of AlexNet [84] as the 4096-dim feature for each proposal with the pretrained model in [77]. After reducing the dimension of proposal features to 200 by using Principle Component Analysis (PCA), we use the proposals from the source domain within each category to train an 8-component Gaussian Mixture Model (GMM), which leads to a total of 160 components for all categories. Finally, we encode each image, which is a bag of 200-dim proposal features, as a $64,000$-dim Fisher vector based on the trained GMMs.

Baselines: We compare our DAFV method with two sets of baselines: domain adaptation baselines and GMM based baselines. We also include Regularized Least Square (RLS) as a baseline. For domain adaptation baselines, we compare our method with feature-based methods GFK [60], SGF [62], SA [53], DIP [7], TCA [119], LSSA [3], the SVM-based method DASVM [39], the instance reweighting method KMM [71], the dictionary learning method SDDL [136], and the low-rank based method LTSL [133]. Note that for feature-based methods [3, 7, 53, 60, 62, 119], we first obtain the transformed features by employing their methods suggested in the original papers [3, 7, 53, 60, 62, 119] and then use the transformed features as input features for RLS.

For GMM based baselines AGMM [36] and EM_RGMM [81], we use different approaches to obtain GMMs, which is explained as follows,

- AGMM [36]: We first train a 160-component GMM by using proposals from the source domain, and then adapt this GMM using the proposals from the target domain. Based on the GMM on the source domain and the adapted GMM, we extract two sets of Fisher vectors for all images from both domains. Based on these two sets of Fisher vectors, we train regressors and obtain the regression values of test images separately, and finally use the average fusion of two sets of regression values for prediction.

- EM_RGMM [81]: We train two 160-component GMMs based on the proposals from the source domain and the target domain, separately. Then, we calculate the interpolated GMM between the two GMMs. Based on the interpolated GMM, we extract Fisher vectors for all images from both domains. Finally, we train regressors and predict the test images based on the extracted Fisher vectors.

Table 6.1: Accuracies (%) of RLS and GMM based baselines, as well as our DAFV method and its two special cases for object recognition. The best result is denoted in boldface

RLS	AGMM	EM_RGMM	DAFV_sim1	DAFV_sim2	DAFV
73.2	76.8	77.4	75.4	77.8	**79.4**

Table 6.2: Accuracies (%) of domain adaptation baselines and our DAFV method for object recognition. The best result is denoted in boldface

KMM	DASVM	GFK	SGF	SA	DIP	TCA	LSSA	SDDL	LTSL	DAFV
73.6	75.8	73.6	74.4	74.2	71.8	74.8	77.8	62.4	77.6	**79.4**

Moreover, in order to validate our MMD-based regularizer and group lasso based regularizer, we compare our method with its two simplified versions. Specifically, we remove the group lasso based regularizer $\sum_{c=1}^{C} \|\tilde{\mathbf{S}}_c\|_{2,1}$ in (6.9) by setting the parameter λ as 0 and refer to this special case as DAFV_sim2. Based on DAFV_sim2, we further remove the MMD-based regularizer $\|\frac{1}{n_s}\mathbf{R}_c\mathbf{X}_c^s\mathbf{1} - \frac{1}{n_t}\mathbf{R}_c\mathbf{X}_c^t\mathbf{1}\|^2$ and denote this special case as DAFV_sim1.

We use accuracy for performance evaluation. Two trade-off parameters γ and λ in (6.9) are empirically set as 1000 and 10 for our DAFV method. For the baseline methods, we choose their optimal parameters based on their accuracies on the test dataset.

Experimental Results: We report the results of RLS, the GMM based baselines, and our DAFV method including its two special cases in Table 6.1, from which we observe that AGMM and EM_RGMM achieve better results than RLS, suggesting the benefits of adapting or interpolating GMMs. We also observe that our DAFV method outperforms DAFV_sim2, which validates the effectiveness of selecting some components of Fisher vectors by using group lasso based regularizer. Additionally, DAFV_sim2 outperforms DAFV_sim1, which validates our MMD based regularizer. Finally, our DAFV method outperforms the GMM based baselines, which shows its effectiveness on reducing domain distribution mismatch between the source domain and the target domain.

Moreover, we report the results of domain adaptation baselines in Table 6.2 and also include the result of our DAFV method for comparison. From Table 6.2, we observe that

Figure 6.1: The top object proposals belonging to the selected Gaussian model for the "beer-mug" category from the Bing dataset

the domain adaptation baselines are generally better than RLS reported in Table 6.1. The results validate the effectiveness of employing different strategies to address the domain shift issue. However, all the domain adaptation baselines are worse than our DAFV method. One possible explanation is that we select the domain invariant components of Fisher vectors, which is designed for Fisher vectors.

Discussion on Domain Invariant Components: As discussed in Section 6.1, the motivation of our DAFV method is that each Gaussian model in the GMM represents the data distribution of a cluster of local descriptors and corresponds to one component in the encoded Fisher vector. Assuming that there exist some Gaussian models representing common distribution shared by both source and target domain, the corresponding components of Fisher vectors should be more domain invariant. The benefit of selecting domain invariant components has been demonstrated in Table 6.1 and Table 6.2, and now we provide some intuitive examples to illustrate the domain invariant components.

First, recall that we train C category-specific GMMs and $\mathbf{S}_c \in \mathcal{R}^{\bar{m}\times 2\bar{K}d}$ is the transformation matrix corresponding to the c-th GMM. For the c-th category, we compute the L_2 norm for each component in each row of $\mathbf{S}_c$, which corresponds to one Gaussian model in the c-th GMM. Then, we sum the computed values over different rows and choose the component with the maximum value, which corresponds to the selected Gaussian model in the c-th GMM. Because there are probabilities $\gamma_i(k)$'s that the i-th proposal belongs to the k-th Gaussian model (see Section 6.3) when training a GMM, we can easily pick out the top proposals that belong to the cluster corresponding to the selected Gaussian model. Let us take the "beer-mug" category as an example to show the top proposals for the selected Gaussian model in Fig 6.1, from which we have an interesting observation

Table 6.3: Accuracies (%) of RLS and GMM based baselines, as well as our DAFV method and its two special cases for human action recognition. The best results on each setting are denoted in boldface

Setting	RLS	AGMM	EM_RGMM	DAFV_sim1	DAFV_sim2	DAFV
1->2	69.94	72.36	73.72	71.00	72.96	**74.92**
1->3	44.11	46.07	46.22	45.02	46.68	**48.49**
1->4	77.64	80.21	80.06	81.27	82.33	**83.99**
2->1	74.17	77.95	74.02	77.04	77.64	**79.61**
2->3	67.37	67.52	67.82	69.94	71.00	**72.96**
2->4	60.88	61.03	61.18	60.57	62.24	**63.90**
3->1	52.87	47.89	51.96	51.21	52.87	**55.74**
3->2	66.92	66.92	67.07	69.18	69.94	**71.90**
3->4	40.03	41.69	41.99	41.69	43.20	**45.47**
4->1	71.75	73.72	72.21	68.73	75.98	**76.13**
4->2	46.37	52.27	52.11	49.40	51.96	**53.92**
4->3	37.31	38.97	36.71	38.52	40.03	**41.69**
Avg	59.11	60.55	60.42	60.30	62.24	**64.06**

that the proposals are all near the handle of beer mug. We conjecture beer mugs from different domains are quite different in shape, color, and pattern of body regions, but the handle regions generally look similar as illustrated in Fig 6.1. Intuitively, the handle regions can be used to discriminate beer mugs against the other categories but are less variant across different domains. So the components of Fisher vectors corresponding to the selected Gaussian models are assigned larger weights, which is helpful for improving the performance of object recognition.

6.5.2 Human Action Recognition

Experimental Settings: We use the ACT4^2 [26] dataset for human action recognition. The ACT4^2 dataset contains videos from 14 categories of human actions, which are

Table 6.4: Accuracies (%) of domain adaptation baselines and our DAFV method for human action recognition. The best results on each setting are denoted in boldface

Setting	KMM	DASVM	GFK	SGF	SA	DIP	TCA	LSSA	SDDL	LTSL	DAFV
1->2	67.67	59.52	73.11	66.16	72.96	72.21	72.81	73.56	72.96	71.75	**74.92**
1->3	45.62	35.65	46.37	45.02	45.92	46.37	46.53	44.11	45.02	45.17	**48.49**
1->4	79.91	74.17	81.72	78.85	80.97	80.51	82.93	82.33	79.00	81.72	**83.99**
2->1	76.74	68.88	77.95	70.85	75.98	75.38	**79.76**	68.88	75.98	75.68	79.61
2->3	69.94	55.29	70.54	66.62	69.79	71.60	69.49	65.41	69.94	68.73	**72.96**
2->4	61.33	56.34	61.48	59.06	62.08	62.84	61.78	62.08	61.33	61.33	**63.90**
3->1	54.98	48.94	53.78	47.73	54.08	54.53	54.68	50.45	53.47	54.08	**55.74**
3->2	70.54	62.08	69.94	69.79	67.07	71.00	67.67	64.20	68.88	67.82	**71.90**
3->4	41.39	32.33	42.60	41.09	42.45	43.20	43.35	40.94	36.40	43.96	**45.47**
4->1	74.62	67.98	73.11	74.17	73.87	73.87	73.26	66.01	74.47	72.36	**76.13**
4->2	**54.83**	46.37	49.24	53.02	53.32	51.66	49.85	52.11	47.43	51.66	53.92
4->3	34.29	36.40	40.03	39.43	38.97	40.33	39.27	39.73	37.61	38.97	**41.69**
Avg	60.99	53.66	61.66	59.32	61.46	61.96	61.78	59.15	60.21	61.10	**64.06**

captured from 4 camera viewpoints. Following [26], we use a subset with 2648 RGB videos from all 4 viewpoints. We treat one view as the source domain and another different view as the target domain, which results in totally 12 settings.

Following [155], we use the source codes provided in [155] to extract four types of Improved Dense Trajectory (IDT) descriptors (i.e., 30-dim trajectories, 96-dim HOG, 108-dim HOF, and 192-dim MBH). Following [155], we first reduce the dimension of descriptors by a factor of two using PCA. Then, we use the descriptors from the videos in the source domain within each category to train an 8-component GMM, which leads to totally 112 components for all categories. Finally, we encode each video, which is a bag of 213-dim IDT descriptors, as a 47712-dim Fisher vector based on the trained GMMs.

Baselines: We compare our DAFV method with the same baselines as discussed in Section 6.5.1. The only difference is that we train 112-component GMMs for AGMM and EM_RGMM. For the human action recognition task, accuracy is still used for perfor-

mance evaluation. Our DAFV method employs the same parameters as used for object recognition while optimal parameters of the baseline methods are chosen according to their accuracies on the test dataset.

Experimental Results: We report the experimental results of RLS and GMM based baselines, as well as our DAFV method and its two special cases on 12 settings in Table 6.3. From the results, we can draw similar conclusions as those for object recognition in Section 6.5.1. In particular, the comparisons among our DAFV method and its two special cases clearly demonstrate the effectiveness of our group lasso based regularizer and the MMD-based regularizer. Moreover, our DAFV method is better than the GMM based baselines on all settings. The results again demonstrate that the recognition performance can be improved by reducing domain distribution mismatch.

Table 6.4 shows the results of domain adaptation baselines. It can be seen that the average accuracies of the domain adaptation baselines are better than that of RLS reported in Table 6.3 except DASVM, which indicates the advantage of coping with domain difference by using various methods. While TCA (*resp.*, KMM) is better than our DAFV method on the setting 2->1 (*resp.*, 4->2), our method achieves the best results on 10 out of 12 settings. Moreover, in terms of the average accuracy over 12 settings, our DAFV method is the best, which again demonstrates it is helpful to address the domain shift issue by selecting domain invariant components of Fisher vectors.

6.6 Summary

In this chapter, we have proposed a Domain Adaptation method based on Fisher Vector (DAFV), which is designed for Fisher vectors. Based on the assumption that some Gaussian models in the GMM can better capture the common data distribution between the source domain and the target domain, our DAFV method is designed to select the domain invariant components of Fisher vectors corresponding to the common Gaussian models and simultaneously solve a regression problem. The effectiveness of our DAFV method for visual recognition has been demonstrated by extensive experiments on two benchmark datasets.

Chapter 7

Conclusion and Future Work

Domain adaptation and domain generalization are very important research topics for visual recognition and have attracted a large number of researchers. In this thesis, we study the domain adaptation and domain generalization problems when learning from web data or well labeled data. When learning from web data, we propose both a domain adaptation framework and a domain generalization method which can handle the label noise of web data, utilize the textual information (*i.e.*, privileged information) associated with the web data, and simultaneously address the domain issue between web training data and test data. For learning from well labeled data, we propose a multi-view domain generalization approach when both training data and test data are associated with multiple types of features, and also specifically design a domain adaptation approach for Fisher vectors. In this chapter, we summarize our proposed methods and also discuss several promising future research directions.

7.1 Conclusion

We summarize our contributions to domain adaptation and generalization for visual recognition as follows,

- We have proposed a MIL-PI framework for visual recognition, which can utilize the textual information associated with the loosely labeled training web and simultaneously handle the label noise in the training web data. We have also extended our MIL-PI framework to MIL-PI-DA framework, which can further reduce the data distribution mismatch between the web training data and test data. The

effectiveness of our proposed frameworks have been clearly demonstrated by the comprehensive experiments on action and event recognition.

- We have proposed a novel weakly supervised domain generalization approach WSDG by learning loosely labeled web data for visual recognition. Our WSDG method can cope with the label noise and generalize the source classifier to arbitrary target domain at the same time. Moreover, our WSDG approach has been extended to WSDG-PI by further utilizing additional textual descriptions associated with the training web data as privileged information. Comprehensive experiments on object recognition and event recognition have indicated the effectiveness of our WSDG and WSDG-PI approaches.

- We have proposed a multi-view domain generalization framework named EMVDG, which builds upon exemplar SVM classifiers. By fusing multiple exemplar SVM classifiers in the testing stage, the integrated classifier is generalizable to any unseen target domain. Moreover, we exploit the relation among multiple types of features. We have also extended our EMVDG framework to a multi-view domain adaptation framework named EMVDA when the unlabeled target domain data are available in the training stage. Extensive experiments on object recognition and human action recognition have shown the effectiveness of our EMVDG and EMVDA frameworks.

- We have proposed a Domain Adaptation method based on Fisher Vector (DAFV) based on Fisher vectors, which can learn the project matrix to select the domain invariant components of Fisher vectors and simultaneously learn the regression matrix for visual recognition. The effectiveness of our DAFV method has been demonstrated by comprehensive experiments on object recognition and human action recognition.

7.2 Future Work

In the following, we propose three possible research directions for future investigation.

- Although tons of works have been done on domain adaptation and generalization, the concept of "domain" is still very ambiguous because there is no clear definition *w.r.t.* what the "domain" is. In our experiments, we can treat the data from different datasets, collected in different environments, or captured from different camera viewpoints as different domains. In the future, we may be specific about the definition of domain and what to transfer from the source domain to the target domain, and then discover a domain adaption or generalization technique that can be well applied to all the specific situations and achieve satisfactory performance.

- In our previous works, we build our self-designed algorithm upon deep learning features. Specifically, we use the output of intermediate layers of deep learning as features, and then apply our own algorithms on the extracted features. However, deep learning has achieved state-of-the-art results for more and more computer vision applications with an end-to-end system. Moreover, lots of research has been done on webly supervised deep learning [35], multi-view deep learning [153], and deep learning for domain adaptation [58]. In the future, we may unify our previous technics (*e.g.*, weakly supervised learning, multi-view learning, learning using privileged information, and transfer learning) with deep learning more coherently and elegantly by directly modifying the neural network structure to generate an end-to-end system for visual recognition.

- In our previous works for learning from web data, we utilize the privilege information associated with the web images/videos to assist in coping with the label noise. However, the associated privilege information is generally quite noisy and not always reliable. To better leverage the privilege information, we can develop some more advanced techniques to suppress the noise in privilege information or select the valid useful information from privilege information, which will be left for future investigation.

References

[1] P-A Absil and André L Tits. Newton-KKT interior-point methods for indefinite quadratic programming. *Computational Optimization and Applications*, 36(1):5–41, 2007.

[2] Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16, 2011.

[3] Rahaf Aljundi, Rémi Emonet, Damien Muselet, and Marc Sebban. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In *CVPR*, 2015.

[4] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

[5] Peter Auer. On learning from multi-instance examples: Empirical evaluation of a theoretical approach. In *ICML*, 1997.

[6] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, 2004.

[7] Mahsa Baktashmotlagh, Mehrtash Harandi, and Mathieu Salzmann Brian Lovell. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, 2013.

[8] Yoshua Bengio et al. Deep learning of representations for unsupervised and transfer learning. *ICML*, 2012.

[9] Alessandro Bergamo and Lorenzo Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *NIPS*, 2010.

[10] Matthew B Blaschko, Christoph H Lampert, and Arthur Gretton. Semi-supervised laplacian regularization of kernel canonical correlation analysis. In *ECML PKDD*, 2008.

[11] John Blitzer, Sham Kakade, and Dean P Foster. Domain adaptation with coupled subspaces. In *AISTATS*, 2011.

[12] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[13] Aaron F Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 352(1358):1257–1265, 1997.

[14] Jakramate Bootkrajang and Ata Kabán. Learning kernel logistic regression in the presence of class label noise. *Pattern Recognition*, 47(11):3641–3655, 2014.

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

[16] L. Bruzzone and M. Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *T-PAMI*, 32:770–787, 2010.

[17] Razvan C Bunescu and Raymond J Mooney. Multiple instance learning for sparse positive bags. In *ICML*, 2007.

[18] 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.

[19] Rui Caseiro, Joao F Henriques, Pedro Martins, and Jorge Batista. Beyond the shortest path: Unsupervised domain adaptation by sampling subspaces along the spline flow. In *CVPR*, 2015.

[20] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *T-IST*, 2(3):27:1–27:27, 2011.

[21] Shih-Fu Chang, Dan Ellis, Wei Jiang, Keansub Lee, Akira Yanagawa, Er C. Loui, and Jiebo Luo. Large-scale multimodal semantic concept detection for consumer video. In *SIGMM*, 2007.

[22] Lin Chen, Lixin Duan, and Dong Xu. Event recognition in videos by learning from heterogeneous web sources. In *CVPR*, 2013.

[23] Xinlei Chen, A. Shrivastava, and A. Gupta. NEIL: Extracting visual knowledge from web data. In *ICCV*, 2013.

[24] Yixin Chen, Jinbo Bi, and James Ze Wang. Miles: Multiple-instance learning via embedded instance selection. *T-PAMI*, 28(12):1931–1947, 2006.

[25] Li Cheng and Sinno Jialin Pan. Semi-supervised domain adaptation on manifolds. *T-NNLS*, 25(12):2240–2249, 2014.

[26] Zhongwei Cheng, Lei Qin, Yituo Ye, Qingming Huang, and Qi Tian. Human daily action analysis with multi-view and color-depth data. In *ECCV*, 2012.

[27] S Chopra, S Balakrishnan, and R Gopalan. DLID: Deep learning for domain adaptation by interpolating between domains. In *ICML*, 2013.

[28] Wen-Sheng Chu, Fernando DelaTorre, and Jeffery Cohn. Selective transfer machine for personalized facial action unit detection. In *CVPR*, 2013.

[29] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[30] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2002.

[31] Hal Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[33] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997.

[34] Zhengming Ding and Yun Fu. Low-rank common subspace for multi-view learning. In *ICDM*, 2014.

[35] Santosh K Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014.

[36] Mandar Dixit, Nikhil Rasiwasia, and Nuno Vasconcelos. Adapted Gaussian models for image classification. In *CVPR*, 2011.

[37] Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *CVPR*, 2013.

[38] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[39] L. Duan, Ivor W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *T-PAMI*, 34:465–479, 2012.

[40] L. Duan, Ivor W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *T-PAMI*, 34:465–479, 2012.

[41] L. Duan, Ivor W. Tsang, D. Xu, and Jiebo Luo. Visual event recognition in videos by learning from web data. *T-PAMI*, 34:1667–1680, 2012.

[42] L. Duan, Ivor W. Tsang, D. Xu, and Jiebo Luo. Visual event recognition in videos by learning from web data. *T-PAMI*, 34:1667–1680, Sep. 2012.

[43] L. Duan, Ivor W. Tsang, D. Xu, and S. J. Maybank. Domain transfer SVM for video concept detection. In *CVPR*, 2009.

[44] Lixin Duan, Dong Xu, and Shi-Fu Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *CVPR*, 2012.

[45] Lixin Duan, Dong Xu, and Ivor W. Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. *T-NNLS*, 23(3):504–518, 2012.

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

[47] Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working set selection using second order information for training support vector machines. *JMLR*, 6:1889–1918, 2005.

[48] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[49] Jason Farquhar, David Hardoon, Hongying Meng, John S Shawe-taylor, and Sandor Szedmak. Two view learning: SVM-2K, theory and practice. In *NIPS*, 2005.

[50] Jason Farquhar, Sandor Szedmak, Hongying Meng, and John Shawe-Taylor. Improving bag-of-keypoints image categorisation: Generative models and pdf-kernels. Technical report, University of Southampton, 2005.

[51] Li Fe-Fei, Rob Fergus, and Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, 2003.

[52] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *ICCV*, 2005.

[53] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[54] Vittorio Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007.

[55] Shereen Fouad, Peter Tino, Somak Raychaudhury, and Petra Schneider. Incorporating privileged information through metric learning. *T-NNLS*, 24(7):1086–1098, 2013.

[56] Thomas Gärtner, Peter A Flach, Adam Kowalczyk, and Alexander J Smola. Multi-instance kernels. In *ICML*, 2002.

[57] Peter Vincent Gehler and Sebastian Nowozin. Infinite kernel learning. Technical report, Max Planck Institute for Biological Cybernetics, 2008.

[58] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 2011.

[59] Boqing Gong, Kristen Grauman, and Fei Sha. Reshaping visual datasets for domain adaptation. In *NIPS*, 2013.

[60] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[61] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*. 2014.

[62] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[63] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[64] A. Gretton, K. M. KBorgwVijayanarasimhan2008ardt, M. J. Rasch, B. Schlkopf, and A. Smola. A kernel two-sample test. *JMLR*, 13:723–773, 2012.

[65] David Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[66] James Heckman. Sample selection bias as a specification error, 2013.

[67] Minh Hoai and Andrew Zisserman. Discriminative sub-categorization. In *CVPR*, 2013.

[68] Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. *ICLR*, 2013.

[69] Judy Hoffman, Kate Saeko, Brian Kulis, and Trevor Darrell. Discovering latent domains for multisource domain adaptation. In *ECCV*. 2012.

[70] Yuxiao Hu, Liangliang Cao, Fengjun Lv, Shuicheng Yan, Yihong Gong, and Thomas S Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *ICCV*, 2009.

[71] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, 2006.

[72] S. J. Hwang and K. Grauman. Learning the relative importance of objects from tagged images for retrieval and cross-modal search. *IJCV*, 100(2):134–153, 2012.

[73] Alexandros Iosifidis, Anastasios Tefas, and Ioannis Pitas. View-invariant action recognition based on artificial neural networks. *T-NNLS*, 23(3):412–424, 2012.

[74] Tommi S Jaakkola, David Haussler, et al. Exploiting generative models in discriminative classifiers. *NIPS*, 1999.

[75] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data AnalysisD*.

[76] I-Hong Jhuo, Dong Liu, DT Lee, Shih-Fu Chang, et al. Robust visual domain adaptation with low-rank reconstruction. In *CVPR*, 2012.

[77] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[78] Yu-Gang Jiang, Subhabrata Bhattacharya, Shih-Fu Chang, and Mubarak Shah. High-level event recognition in unconstrained videos. *JMIR*, 2(2):73–101, 2013.

[79] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel Ellis, and Alexander C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*, 2011.

[80] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*. 2012.

[81] Hyunwoo J Kim, Nagesh Adluru, Monami Banerjee, Baba C Vemuri, and Vikas Singh. Interpolation on the manifold of K component GMMs. In *ICCV*, 2015.

[82] Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. Lp-norm multiple kernel learning. *JMLR*, 12:953–997, 2011.

[83] Marius Klotf, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. Lp-norm multiple kernel learning. *JMLR*, 12:953–997, 2011.

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[85] Hildegard Kuehne, Hueihan Jhuang, Est'ibaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[86] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

[87] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

[88] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72, 2004.

[89] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[90] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[91] Thomas Leung, Yang Song, and John Zhang. Handling label noise in video classification via multiple instance learning. In *ICCV*, 2011.

[92] Quannan Li, Jiajun Wu, and Zhuowen Tu. Harvesting mid-level visual concepts from large-scale internet images. In *CVPR*, 2013.

[93] W. Li, L. Duan, Ivor W. Tsang, and D. Xu. Batch mode adaptive multiple instance learning for computer vision tasks. In *CVPR*, 2012.

[94] W. Li, L. Duan, Ivor W. Tsang, and D. Xu. Co-labeling: A new multi-view learning approach for ambiguous problems. In *ICDM*, 2012.

[95] Wen Li, Lixin Duan, Dong Xu, and I Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *T-PAMI*, 36(6):1134–1148, 2013.

[96] Wen Li, Lixin Duan, Dong Xu, and Ivor W. Tsang. Text-based image retrieval using progressive multi-instance learning. In *ICCV*, 2011.

[97] Wen Li, Lixin Duan, Dong Xu, and Ivor W. Tsang. Text-based image retrieval using progressive multi-instance learning. In *ICCV*, 2011.

[98] Wen Li, Li Niu, and Dong Xu. Exploiting privileged information from web data for image categorization. In *ECCV*. 2014.

[99] Yu-Feng Li, James T Kwok, Ivor W Tsang, and Zhi-Hua Zhou. A convex method for locating regions of interest with multi-instance learning. In *ECML PKDD*. 2009.

[100] Yu-Feng Li, Ivor W. Tsang, James Tin-Yau Kwok, and Zhi-Hua Zhou. Tighter and convex maximum margin clustering. In *AISTATS*, 2009.

[101] Lichen Liang, Feng Cai, and Vladimir Cherkassky. Predictive learning with structured (grouped) data. *Neural Networks*, 22:766–773, 2009.

[102] Zhe Lin, Zhuolin Jiang, and Larry S Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009.

[103] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

[104] Lingqiao Liu, Chunhua Shen, Lei Wang, Anton van den Hengel, and Chao Wang. Encoding high dimensional local features by sparse coding based Fisher vectors. In *NIPS*, 2014.

[105] Philip M Long and Lei Tan. Pac learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *ML*, 30(1):7–21, 1998.

[106] Alexander C. Loui, Jiebo Luo, Shih-Fu Chang, Dan Ellis, Wei Jiang, Lyndon Kennedy, Keansub Lee, and Akira Yanagawa. Kodak's consumer video benchmark data set. In *SIGMM*, 2007.

[107] Alexander C. Loui, Jiebo Luo, Shih-Fu Chang, Dan Ellis, Wei Jiang, Lyndon Kennedy, Keansub Lee, and Akira Yanagawa. Kodak's consumer video benchmark data set. In *SIGMM*, 2007.

[108] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of Exemplar-SVMs for object detection and beyond. In *ICCV*, 2011.

[109] Vlad I Morariu and Larry S Davis. Multi-agent event recognition in structured scenarios. In *CVPR*, 2011.

[110] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013.

[111] Klaus-Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf. An introduction to kernel-based learning algorithms. *T-NN*, 12(2):181–201, 2001.

[112] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NIPS*, 2013.

[113] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008.

[114] L Niu, W Li, D Xu, and J Cai. Visual recognition by learning from web data via weakly supervised domain generalization. *T-NNLS*, pages 1–15, 2016.

[115] Li Niu, Jianfei Cai, and Dong Xu. Domain adaptive fisher vector for visual recognition. In *ECCV*, 2016.

[116] Li Niu, Wen Li, and Dong Xu. Multi-view domain generalization for visual recognition. In *ICCV*, 2015.

[117] Li Niu, Wen Li, and Dong Xu. Visual recognition by learning from web data: A weakly supervised domain generalization approach. In *CVPR*, 2015.

[118] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Action and event recognition with Fisher vectors on a compact feature set. In *ICCV*, 2013.

[119] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *T-NN*, 22:199–210, Feb. 2011.

[120] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *T-NN*, 22(2):199–210, 2011.

[121] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *T-KDE*, 22(10):1345–1359, 2010.

[122] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *Signal Processing Magazine*, 32(3):53–69, 2015.

[123] Xiaojiang Peng, Changqing Zou, Yu Qiao, and Qiang Peng. Action recognition with stacked Fisher vectors. In *ECCV*. 2014.

[124] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[125] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*. 2010.

[126] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, April 1998.

[127] John Platt et al. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods*, 3, 1999.

[128] Qiang Qiu, Vishal Patel, Pavan Turaga, and Rama Chellappa. Domain adaptive dictionary learning. In *ECCV*, 2012.

[129] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, 2007.

[130] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, 2010.

[131] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. *T-PAMI*, 33(4):754–766, 2011.

[132] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004.

[133] Ming Shao, Dmitry Kit, and Yun Fu. Generalized transfer subspace learning through low-rank constraint. *IJCV*, 109(1-2):74–93, 2014.

[134] Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. In *CVPR*, 2012.

[135] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H. Lampert. Learning to rank using privileged information. In *ICCV*, 2013.

[136] Sumit Shekhar, Vishal Patel, Hien Nguyen, and Rama Chellappa. Generalized domain-adaptive dictionaries. In *CVPR*, 2013.

[137] Yifan Shi, Yan Huang, David Minnen, Aaron Bobick, and Irfan Essa. Propagation networks for recognition of partially ordered sequential action. In *CVPR*, 2004.

[138] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *JSPI*, 90(2):227–244, 2000.

[139] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Data-driven visual similarity for cross-domain image matching. In *ACM Transactions on Graphics*, volume 30, page 154, 2011.

[140] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Fisher networks for large-scale image classification. In *NIPS*, 2013.

[141] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *ICML*, 2005.

[142] Masashi Sugiyama and Klaus-Robert Müller. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4/2005):249–279, 2005.

[143] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, T-S Chua, and Jintao Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.

[144] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[145] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *T-PAMI*, 30(11):1958–1970, 2008.

[146] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010.

[147] Son D Tran and Larry S Davis. Event modeling and recognition using markov logic networks. In *ECCV*. 2008.

[148] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[149] Tiberio Uricchio, Marco Bertini, Lorenzo Seidenari, and Alberto Bimbo. Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging. In *ICCV*, 2015.

[150] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.

[151] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.

[152] S. Vijayanarasimhan and K. Grauman. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In *CVPR*, 2008.

[153] Anran Wang, Jiwen Lu, Jianfei Cai, Tat-Jen Cham, and Gang Wang. Large-margin multi-modal deep learning for rgb-d object recognition. *T-MM*, 17(11):1887–1898, 2015.

[154] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[155] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[156] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[157] Liang Wang, Yizhou Wang, and Wen Gao. Mining layered grammar rules for action recognition. *IJCV*, 2011.

[158] Xinggang Wang, Baoyuan Wang, Xiang Bai, Wenyu Liu, and Zhuowen Tu. Max-margin multiple-instance dictionary learning. In *ICML*, 2013.

[159] Yunchao Wei, Wei Xia, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. CNN: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*, 2014.

[160] Caiming Xiong, Scott McCloskey, Shao-Hang Hsieh, and Jason J Corso. Latent domains modeling for visual domain adaptation. In *AAAI*, 2014.

[161] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[162] Dong Xu and Shih-Fu Chang. Video event recognition using kernel methods with multilevel temporal alignment. *T-PAMI*, 30(11):1985–1997, 2008.

[163] X Xu, W Li, and D Xu. Distance metric learning using privileged information for face verification and person re-identification. *T-NNLS*, (99):1–1, 2015.

[164] Zheng Xu, Wen Li, Li Niu, and Dong Xu. Exploiting low-rank structure from latent domains for domain generalization. In *ECCV*. 2014.

[165] Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM MM*, 2007.

[166] Pei Yang and Wei Gao. Multi-view discriminant transfer learning. In *IJCAI*, 2013.

[167] Guangnan Ye, Dong Liu, I-Hong Jhuo, and Shih-Fu Chang. Robust late fusion with rank minimization. In *CVPR*, 2012.

[168] Gang Yu, Zicheng Liu, and Junsong Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *ACCV*, 2014.

[169] Tsz-Ho Yu, Tae-Kyun Kim, and Roberto Cipolla. Real-time action recognition by spatiotemporal semantic and structural forests. In *BMVC*, 2010.

[170] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[171] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *ICML*, 2004.

[172] Zhi Zeng and Qiang Ji. Knowledge based activity recognition with dynamic bayesian network. In *ECCV*. 2010.

[173] Joaquin Zepeda and Patrick Perez. Exemplar svms as visual feature encoders. In *CVPR*, 2015.

[174] Dan Zhang, Jingrui He, Yan Liu, Luo Si, and Richard D. Lawrence. Multi-view transfer learning with a large margin approach. In *SIGKDD*, 2011.

[175] Dan Zhang, Fei Wang, Luo Si, and Tao Li. M3IC: Maximum margin multiple instance clustering. In *IJCAI*, 2009.

[176] Min-Ling Zhang and Zhi-Hua Zhou. M3MIML: A maximum margin method for multi-instance multi-label learning. In *ICDM*, 2008.

[177] Qi Zhang, Sally A Goldman, Wei Yu, and Jason E Fritts. Content-based image retrieval using multiple-instance learning. In *ICML*, 2002.

[178] Xi Zhou, Na Cui, Zhen Li, Feng Liang, and Thomas S Huang. Hierarchical Gaussianization for image classification. In *ICCV*, 2009.

[179] Z. Zhou and M. Zhang. Multi-instance multi-label learning with application to scene classification. In *NIPS*, 2006.

[180] Guangyu Zhu, Ming Yang, Kai Yu, Wei Xu, and Yihong Gong. Detecting video events based on action recognition in complex scenes using spatio-temporal descriptor. In *ACM MM*, 2009.

[181] Xiaojin Zhu. Semi-supervised learning. In *Encyclopedia of machine learning*, pages 892–897. 2011.

Appendix

Appendix A: Proof of Proposition 1

Proof: By introducing the dual variables $\hat{\boldsymbol{\alpha}} = [\alpha_1, \ldots, \alpha_{L^+}]' \in \mathbb{R}^{L^+}$ for the constraints in (3.18), $\bar{\boldsymbol{\alpha}} = [\alpha_{L^+ +1}, \ldots, \alpha_m]' \in \mathbb{R}^{m-L^+}$ for the constraints (3.19), $\hat{\boldsymbol{\beta}} = [\beta_1, \ldots, \beta_{L^+}]' \in \mathbb{R}^{L^+}$ for the constraints in (3.20), $\bar{\boldsymbol{\beta}} = [\beta_{L^+ +1}, \ldots, \beta_m]' \in \mathbb{R}^{m-L^+}$ for the constraints in (3.21), and $\boldsymbol{\zeta} = [\nu_1, \ldots, \nu_m]'$ for the constraints in (3.22), we arrive at its Lagrangian as follows:

$$
\begin{aligned}
\mathcal{L} = \quad & \frac{1}{2}\left(\|\mathbf{w}\|^2 + \gamma\|\tilde{\mathbf{w}}\|^2\right) + C_1 \sum_{i=1}^{L^+} (\tilde{\mathbf{w}}'\tilde{\mathbf{z}}_i^s + \tilde{b}) \\
& + \sum_{i=L^+ +1}^{m} \eta_i + \frac{\lambda}{2}\|\hat{\mathbf{w}} - \rho\mathbf{v}\|^2 + C_2 \sum_{i=1}^{m} (\hat{\mathbf{w}}'\mathbf{z}_i^s + \hat{b}) \\
& - \sum_{i=1}^{L^+} \hat{\alpha}_i (\mathbf{w}'\mathbf{z}_i^s + b - p_i + \tilde{\mathbf{w}}'\tilde{\mathbf{z}}_i^s + \tilde{b} + \hat{\mathbf{w}}'\mathbf{z}_i^s + \hat{b}) \\
& - \sum_{i=L^+ +1}^{m} \bar{\alpha}_i (-\mathbf{w}'\mathbf{z}_i^s - b - 1 + \eta_i + \hat{\mathbf{w}}'\mathbf{z}_i^s + \hat{b}) \\
& - \sum_{i=1}^{L^+} \hat{\beta}_i (\tilde{\mathbf{w}}'\tilde{\mathbf{z}}_i^s + \tilde{b}) - \sum_{i=L^+ +1}^{m} \bar{\beta}_i \eta_i - \sum_{i=1}^{m} \nu_i (\hat{\mathbf{w}}'\mathbf{z}_i^s + \hat{b}),
\end{aligned}
\tag{7.1}
$$

Let us define $\boldsymbol{\alpha} = [\hat{\boldsymbol{\alpha}}', \bar{\boldsymbol{\alpha}}']'$, $\boldsymbol{\beta} = [\hat{\boldsymbol{\beta}}', \bar{\boldsymbol{\beta}}']'$, $\mathbf{Z} = [\mathbf{z}_1^s, \ldots, \mathbf{z}_m^s]$, $\tilde{\mathbf{Z}} = [\tilde{\mathbf{z}}_1^s, \ldots, \tilde{\mathbf{z}}_{L^+}^s]$, and $\mathbf{y} = [\mathbf{1}'_{L^+}, -\mathbf{1}'_{m-L^+}]'$, then the derivatives of the Lagrangian w.r.t. $\mathbf{w}, b, \tilde{\mathbf{w}}, \tilde{b}, \hat{\mathbf{w}}, \hat{b}, \boldsymbol{\eta}$ can

be obtained as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= \mathbf{w} - \mathbf{Z}(\boldsymbol{\alpha} \circ \mathbf{y}), \\
\frac{\partial \mathcal{L}}{\partial b} &= -\boldsymbol{\alpha}'\mathbf{y}, \\
\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{w}}} &= \gamma\tilde{\mathbf{w}} - \tilde{\mathbf{Z}}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1}_{L^+}), \\
\frac{\partial \mathcal{L}}{\partial \tilde{b}} &= -\mathbf{1}'_{L^+}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1}_{L^+}), \\
\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{w}}} &= \lambda\hat{\mathbf{w}} - \lambda\rho\mathbf{v} - \mathbf{Z}(\boldsymbol{\alpha} + \boldsymbol{\zeta} - C_2\mathbf{1}_m), \\
\frac{\partial \mathcal{L}}{\partial \hat{b}} &= -\mathbf{1}'_m(\boldsymbol{\alpha} + \boldsymbol{\zeta} - C_2\mathbf{1}), \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{\eta}} &= \mathbf{1}_{m-L^+} - \bar{\boldsymbol{\alpha}} - \bar{\boldsymbol{\beta}}.
\end{aligned}$$

By setting those derivatives to zeros, we have the following equations:

$$\mathbf{w} = \mathbf{Z}(\boldsymbol{\alpha} \circ \mathbf{y}), \tag{7.2}$$

$$\tilde{\mathbf{w}} = \frac{1}{\gamma}\tilde{\mathbf{Z}}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1}_{L^+}), \tag{7.3}$$

$$\hat{\mathbf{w}} = \rho\mathbf{v} + \frac{1}{\lambda}\mathbf{Z}(\boldsymbol{\alpha} + \boldsymbol{\zeta} - C_2\mathbf{1}_m), \tag{7.4}$$

as well as the following constraints, $\boldsymbol{\alpha}'\mathbf{y} = 0$, $\mathbf{1}'_{L^+}(\hat{\boldsymbol{\alpha}}+\hat{\boldsymbol{\beta}}-C_1\mathbf{1}_{L^+}) = 0$, $\mathbf{1}'_m(\boldsymbol{\alpha}+\boldsymbol{\zeta}-C_2\mathbf{1}_m) = 0$, $\bar{\boldsymbol{\alpha}} \leq \mathbf{1}_{m-L^+}$. Substituting the equations (7.2), (7.3) and (7.4) into (7.1) and considering $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta} \geq \mathbf{0}$, we obtain the following dual form,

$$\begin{aligned}
\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\zeta}} \quad & -\mathbf{p}'\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{K} \circ \mathbf{y}\mathbf{y}')\boldsymbol{\alpha} + \frac{1}{2\gamma}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1})'\tilde{\mathbf{K}}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1}) \\
& + \frac{1}{2\lambda}(\boldsymbol{\alpha} + \boldsymbol{\zeta} - C_2\mathbf{1}_m)'\mathbf{K}(\boldsymbol{\alpha} + \boldsymbol{\zeta} - C_2\mathbf{1}_m) + \rho\mathbf{v}'\mathbf{Z}(\boldsymbol{\alpha} + \boldsymbol{\zeta} - C_2\mathbf{1}_m) \qquad (7.5) \\
\text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{y} = 0, \quad \mathbf{1}'_{L^+}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1}_{L^+}) = 0, \qquad (7.6) \\
& \bar{\boldsymbol{\alpha}} \leq \mathbf{1}_{m-L^+}, \\
& \mathbf{1}'_m(\boldsymbol{\alpha} + \boldsymbol{\zeta} - C_2\mathbf{1}_m) = 0, \quad \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta} \geq \mathbf{0},
\end{aligned}$$

Let us define $\boldsymbol{\theta} = \frac{1}{C_2}(\boldsymbol{\alpha}+\boldsymbol{\zeta})$, then the constraint $\mathbf{1}'_m(\boldsymbol{\alpha}+\boldsymbol{\zeta}-C_2\mathbf{1}_m) = 0$ becomes $\mathbf{1}'_m\boldsymbol{\theta} = m$, and the constraint $\boldsymbol{\zeta} \geq \mathbf{0}$ becomes $\boldsymbol{\alpha} \leq C_2\boldsymbol{\theta}$. Let us define the feasible set for $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta})$ as

$\mathcal{A} = \{\boldsymbol{\alpha}'\mathbf{y} = 0, \mathbf{1}'_{L^+}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1}_{L^+}) = 0, \bar{\boldsymbol{\alpha}} \leq \mathbf{1}_{m-L^+}, \mathbf{1}'_m\boldsymbol{\theta} = m, \boldsymbol{\alpha} \leq C_2\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta} \geq \mathbf{0}\}$. Substituting $\boldsymbol{\theta} = \frac{1}{C_2}(\boldsymbol{\alpha} + \boldsymbol{\zeta})$ into (7.5), we arrive at,

$$\begin{aligned}\min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})\in\mathcal{A}} \quad & -\mathbf{p}'\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{K}\circ\mathbf{y}\mathbf{y}')\boldsymbol{\alpha} + \frac{1}{2\gamma}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1})'\tilde{\mathbf{K}}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1}) \\ & + \frac{(C_2)^2}{2\lambda}(\boldsymbol{\theta} - \mathbf{1}_m)'\mathbf{K}(\boldsymbol{\theta} - \mathbf{1}_m) + \rho C_2\mathbf{v}'\mathbf{Z}(\boldsymbol{\theta} - \mathbf{1}_m)\end{aligned}$$

Recall in the main text we have defined $H(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\mathbf{p}'\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{K}\circ\mathbf{y}\mathbf{y}')\boldsymbol{\alpha} + \frac{1}{2\gamma}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1})'\tilde{\mathbf{K}}(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}} - C_1\mathbf{1})$, then we simplify the objective function in (7.7) as follows,

$$\min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})\in\mathcal{A}} \quad H(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \frac{(C_2)^2}{2\lambda}(\boldsymbol{\theta} - \mathbf{1}_m)'\mathbf{K}(\boldsymbol{\theta} - \mathbf{1}_m) + \rho C_2\mathbf{v}'\mathbf{Z}(\boldsymbol{\theta} - \mathbf{1}_m) \tag{7.7}$$

Now, we derive the objective function as follows,

$$\begin{aligned}&\min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})\in\mathcal{A}} \quad H(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \frac{(C_2)^2}{2\lambda}(\boldsymbol{\theta} - \mathbf{1}_m)'\mathbf{K}(\boldsymbol{\theta} - \mathbf{1}_m) + \rho C_2\mathbf{v}'\mathbf{Z}(\boldsymbol{\theta} - \mathbf{1}_m) && (7.8)\\ \Leftrightarrow &\min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})\in\mathcal{A}} \quad H(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \frac{(C_2)^2}{2\lambda}(\boldsymbol{\theta}'\mathbf{K}\boldsymbol{\theta} - 2\mathbf{1}'_m\mathbf{K}\boldsymbol{\theta}) + \rho C_2\mathbf{v}'\mathbf{Z}\boldsymbol{\theta} && (7.9)\\ \Leftrightarrow &\min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})\in\mathcal{A}} \quad H(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \frac{(C_2)^2}{2\lambda}\boldsymbol{\theta}'\mathbf{K}\boldsymbol{\theta} - \frac{(C_2)^2}{\lambda}\mathbf{1}'_m\mathbf{K}\boldsymbol{\theta} + \frac{\rho C_2}{m}\mathbf{1}'_m\mathbf{K}\boldsymbol{\theta} - \frac{\rho C_2}{n_t}\mathbf{1}'_{n_t}\mathbf{K}_{ts}\boldsymbol{\theta} && (7.10)\end{aligned}$$

where in (7.9) we omit the constant terms, and in (7.10) we use the equation that $\mathbf{v}'\mathbf{Z} = \frac{1}{m}\mathbf{1}'_m\mathbf{K} - \frac{1}{n_t}\mathbf{1}'_{n_t}\mathbf{K}_{ts}$ with $\mathbf{K}_{ts} \in \mathbb{R}^{n_t\times m}$ being the kernel matrix between the target domain samples and the source domain samples. Let us define $\lambda = \frac{(C_2 m)^2}{\mu}$ and $\rho = \frac{C_2 m}{\lambda} = \frac{\mu}{C_2 m}$ and omit the constant term, then the problem in (7.10) becomes

$$\begin{aligned}&\min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})\in\mathcal{A}} \quad H(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \frac{\mu}{2m^2}\boldsymbol{\theta}'\mathbf{K}\boldsymbol{\theta} - \frac{\mu}{mn_t}\mathbf{1}'_{n_t}\mathbf{K}_{ts}\boldsymbol{\theta} \\ \Leftrightarrow &\min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})\in\mathcal{A}} \quad H(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \frac{\mu}{2m^2}\boldsymbol{\theta}'\mathbf{K}\boldsymbol{\theta} - \frac{\mu}{mn_t}\mathbf{1}'_{n_t}\mathbf{K}_{ts}\boldsymbol{\theta} + \frac{\mu}{2n_t^2}\mathbf{1}'_{n_t}\mathbf{K}_t\mathbf{1}_{n_t} && (7.11)\\ \Leftrightarrow &\min_{(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\theta})\in\mathcal{A}} \quad H(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \frac{\mu}{2}\|\frac{1}{m}\sum_{i=1}^{m}\theta_i\mathbf{z}_i^s - \frac{1}{n_t}\sum_{i=1}^{n_t}\mathbf{z}_i^t\|^2, && (7.12)\end{aligned}$$

where in (7.11) we add a constant $\frac{\mu}{2n_t^2}\mathbf{1}'_{n_t}\mathbf{K}_t\mathbf{1}_{n_t}$ to the objective function with $\mathbf{K}_t \in \mathbb{R}^{n_t\times n_t}$ being the kernel matrix on the target domain samples. Note the problem in (7.12) is exactly the problem in (3.13). We complete the proof here.

Appendix B: Derivations of (4.13) and (4.29)

The derivations of (4.13) and (4.29) are very similar. In fact, the derivation of (4.13) can be used for deriving (4.29) by removing the terms related to privileged information. In the following, we firstly provide the derivation of (4.29), and then discuss how to derive (4.13).

To derive the dual form of (4.24), we firstly reformulate it into a simpler form. In particular, an intermediate variable $\theta_{i,c,m,\tilde{m}}$ is introduced as follows,

$$\theta_{i,c,m,\tilde{m}} = \begin{cases} \hat{\beta}_{i,m} & c = y_i, \\ \delta(m = \tilde{m}) & c \neq y_i. \end{cases} \tag{7.13}$$

Then, we can have $\sum_{m=1}^{M} \hat{\beta}_{i,m}(\mathbf{w}_{y_i,m})'\phi(\mathbf{x}_i) = \sum_{m=1}^{M} \theta_{i,y_i,m,\tilde{m}}(\mathbf{w}_{y_i,m})'\phi(\mathbf{x}_i)$ and $(\mathbf{w}_{c,\tilde{m}})'\phi(\mathbf{x}_i) = \sum_{m=1}^{M} \theta_{i,c,m,\tilde{m}}(\mathbf{w}_{c,m})'\phi(\mathbf{x}_i)$. Similarly, we can represent $\sum_{m=1}^{M} \hat{\beta}_{i,m}(\tilde{\mathbf{w}}_{y_i,m})'\tilde{\phi}(\mathbf{z}_i)$ an $(\tilde{\mathbf{w}}_{c,\tilde{m}})'\tilde{\phi}(\mathbf{z}_i)$ by using θ.

Let us define a function $G(\mathcal{B}_l, \tilde{c}, \tilde{m}) = \frac{1}{|\mathcal{B}_l|} \sum_{i \in \mathcal{I}_l} h_i \left(\sum_{m=1}^{M} \theta_{i,Y_l,m,\tilde{m}}(\mathbf{w}_{Y_l,m})'\phi(\mathbf{x}_i) - \sum_{m=1}^{M} \theta_{i,\tilde{c},m,\tilde{m}}(\mathbf{w}_{\tilde{c},m})'\phi(\mathbf{x}_i) \right)$. By similarly defining $\tilde{G}(\mathcal{B}_l, \tilde{c}, \tilde{m})$ using $\theta_{i,c,m,\tilde{m}}$'s, the constraints in (4.25) and (4.27) can be uniformly written as follows,

$$G(\mathcal{B}_l, \tilde{c}, \tilde{m}) \geq \zeta_{l,\tilde{c},\tilde{m}} - \tilde{G}(\mathcal{B}_l, \tilde{c}, \tilde{m}) - \xi_l, \quad \forall l, \tilde{c}, \tilde{m}, \tag{7.14}$$

in which $\zeta_{l,\tilde{c},\tilde{m}} = 0$ if $\tilde{c} = Y_l$, and $\zeta_{l,\tilde{c},\tilde{m}} = \eta$ otherwise.

Similarly, the constraints in (4.26) and (4.28) can be uniformly written as $\tilde{G}(\mathcal{B}_l, \tilde{c}, \tilde{m}) \geq \zeta_{l,\tilde{c},\tilde{m}} - \epsilon_l, \quad \forall l, \tilde{c}, \tilde{m}$.

All $\mathbf{w}_{c,m}$'s are concatenated and we define $\mathbf{w} = [\mathbf{w}'_{1,1}, \ldots, \mathbf{w}'_{1,M}, \mathbf{w}'_{2,1}, \ldots, \mathbf{w}'_{C,M}]'$. Furthermore, a new mapping function is defined for each $\mathcal{B}_l$ as $\varphi(\mathbf{h}, \mathcal{B}_l, c, \tilde{m}) = [\frac{1}{|B_l|} \sum_{i \in I_l} h_i \ \theta_{i,1,1,\tilde{m}}\delta(c = 1)\phi(\mathbf{x}_i)', \ldots, \frac{1}{|B_l|} \sum_{i \in I_l} h_i \ \theta_{i,C,M,\tilde{m}}\delta(c = C)\phi(\mathbf{x}_i)']'$. Similarly, we concatenate all $\tilde{\mathbf{w}}_{c,m}$'s as $\tilde{\mathbf{w}}$ and define $\tilde{\varphi}(\mathbf{h}, \mathcal{B}_l, c, \tilde{m})$ by replacing $\phi(\mathbf{x}_i)$'s with $\tilde{\phi}(\mathbf{z}_i)$'s. By further denoting $\psi(\mathbf{h}, \mathcal{B}_l, c, \tilde{m}) = \varphi(\mathbf{h}, \mathcal{B}_l, Y_l, \tilde{m}) - \varphi(\mathbf{h}, \mathcal{B}_l, c, \tilde{m})$ and $\tilde{\psi}(\mathbf{h}, \mathcal{B}_l, c, \tilde{m}) = \tilde{\varphi}(\mathbf{h}, \mathcal{B}_l, Y_l, \tilde{m}) - \tilde{\varphi}(\mathbf{h}, \mathcal{B}_l, c, \tilde{m})$, we observe $G(\mathcal{B}_l, \tilde{c}, \tilde{m})$ and $\tilde{G}(\mathcal{B}_l, \tilde{c}, \tilde{m})$ can be represented as $\mathbf{w}'\psi(\mathbf{h}, \mathcal{B}_l, c, m)$ and $\tilde{\mathbf{w}}'\tilde{\psi}(\mathbf{h}, \mathcal{B}_l, c, m)$ respectively, so we can simply the objective func-

tion in (4.24) as follows,

$$\begin{aligned}
\min_{\substack{\mathbf{h}\in\mathcal{H},\mathbf{w},\tilde{\mathbf{w}}\\ \xi_l,\epsilon_l}} \quad & \frac{1}{2}(\|\mathbf{w}\|^2+\lambda\|\tilde{\mathbf{w}}\|^2)+C_1\sum_{l=1}^{L}(\xi_l+\epsilon_l) \\
& -C_2\,\rho(\mathbf{B},\mathbf{K}\circ(\mathbf{h}\mathbf{h}'))+C_3\tilde{\mathbf{w}}'\tilde{\psi}(\mathbf{h},\mathcal{B}_l,c,m) && (7.15)\\
\text{s.t.} \quad & \mathbf{w}'\psi(\mathbf{h},\mathcal{B}_l,c,m)\geq\zeta_{l,c,m}-\tilde{\mathbf{w}}'\tilde{\psi}(\mathbf{h},\mathcal{B}_l,c,m)-\xi_l,\forall l,c,m, && (7.16)\\
& \tilde{\mathbf{w}}'\tilde{\psi}(\mathbf{h},\mathcal{B}_l,c,m)\geq\zeta_{l,c,m}-\epsilon_l,\quad \forall l,c,m. && (7.17)
\end{aligned}$$

We introduce a dual variable $\alpha_{l,c,m}$ and $\varsigma_{l,c,m}$ for each constraint in (7.16) and (7.17) respectively. When the derivatives of the Lagrangian form of (7.15) *w.r.t.* ξ_l's and ϵ_l's are set to zeros respectively, we can obtain $\sum_{c,m}\alpha_{l,c,m}=C_1,\forall l$ and $\sum_{c,m}\varsigma_{l,c,m}=C_1,\forall l$. By respectively setting the derivative of the Lagrangian of (7.15) *w.r.t.* $\mathbf{w}$ and $\tilde{\mathbf{w}}$ as zero, we can obtain the following equations:

$$\mathbf{w}=\sum_{l,c,m}\alpha_{l,c,m}\psi(\mathbf{h},\mathcal{B}_l,c,m), \tag{7.18}$$

$$\tilde{\mathbf{w}}=\frac{1}{\lambda}\sum_{l,c,m}(\alpha_{l,c,m}+\varsigma_{l,c,m}-C_3)\tilde{\psi}(\mathbf{h},\mathcal{B}_l,c,m). \tag{7.19}$$

By substituting (7.18) and (7.19) back into the Lagrangian of (7.15), we can arrive at the dual form of (7.15) as follows,

$$\begin{aligned}
\min_{\mathbf{h}\in\mathcal{H}}\max_{\boldsymbol{\alpha},\boldsymbol{\varsigma}} \quad & -\frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}^{\mathbf{h}}\boldsymbol{\alpha}-\frac{1}{2\lambda}(\boldsymbol{\alpha}+\boldsymbol{\varsigma}-C_3\mathbf{1})'\tilde{\mathbf{Q}}^{\mathbf{h}}(\boldsymbol{\alpha}+\boldsymbol{\varsigma}-C_3\mathbf{1}) \\
& +\boldsymbol{\zeta}'(\boldsymbol{\alpha}+\boldsymbol{\varsigma})-C_2\rho(\mathbf{B},\mathbf{K}\circ(\mathbf{h}\mathbf{h}')) && (7.20)\\
\text{s.t.} \quad & \sum_{c,m}\alpha_{l,c,m}=C_1,\quad \forall l \\
& \alpha_{l,c,m}\geq 0,\quad \forall l,c,m, \\
& \sum_{c,m}\varsigma_{l,c,m}=C_1,\quad \forall l \\
& \varsigma_{l,c,m}\geq 0,\quad \forall l,c,m,
\end{aligned}$$

where $\boldsymbol{\alpha}\in\mathbb{R}^{\tilde{D}}$ (*resp.*, $\boldsymbol{\varsigma}\in\mathbb{R}^{\tilde{D}}$) is a vector containing the dual variables $\alpha_{l,c,m}$ (*resp.*, $\varsigma_{l,c,m}$), $\tilde{D}=L\cdot C\cdot M$, $\boldsymbol{\zeta}\in\mathbb{R}^{\tilde{D}}$ is a vector, in which each entry $\zeta_{l,c,m}=0$ if $c=Y_l$ and $\zeta_{l,c,m}=\eta$ otherwise., $\mathbf{Q}^{\mathbf{h}}\in\mathbb{R}^{\tilde{D}\times\tilde{D}}$ is a matrix with each entry being $Q^{\mathbf{h}}_{u,v}=\psi(\mathbf{h},\mathcal{B}_l,c,m)'\psi(\mathbf{h},\mathcal{B}_{\tilde{l}},\tilde{c},\tilde{m})$, in which u and v are the indices defined in the paragraph below (4.13), and $\tilde{\mathbf{Q}}^{\mathbf{h}}$ is similarly defined as $\mathbf{Q}^{\mathbf{h}}$ by replacing $\psi(\mathbf{h},\mathcal{B}_l,c,m)$ with $\tilde{\psi}(\mathbf{h},\mathcal{B}_l,c,m)$.

In the following, we derive the detailed form of $Q^{\mathbf{h}}_{u,v}$ and $\tilde{Q}^{\mathbf{h}}_{u,v}$. Recall that $\psi(\mathbf{h}, \mathcal{B}_l, c, \tilde{m})$ $= \varphi(\mathbf{h}, \mathcal{B}_l, Y_l, \tilde{m}) - \varphi(\mathbf{h}, \mathcal{B}_l, c, \tilde{m})$, then we can obtain that

$$\begin{aligned} & \psi(\mathbf{h}, \mathcal{B}_l, c, m)' \psi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m}) \\ = \quad & (\varphi(\mathbf{h}, \mathcal{B}_l, Y_l, m) - \varphi(\mathbf{h}, \mathcal{B}_l, c, m))' \left(\varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, Y_{\tilde{l}}, \tilde{m}) - \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m})\right) \\ = \quad & -\varphi(\mathbf{h}, \mathcal{B}_l, Y_l, m)' \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m}) + \varphi(\mathbf{h}, \mathcal{B}_l, Y_l, m)' \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, Y_{\tilde{l}}, \tilde{m}) \\ & + \varphi(\mathbf{h}, \mathcal{B}_l, c, m)' \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m}) - \varphi(\mathbf{h}, \mathcal{B}_l, c, m)' \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, Y_{\tilde{l}}, \tilde{m}) \end{aligned} \tag{7.21}$$

Let us define $S_1 = \varphi(\mathbf{h}, \mathcal{B}_l, Y_l, m)'\ \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m})$, $S_2 = \varphi(\mathbf{h}, \mathcal{B}_l, Y_l, m)'\ \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, Y_{\tilde{l}}, \tilde{m})$, $S_3 = \varphi(\mathbf{h}, \mathcal{B}_l, c, m)'\ \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m})$, $S_4 = \varphi(\mathbf{h}, \mathcal{B}_l, c, m)'\ \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, Y_{\tilde{l}}, \tilde{m})$, then $\psi(\mathbf{h}, \mathcal{B}_l, c, m)'$ $\psi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m}) = -S_1 + S_2 + S_3 - S_4$. We derive the detailed form of S_1, S_2, S_3 and S_4 as follows. Recall that $\varphi(\mathbf{h}, \mathcal{B}_l, c, \tilde{m}) = [\frac{1}{|B_l|} \sum_{i \in I_l} h_i\ \theta_{i,1,1,\tilde{m}} \delta(c = 1)\phi(\mathbf{x}_i)', \ldots, \frac{1}{|B_l|} \sum_{i \in I_l} h_i$ $\theta_{i,C,M,\tilde{m}} \delta(c = C)\phi(\mathbf{x}_i)']'$. The first term S_1 can be derived as,

$$\begin{aligned} S_1 \quad & = \varphi(\mathbf{h}, \mathcal{B}_l, Y_l, m)' \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m}) \\ & = \frac{1}{|\mathcal{B}_l||\mathcal{B}_{\tilde{l}}|} \sum_{i \in \mathbf{I}_l} \sum_{j \in \mathbf{I}_{\tilde{l}}} h_i h_j [\delta(c = Y_l)\delta(c = Y_{\tilde{l}}) \sum_{q=1}^{M} \hat{\beta}_{i,q} \hat{\beta}_{j,q} \\ & \quad \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j) + (1 - \delta(c = Y_l))\delta(c = Y_{\tilde{l}}) \hat{\beta}_{j,m} \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)] \end{aligned} \tag{7.22}$$

The second term S_2 is derived as,

$$\begin{aligned} S_2 \quad & = \varphi(\mathbf{h}, \mathcal{B}_l, Y_l, m)' \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, Y_{\tilde{l}}, \tilde{m}) \\ & = \frac{\delta(Y_l = Y_{\tilde{l}})}{|\mathcal{B}_l||\mathcal{B}_{\tilde{l}}|} \sum_{i \in \mathbf{I}_l} \sum_{j \in \mathbf{I}_{\tilde{l}}} h_i h_j \sum_{q=1}^{M} \hat{\beta}_{i,q} \hat{\beta}_{j,q} \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j). \end{aligned} \tag{7.23}$$

Similarly, we can derive the third term S_3 as follows,

$$\begin{aligned} S_3 \quad & = \varphi(\mathbf{h}, \mathcal{B}_l, c, m)' \varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m}), \\ & = \frac{\delta(c = \tilde{c})}{|\mathcal{B}_l||\mathcal{B}_{\tilde{l}}|} \sum_{i \in \mathbf{I}_l} \sum_{j \in \mathbf{I}_{\tilde{l}}} h_i h_j [\delta(c = Y_l)\delta(\tilde{c} = Y_{\tilde{l}}) \sum_{q=1}^{M} \hat{\beta}_{i,q} \hat{\beta}_{j,q} \\ & \quad \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j) + (1 - \delta(c = Y_l))\delta(\tilde{c} = Y_{\tilde{l}}) \hat{\beta}_{j,m} \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j) \\ & \quad + \delta(c = Y_l)(1 - \delta(\tilde{c} = Y_{\tilde{l}})) \hat{\beta}_{i,\tilde{m}} \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j) \\ & \quad + (1 - \delta(c = Y_l))(1 - \delta(\tilde{c} = Y_{\tilde{l}}))\delta(m = \tilde{m}) \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)]. \end{aligned} \tag{7.24}$$

Finally, the last term S_4 is derived as,

$$\begin{aligned} S_4 &= \varphi(\mathbf{h}, \mathcal{B}_l, c, m)'\varphi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, Y_{\tilde{l}}, \tilde{m}) \\ &= \frac{1}{|\mathcal{B}_l||\mathcal{B}_{\tilde{l}}|} \sum_{i \in \mathbf{I}_l} \sum_{j \in \mathbf{I}_{\tilde{l}}} h_i h_j [\delta(\tilde{c} = Y_{\tilde{l}}) \delta(\tilde{c} = Y_l) \sum_{q=1}^{M} \hat{\beta}_{j,q} \hat{\beta}_{i,q} \\ &\quad \phi(\mathbf{x}_i)'\phi(\mathbf{x}_j) + (1 - \delta(\tilde{c} = Y_{\tilde{l}}))\delta(\tilde{c} = Y_l)\hat{\beta}_{i,\tilde{m}}\phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)]. \end{aligned} \tag{7.25}$$

By substituting (7.22), (7.23), (7.24), and (7.25) into (7.21), and combining similar terms, we arrive at

$$\begin{aligned} &\psi(\mathbf{h}, \mathcal{B}_l, c, m)'\psi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m}) \\ &= -S_1 + S_2 + S_3 - S_4 \\ &= \frac{1}{|\mathcal{B}_l||\mathcal{B}_{\tilde{l}}|} \sum_{i \in \mathbf{I}_l} \sum_{j \in \mathbf{I}_{\tilde{l}}} h_i h_j [\delta(Y_l = Y_{\tilde{l}})(1 - \delta(c = Y_l)) \\ &\quad (1 - \delta(\tilde{c} = Y_{\tilde{l}})) \sum_{q=1}^{M} \hat{\beta}_{j,q}\hat{\beta}_{i,q}\phi(\mathbf{x}_i)'\phi(\mathbf{x}_j) \\ &\quad -(1 - \delta(c = Y_l))(1 - \delta(c = \tilde{c}))\delta(c = Y_{\tilde{l}}) \\ &\quad \hat{\beta}_{j,m}\phi(\mathbf{x}_i)'\phi(\mathbf{x}_j) - (1 - \delta(\tilde{c} = Y_{\tilde{l}}))(1 - \delta(c = \tilde{c})) \\ &\quad \delta(\tilde{c} = Y_l)\hat{\beta}_{i,\tilde{m}}\phi(\mathbf{x}_i)'\phi(\mathbf{x}_j) + \delta(m = \tilde{m})\delta(c = \tilde{c}) \\ &\quad (1 - \delta(\tilde{c} = Y_{\tilde{l}}))(1 - \delta(c = Y_l))\phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)] \\ &= \frac{1}{|\mathcal{B}_l||\mathcal{B}_{\tilde{l}}|} \sum_{i \in \mathbf{I}_l} \sum_{j \in \mathbf{I}_{\tilde{l}}} h_i h_j \phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)\gamma(i, j, c, \tilde{c}, m, \tilde{m}), \end{aligned}$$

where $\gamma(i, j, c, \tilde{c}, m, \tilde{m})$ is defined in the paragraph below (4.13).

Recall that $Q^{\mathbf{h}}_{u,v} = \psi(\mathbf{h}, \mathcal{B}_l, c, m)'\psi(\mathbf{h}, \mathcal{B}_{\tilde{l}}, \tilde{c}, \tilde{m})$, where u and v are the indices defined in the paragraph below (4.13), so $Q^{\mathbf{h}}_{u,v} = \frac{1}{|\mathcal{B}_l||\mathcal{B}_{\tilde{l}}|} \sum_{i \in \mathbf{I}_l} \sum_{j \in \mathbf{I}_{\tilde{l}}} h_i h_j \, \phi(\mathbf{x}_i)'\phi(\mathbf{x}_j) \, \gamma(i, j, c, \tilde{c}, m, \tilde{m})$. Note that the detailed form of each entry in $\tilde{\mathbf{Q}}^{\mathbf{h}}$, *i.e.*, $\tilde{Q}^{\mathbf{h}}_{u,v}$, can be similarly derived by replacing $\phi(\mathbf{x}_i)$ with $\tilde{\phi}(\mathbf{z}_i)$. Given the detailed form of each entry in $\mathbf{Q}^{\mathbf{h}}$ and $\tilde{\mathbf{Q}}^{\mathbf{h}}$, the optimization problem in (7.20) is equivalent to (4.29), so we complete the derivation of (4.29) here.

To derive (4.13), by concatenating all $\mathbf{w}_{c,m}$'s as $\mathbf{w}$ and using the same definition of

$\varphi(\mathbf{h}, \mathcal{B}_l, c, \tilde{m})$, we can simplify the problem in (4.9) as follows,

$$\begin{aligned} \min_{\substack{\mathbf{h}\in\mathcal{H} \\ \mathbf{w},\xi_l}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{l=1}^{L} \xi_l - C_2\, \rho(\mathbf{B}, \mathbf{K} \circ (\mathbf{h}\mathbf{h}')) \\ \text{s.t.} \quad & \mathbf{w}'\psi(\mathbf{h}, \mathcal{B}_l, c, m) \geq \zeta_{l,c,m} - \xi_l, \quad \forall l, c, m. \end{aligned} \tag{7.26}$$

After introducing dual variables $\alpha_{l,c,m}$'s for the constraints in (7.26), we can similarly obtain the dual form of (7.26) as (4.13).

Appendix C: Derivation of (2)

We rewrite (5.32) as follows,

$$\begin{aligned} \min_{\mathbf{d}} \max_{\boldsymbol{\alpha}_i} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V} d_v \boldsymbol{\alpha}_i' \mathbf{M}_i^v \boldsymbol{\alpha}_i + \sum_{i=1}^{n} 1'\boldsymbol{\alpha}_i - \frac{\zeta_2}{2}\sum_{i=1}^{n}\|\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\|^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}, \end{aligned} \tag{7.27}$$

in which β_i is the i-th column of $\mathbf{B}$.

Now we prove that the primal form of (5.32) can be written as follows,

$$\min_{\substack{\mathbf{d},\mathbf{w}_i^v \\ \tilde{\xi}_i,\tilde{\epsilon}_{ij}}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V}\frac{\|\mathbf{w}_i^v\|^2}{d_v} + \frac{1}{2\zeta_2}\Big(\sum_{i=1}^{n}\tilde{\xi}_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{\epsilon}_{ij}^2\Big) \tag{7.28}$$

$$\text{s.t.} \quad \sum_{v=1}^{V}\mathbf{w}_i^{v'}\mathbf{x}_i^{v+} \geq (1 + \zeta_2\beta_i^+) - \tilde{\xi}_i, \quad \forall i, \tag{7.29}$$

$$\sum_{v=1}^{V}\mathbf{w}_i^{v'}\mathbf{x}_j^{v-} \leq -(1 + \zeta_2\beta_{ij}^-) + \tilde{\epsilon}_{ij}, \quad \forall i, \forall j, \tag{7.30}$$

$$\mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0}, \tag{7.31}$$

where β_i^+'s and β_{ij}^-'s are newly introduced variables, and $\tilde{\xi}_i$'s and $\tilde{\epsilon}_{ij}$'s are the slack variables.

After introducing the dual variables α_i^+'s for the constraints in (5.34) and α_{ij}^-'s for

the constraints in (5.35), we arrive at the Lagrangian form of (5.33) as,

$$\begin{aligned}\mathcal{L}_{\mathbf{w}} = \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V}\frac{\|\mathbf{w}_i^v\|^2}{d_v} + \frac{1}{2\zeta_2}(\sum_{i=1}^{n}\tilde{\xi}_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{\epsilon}_{ij}^2) \\ & - \sum_{i=1}^{n}\alpha_i^+(\sum_{v=1}^{V}\mathbf{w}_i^{v'}\mathbf{x}_i^{v+} - 1 - \zeta_2\beta_i^+ + \tilde{\xi}_i) \\ & + \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}^-(\sum_{v=1}^{V}\mathbf{w}_i^{v'}\mathbf{x}_j^{v-} + 1 + \zeta_2\beta_{ij}^- - \tilde{\epsilon}_{ij})\end{aligned} \tag{7.32}$$

By setting the derivatives of $\mathcal{L}_{\mathbf{w}}$ *w.r.t.* $\tilde{\xi}_i$, $\tilde{\epsilon}_{ij}$, and $\mathbf{w}_i^v$ to zeros separately, we obtain $\tilde{\xi}_i = \zeta_2\alpha_i^+$, $\tilde{\epsilon}_{ij} = \zeta_2\alpha_{ij}^-$, and the following equation:

$$\mathbf{w}_i^v = d_v\mathbf{X}_i^v(\boldsymbol{\alpha}_i \circ \mathbf{y}), \tag{7.33}$$

in which $\mathbf{X}_i^v$ and $\mathbf{y}$ are the same as defined in the paragraph below (5.19), and $\boldsymbol{\alpha}_i = [\alpha_i^+, \alpha_{i1}^-, \ldots, \alpha_{im}^-]'$ corresponds to the dual vector in (5.32). By substituting (7.33) back into (7.32), we can obtain the dual form of (5.33) as,

$$\begin{aligned}\min_{\mathbf{d}}\max_{\boldsymbol{\alpha}_i} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{v=1}^{V}d_v\boldsymbol{\alpha}_i'\mathbf{M}_i^v\boldsymbol{\alpha}_i + \sum_{i=1}^{n}(\mathbf{1} + \zeta_2\boldsymbol{\beta}_i)'\boldsymbol{\alpha}_i - \frac{\zeta_2}{2}\sum_{i=1}^{n}\|\boldsymbol{\alpha}_i\|^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \mathbf{0}, \quad \forall i, \\ & \mathbf{1}'\mathbf{d} = 1, \quad \mathbf{d} \geq \mathbf{0},\end{aligned} \tag{7.34}$$

where $\boldsymbol{\beta}_i = [\beta_i^+, \beta_{i1}^-, \ldots, \beta_{im}^-]'$ corresponds to $\boldsymbol{\beta}_i$ in (5.32). After adding a constant term $-\frac{\zeta_2}{2}\sum_{i=1}^{n}\|\boldsymbol{\beta}_i\|^2$ in (7.34) followed by some simplifications, we can arrive at the exact form of (5.32). So we complete the proof here.

Publication

Journal Publications

- **Li Niu**, Wen Li, Dong Xu, and Jianfei Cai, "An Exemplar-based Multi-view Domain Generalization Framework for Visual Recognition," *IEEE Transactions on Neural Networks and Learning Systems* (**T-NNLS**), vol. PP, no. 99, pp. 1-14, November 2016.

- **Li Niu**, Wen Li, Dong Xu, and Jianfei Cai, "Visual Recognition by Learning from Web Data via Weakly Supervised Domain Generalization," *IEEE Transactions on Neural Networks and Learning Systems* (**T-NNLS**), vol. PP, no. 99, pp. 1-15, June 2016.

- **Li Niu**, Xinxing Xu, Lin Chen, Lixin Duan, and Dong Xu, "Action and Event Recognition in Videos by Learning from Heterogeneous Web Sources," *IEEE Transactions on Neural Networks and Learning Systems* (**T-NNLS**), vol. PP, no. 99, pp. 1-15, March 2016.

- **Li Niu**, Wen Li, and Dong Xu, "Exploiting Privileged Information from Web Data for Action and Event Recognition," *International Journal of Computer Vision* (**IJCV**), vol. 118, no. 2, pp. 130-150, June 2016.

Conference Publications

- **Li Niu**, Jianfei Cai, and Dong Xu, "Domain Adaptive Fisher Vector for Visual Recognition," *in Proceedings of European Conference on Computer Vision* (**ECCV**), Amsterdam, Netherlands, October 2016, pp. 550-566.

- **Li Niu**, Wen Li, and Dong Xu, "Multi-view Domain Generalization for Visual Recognition," *in Proceedings of IEEE International Conference on Computer Vision* (**ICCV**), Santiago, Chile, November 2015, pp. 4193-4201.

- **Li Niu**, Wen Li, and Dong Xu, "Visual Recognition by Learning from Web Data: A Weakly Supervised Domain Generalization Approach," *in Proceedings of IEEE Interna-tional Conference on Computer Vision and Pattern Recognition* (**CVPR**), Boston, U.S., June 2015, pp. 2774-2783.

- **Li Niu** *, Wen Li *, and Dong Xu, "Exploiting Privileged Information from Web Data for Image Categorization," *in Proceedings of European Conference on Computer Vision* (**ECCV**), Zurich, Switzerland, September 2014, pp. 437-452. * indicates equal contribution

- Zheng Xu, Wen Li, **Li Niu**, and Dong Xu, "Exploiting Low-rank Structure from Latent Domains for Domain Generalization," *in Proceedings of European Conference on Computer Vision* (**ECCV**), Zurich, Switzerland, September 2014, pp. 628-643.