ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200267

Adversarial Domain Adaptation Being Aware of Class Relationships

Zeya Wang [1 2] and **Baoyu Jing** [3] and **Yang Ni** [4] and **Nanqing Dong** [5] and **Pengtao Xie** [1] and **Eric Xing** [1]

Abstract. Adversarial training is a useful approach to promote the learning of transferable representations across the source and target domains, which has been widely applied for domain adaptation (DA) tasks based on deep neural networks. Until very recently, existing adversarial domain adaptation (ADA) methods ignore the useful information from the label space, which is an important factor accountable for the complicated data distributions associated with different semantic classes. Especially, the inter-class semantic relationships have been rarely considered and discussed in the current work of transfer learning. In this paper, we propose a novel relationship-aware adversarial domain adaptation (RADA) algorithm, which first utilizes a single multi-class domain discriminator to enforce the learning of inter-class dependency structure during domain-adversarial training and then aligns this structure with the inter-class dependencies that are characterized from training the label predictor on source domain. Specifically, we impose a regularization term to penalize the structure discrepancy between the inter-class dependencies respectively estimated from domain discriminator and label predictor. Through this alignment, our proposed method makes the adversarial domain adaptation aware of the class relationships. Empirical studies show that the incorporation of class relationships significantly improves the performance on benchmark datasets.

1 INTRODUCTION

The success of deep learning largely depends on large-scale datasets with labels (e.g. ImageNet [9]). Manually annotating labels is costly and time-consuming, which becomes an obstacle for applying deep learning models to new datasets [12]. An effective approach to build a model on unlabeled data in a target domain is to leverage off-the-shelf labeled data from its relevant source domains. However, due to domain shift [32], models trained on source domains usually do not generalize well to target domains.

Recently, adversarial training has been introduced to learn domain-invariant features and substantially improves the domain adaptation performance [12]. These adversarial-learning-based methods incorporate a domain discriminator to encourage domain confusion for minimizing the distribution discrepancy between source and target domains [12, 2, 26, 33, 28]. Despite the significant improvement from adversarial domain adaptation, most existing methods simply match the distributions across domains without considering the structure behind the complicated data distributions. The conditional distributions of data given different associated semantic classes can be different, which may lead to multimodal distributions for multi-class classification. Failing to capture the modes of data distribution will mislead the alignment of distributions across domains [28, 24, 1]. Current attempts focus on revealing this complex structure within the feature space, but ignore the high-level semantics from the label space. Some method has been proposed to disentangle semantic latent variables from domain latent variables in the latent manifold, but it relies on a variational auto-encoder to reconstruct the latent variables and does not explicitly account for the high-level semantics from the label space [3]. Some recent studies design separate class-wise domain discriminators, where each discriminator is only responsible for the distribution alignment for one semantic class [28, 7]. Including the class information, these approaches successfully mitigates the false distribution alignment across domains. However, assigning separate discriminator for each class essentially constrains all classes to be orthogonal with each other. These methods do not explore the inter-class semantic relationships in the label space for DA.

Utilizing structure information from the label space could be helpful for capturing the multimodal structure more accurately. Intuitively, the class relationships are supposed to remain consistent across domains (Figure 1), which motivates us to exploit the structure information among semantic classes and inject it into the learning process of DA. Multiple task learning (MTL) jointly learns multiple related tasks through knowledge sharing, where structure learning has gained growing popularity for explicitly exploiting hidden task structures. Gaussian graphical model is a powerful tool for studying conditional dependency structure among random variables, so has been widely used for learning the structure of task relationships [13]. Recently, this approach has been extended to exploit the class relationships with deep neural networks (DNNs) for improved video categorization performance [20], which provides an effective solution for characterizing inter-class relationships in our work.

Inspired by this line of work, we first design a single multi-class domain discriminator that implements class-specific domain classification. In doing so, we encourage knowledge sharing across classes for domain classification, which enables the learning of inter-class dependencies, and also favor a parsimonious network. We introduce a structure regularization to constrain the class relationships captured by the domain discrimination maximally agree to the inter-class dependencies that are revealed from label prediction on source domain data. Given that this work focuses on how class relationships are incorporated to improve domain adaptation, we build our model on top of domain adversarial neural network (DANN) [12], which is the

[1] Petuum, Inc., United States, email: zw17.rice@gmail.com, {pengtao.xie, eric.xing}@petuum.com
[2] Rice University, United States
[3] University of Illinois at Urbana-Champaign, United States, email: baoyuj2@illinois.edu
[4] Texas A&M University, United States, email: yni@stat.tamu.edu
[5] University of Oxford, United Kingdom, email: nanqing.dong@cs.ox.ac.uk

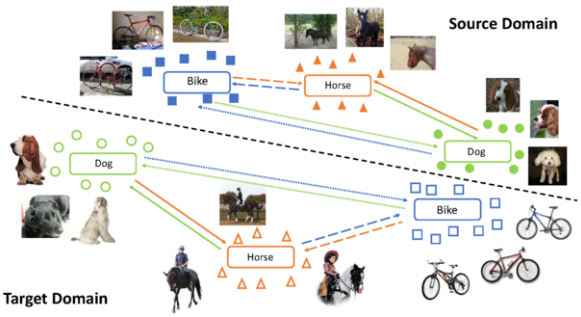


Figure 1. Class relationships are intuitively supposed to be similar between source and target domains (different line types imply different degree of class relationships). We are motivated to encourage the similarity for making adversarial domain adaptation aware of class relationships.

plain adversarial domain adaptation framework. We point out that the presented design and regularizer can be seen as an "add-on" and be easily integrated to other adversarial domain adaptation frameworks. Experiments on benchmark datasets show the proposed approach outperforms the competing methods.

2 RELATED WORK

Adversarial domain adaptation Deep domain adaptation methods attempt to generalize the deep neural networks across different domains. The most commonly used approaches are based on discrepancy minimization [34, 23, 30, 26, 25, 6, 21] or adversarial training [11, 12, 33, 36, 10]. Adversarial training, inspired by generative modeling in GANs [16], is an important approach for deep transfer learning tasks. DANN [13] is proposed with a domain discriminator for classifying whether a sample is from the source or target domains [12, 11]. With a gradient reversal layer (GRL), it promotes the learning of discriminative features for classification, and ensures the learned feature distributions over different domains are similar. Recent works realize the importance of exploiting the complex structure behind the data distributions for domain adaptation rather than just aligning the whole source and target distributions [24, 28]. Multi-adversarial domain adaptation(MADA) utilizes the information from the label space by assigning class-wise discriminators to capture the multimodal structure owing to different classes [28]. However, the structure information from label space is unexplored for DA.

Structure learning Multi-task learning (MTL) seeks to improve the generalization performance by transferring knowledge among related tasks. This knowledge sharing feature makes it possible for learning the structure among tasks, so structure learning, which studies how to accurately characterize the task relationships, has become a central issue of MTL [13, 37].As one of the earliest MTL models, DNNs also share certain commonalities (neurons of the hidden layer) among the neurons of the output layers [5, 20]. Inspired by the methods explicitly modeling task relationships in MTL [14, 37], recent studies for multi-class classification using CNNs exploit and harness the inter-class relationships through imposing a regularization, which has been successfully validated for improving the video categorization performance [20].

3 METHODS

In this section, we first discuss how class relationships are modeled with DNNs, followed by the design of single discriminator to perform class-specific domain classification. We then introduce our RADA algorithm that is able to keep the domain adversarial training aware of class relationships.

3.1 Inter-class dependency structure learning with deep neural networks

For the multi-class classification problem, the data $\mathcal{D} = \{\mathbf{x_m}, \mathbf{y_m}\}_{m=1}^{M}$ is given, where $\mathbf{x_m}$ represents the input features and $\mathbf{y_m} \in \mathbb{R}^K$ is the associated label for each sample. A DNN $f : \mathbf{x} \mapsto \mathbf{y}$ is used to map the input features of each sample to its associated class $k = 1, \cdots, K$ through a large number of interconnected neurons. Typically, these neurons are arranged in multiple layers, e.g., convolutional and pooling layers. In the classification task, a stack of fully connected (FC) layers are often on top of these layers for predicting the final class scores. Only considering the FC layers in a network with L layers in total, we use $\mathbf{W}^{[l]} \in \mathbb{R}^{N_{l-1} \times N_l}$ and $\mathbf{b}^{[l]} \in \mathbb{R}^{N_l}$ to denote the weight matrix and bias vector of neurons in the l-th layer respectively, where N_l denotes the number of neurons in that layer. Let $\mathbf{a}^{[l-1]}$ and $\mathbf{a}^{[l]}$ denote the input and output of the l-th layer with an activation function $g(\cdot)$. We have $\mathbf{a}^{[l]} = g({\mathbf{W}^{[l]}}^T \mathbf{a}^{[l-1]} + \mathbf{b}^{[l]})$, and the final output of the network is $\hat{\mathbf{y}} = f(\mathbf{x}) = \mathbf{a}^{[L]}$. For simplicity of the following discussion, we concatenate $\mathbf{b}^{[l]}$ to the row vectors of $\mathbf{W}^{[l]}$ to have a unified weight matrix $\mathbf{W}^{[l]} \in \mathbb{R}^{(N_{l-1}+1) \times N_l}$. The training objective can be calculated through a cross entropy loss L_y:

$$\min \sum_m L_y(f(\mathbf{x_m}), \mathbf{y_m}). \tag{1}$$

Inspired by recent research for learning task relationships in MTL [37, 20, 14], in classification problems, DNN has been used to exploit the inter-class dependency structure through additional regularization on the output layer to enforce knowledge sharing across different classes. One typical way to model the dependency structure among K classes is through a precision matrix $\mathbf{\Omega} \in \mathbb{R}^{K \times K}$, of which each off-diagonal element captures the pairwise partial correlation between classes. Specifically, we assume the row vectors of weight matrix $\mathbf{W}^{[L]}$ of the output layer follow a multivariate Gaussian distribution $\mathbf{W}_i^{[L]} \sim \mathcal{N}(0, \mathbf{\Omega}^{-1}), i = 1, \cdots, N_{L-1} + 1$. Let $d = N_{L-1} + 1$. By maximizing the log-likelihood of $\mathbf{\Omega}$ following this assumption of Gaussian distribution subject to the positive semidefinite constraint (denoted as $\mathbf{\Omega} \succeq 0$), $\mathbf{\Omega}$ can be optimized concurrently with the training objective in equation (1) by:

$$\begin{aligned} &\min_{\mathbf{\Omega}} -d \log\det(\mathbf{\Omega}) + \mathrm{Tr}(\mathbf{W}^{[L]} \mathbf{\Omega} {\mathbf{W}^{[L]}}^\top), \\ &s.t. \quad \mathbf{\Omega} \succeq 0. \end{aligned} \tag{2}$$

3.2 Multi-class adversarial domain adaptation

In an *unsupervised domain adaptation*(UDA) problem, we are given labeled source domain data $\mathcal{D}_s = \{\mathbf{x}_m^s, \mathbf{y}_m^s\}_{m=1}^{M_s}$ and unlabeled target domain data $\mathcal{D}_t = \{\mathbf{x}_m^t\}_{m=1}^{M_t}$. DANN has been designed to extract domain invariant features between source and target domains through an adversarial training scheme [12]. The whole architecture consists of three parts: a feature extractor G_f, a label predictor G_y,

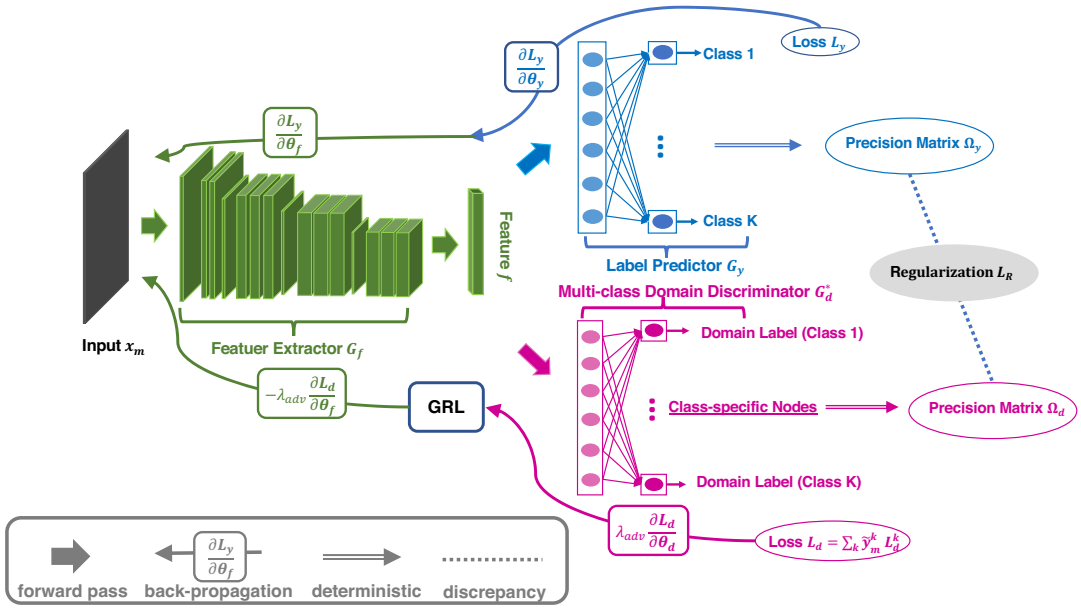


Figure 2. The architecture of the proposed RADA algorithm built on top of the plain DANN model. In our paper we use a one-layer domain discriminator with $(x \to 512 \to K)$. Note that double arrows represent deterministic inference and dashed lines denote the structure discrepancy.

and a domain discriminator G_d. G_f and G_y together form a standard feed-forward DNN $f(\cdot)$ for predicting class labels. G_d is trained to discriminate samples between source and target domains, while G_f is fine-tuned to confuse G_d. Let θ_f, θ_y, and θ_d denote the parameters of G_f, G_y, and G_d, respectively. In the adversarial training procedure, θ_d is learned by minimizing a binary cross entropy loss L_d over the domain labels $\mathbf{d}_m$, while θ_f is learned by maximizing L_d jointly with minimizing L_y (equation (3)). This is achieved by integrating a gradient reversal layer (GRL) between G_f and G_d, finally ensuring the feature distributions over the source and target domains are made similar.

$$\begin{aligned} &\min \frac{1}{M_s} \sum_{\mathbf{x_m} \in \mathcal{D}_s} L_y(G_y(G_f(\mathbf{x_m})), \mathbf{y_m}) \\ &+ \frac{\lambda_{adv}}{M_s + M_t} \sum_{\mathbf{x_m} \in \mathcal{D}_s \cup \mathcal{D}_t} L_d(G_d(\mathcal{R}(G_f(\mathbf{x_m}))), \mathbf{d_m}) \end{aligned} \tag{3}$$

where $\mathcal{R}(\cdot)$ is a pseudo-function for GRL [12], and λ_{adv} is a balancing parameter for adversarial loss.

In order to capture the multimodal structure of data distribution that is accountable by different semantic classes for domain adaptation, a design of multiple discriminators has been applied, such that one discriminator is responsible for matching the source and target domain data associated with one certain class [28]. This design has been proved to successfully enhance positive transfer and alleviate negative transfer, a phenomenon where the source domain data play a part in the reduced learning performance in the target domain [22, 18, 35]. However, there are still two concerns: 1) it has a strong assumption of orthogonality across classes during distribution alignment, i.e., it neglects the structure information among the semantic classes 2) the number of discriminators increased with the number of classes elevates the memory cost for network parameters. In addressing these concerns, we first present a multi-class adversarial domain adaptation G_d^*, where it should be noticed that the way we use "multi-class" differs from that in standard multi-class classification. Instead of adopting separate discriminators, we use one single discriminator with a multi-branch design to match the multimodal structure across different classes.

Figure 2 gives a demonstration of G_d^* in the whole network. One shared hidden layer in G_d^* encodes the common discriminative features that can be used to classify domains for all classes. This shared layer is followed by a layer with class-specific nodes, where each node/branch is only responsible for predicting the domain label of the samples from its associated class, and the node will be muted when the samples to be estimated with domain labels are associated with other classes. For one sample, the ground truth of domain label for each class-specific node is consistent with the ground truth of domain label for that sample. We use L_d^k $(k = 1, \cdots, K)$ to denote the binary domain classification loss associated with class k. With label information $\mathbf{y_m}$, source domain data can be easily assigned to each class-specific node. For the unlabeled target domain data, a weighted sum of loss values from different nodes are calculated, where the probability score vector $\hat{\mathbf{y}}_\mathbf{m}$ given by G_y are used as the weights. Integrating this new design, we update the objective of our multi-class adversarial domain adaptation as:

$$\begin{aligned} &\min \frac{1}{M_s} \sum_{\mathbf{x_m} \in \mathcal{D}_s} L_y(G_y(G_f(\mathbf{x_m})), \mathbf{y_m}) \\ &+ \frac{\lambda_{adv}}{M_s + M_t} \sum_{k=1}^{K} \sum_{\mathbf{x_m} \in \mathcal{D}_s \cup \mathcal{D}_t} \tilde{y}_m^k L_d^k(G_d^*(\mathcal{R}(G_f(\mathbf{x_m}))), \mathbf{d_m}) \end{aligned} \tag{4}$$

where $\tilde{\mathbf{y}}_\mathbf{m} = \{\tilde{y}_m^k\}_{k=1}^K$ is one-hot encoding of $\mathbf{y_m}$ for $\mathbf{x_m} \in \mathcal{D}_s$ and $\tilde{\mathbf{y}}_\mathbf{m} = \hat{\mathbf{y}}_\mathbf{m}$ for $\mathbf{x_m} \in \mathcal{D}_t$.

3.3 Adversarial domain adaptation being aware of class relationships

Incorporating the information of class relationships to the alignment process between the source and target data distributions will relax

the orthogonality assumption and help maximally match the multi-modal structure of data distributions. Recall that $\mathbf{\Omega}$ is used to model the inter-class dependency structure with DNN from Section 3.1. By implicitly injecting $\mathbf{\Omega}$ into the adversarial training process, we make adversarial domain adaptation automatically aware of class relationships.

With the extracted features from G_f, G_y predicts class labels, where the class relationships can be characterized from the prediction at the same time. During the process of domain adaptation, G_d^* aligns the source and target data distributions, which should have similar class structure information. In order for G_d^* to capture a similar inter-class dependency structure during the alignment, the precision matrix $\mathbf{\Omega_d}$, which can be estimated from the class-specific domain classification job implemented by G_d^*, is supposed to be consistent with $\mathbf{\Omega_y}$ from the prediction task done by G_y. To maximize this consistency, we propose an approach that minimizes the discrepancy between the class relationships respectively learned from G_y and G_d^* (as shown in Figure 2). Let $\mathbf{\Omega_y}$ and $\mathbf{\Omega_d}$ denote the precision matrices w.r.t the weight matrices ${\mathbf{W_y}}^{[L]}$ and ${\mathbf{W_d}}^{[L]}$ of the output layers in G_y and G_d^*, thus $\mathbf{\Omega_y}$ and $\mathbf{\Omega_d}$ are used to characterize the inter-class dependencies w.r.t G_y and G_d. Let d_y be the value of d calculated w.r.t G_y, according to equation (2), we can solve $\mathbf{\Omega_y}$ from

$$\begin{aligned} &\min_{\mathbf{\Omega_y}} -d_y \log\det(\mathbf{\Omega_y}) + \mathrm{Tr}({\mathbf{W_y}}^{[L]}\mathbf{\Omega_y}{\mathbf{W_y}}^{[L]^T}); \\ &s.t. \quad \mathbf{\Omega_y} \succeq 0. \end{aligned} \tag{5}$$

It is straightforward to derive the solution to this minimization problem, assuming ${\mathbf{W_y}}^{[L]^T}{\mathbf{W_y}}^{[L]}$ is positive definite. Using the spectral theorem, we can conclude $\mathbf{\Omega_y} = d_y({\mathbf{W_y}}^{[L]^T}{\mathbf{W_y}}^{[L]})^{-1}$ by assuming the positive definiteness of ${\mathbf{W_y}}^{[\mathbf{L}]^T}{\mathbf{W_y}}^{[L]}$(the derivation is included in the appendix). In practice, this positive definiteness is usually guaranteed given the much larger number of hidden nodes compared with the number of classes. In a rare case when it cannot be satisfied, a shrinkage approach can be applied, e.g., adding a small multiple of the identity matrix. Similarly, assuming the positive definiteness of ${\mathbf{W_d}}^{[\mathbf{L}]^T}{\mathbf{W_d}}^{[L]}$ solving

$$\begin{aligned} &\min_{\mathbf{\Omega_d}} -d_d \log\det(\mathbf{\Omega_d}) + \mathrm{Tr}({\mathbf{W_d}}^{[L]}\mathbf{\Omega_d}{\mathbf{W_d}}^{[L]^T}); \\ &s.t. \quad \mathbf{\Omega_d} \succeq 0. \end{aligned} \tag{6}$$

With d_d being the value of d from G_d^*, we obtain $\mathbf{\Omega_d} = d_d({\mathbf{W_d}}^{[L]^T}{\mathbf{W_d}}^{[L]})^{-1}$. We adopt a structure regularization approach that minimizes the discrepancy between precision matrices $\mathbf{\Omega_y}$ and $\mathbf{\Omega_d}$ (a.k.a. KL divergence [8]). It is an asymmetric metric, thus we can formulate the discrepancy in two ways, as:

$$D_{KL}(\mathbf{\Omega_y}||\mathbf{\Omega_d}) = \mathrm{Tr}({\mathbf{\Omega_y}}^{-1}\mathbf{\Omega_d}) - \log\det({\mathbf{\Omega_y}}^{-1}\mathbf{\Omega_d}) - K \tag{7}$$

, which minimizes the divergence from $\mathbf{\Omega_d}$ to $\mathbf{\Omega_y}$, or

$$D_{KL}(\mathbf{\Omega_d}||\mathbf{\Omega_y}) = \mathrm{Tr}({\mathbf{\Omega_d}}^{-1}\mathbf{\Omega_y}) - \log\det({\mathbf{\Omega_d}}^{-1}\mathbf{\Omega_y}) - K. \tag{8}$$

, which minimizes the divergence from $\mathbf{\Omega_y}$ to $\mathbf{\Omega_d}$. Inserting $\mathbf{\Omega_y} = d_y({\mathbf{W_y}}^{[L]^T}{\mathbf{W_y}}^{[L]})^{-1}$ and $\mathbf{\Omega_d} = d_d({\mathbf{W_d}}^{[L]^T}{\mathbf{W_d}}^{[L]})^{-1}$ into equation (7) or (8), we design a regularization to minimize the discrepancy of class relationships between G_y and G_d^* in two ways:

$$\begin{aligned} &\mathbf{d} \to \mathbf{y} : L_R = \mathrm{Tr}({\mathbf{W_y}}^{[L]}({\mathbf{W_d}}^{[L]^T}{\mathbf{W_d}}^{[L]})^{-1}{\mathbf{W_y}}^{[L]^T}) \\ &-\frac{d_y}{d_d}\{\log\det({\mathbf{W_y}}^{[L]^T}{\mathbf{W_y}}^{[L]}) - \log\det({\mathbf{W_d}}^{[L]^T}{\mathbf{W_d}}^{[L]})\} \end{aligned} \tag{9}$$

$$\begin{aligned} &\mathbf{y} \to \mathbf{d} : L_R = \mathrm{Tr}({\mathbf{W_d}}^{[L]}({\mathbf{W_y}}^{[L]^T}{\mathbf{W_y}}^{[L]})^{-1}{\mathbf{W_d}}^{[L]^T}) \\ &-\frac{d_d}{d_y}\{\log\det({\mathbf{W_d}}^{[L]^T}{\mathbf{W_d}}^{[L]}) - \log\det({\mathbf{W_y}}^{[L]^T}{\mathbf{W_y}}^{[L]})\} \end{aligned} \tag{10}$$

Given the asymmetry of the KL divergence between matrices, we will implement both of $D_{KL}(\mathbf{\Omega_d}||\mathbf{\Omega_y})$ and $D_{KL}(\mathbf{\Omega_y}||\mathbf{\Omega_d})$ in our experiments and investigate their difference in our following discussion. Integrating the penalty L_R to equation (4), we have our final training objective:

$$\begin{aligned} \min &\frac{1}{M_s}\sum_{\mathbf{x_m}\in\mathcal{D}_s} L_y(G_y(G_f(\mathbf{x_m})),\mathbf{y_m}) + \lambda_R L_R \\ &+\frac{\lambda_{adv}}{M_s+M_t}\sum_{k=1}^{K}\sum_{\mathbf{x_m}\in\mathcal{D}_s\cup\mathcal{D}_t} \tilde{y}_m^k L_D^k(G_d^*(\mathcal{R}(G_f(\mathbf{x_m}))),\mathbf{d_m}) \end{aligned} \tag{11}$$

where λ_R is a balancing parameter for the relationship-aware regularization term.

4 EXPERIMENTS

4.1 Experiment setup

Datasets We evaluate our model performance on two benchmarks. The first dataset is *ImageCLEF-DA* [6]. All the images are collected from three public datasets: *Caltech256* (**C**), *ImageNet ILSVRC 2012* (**I**), and *Pascal VOC 2012* (**P**). They are in 12 common categories shared by the three datasets, with 50 images in each category. We evaluate our method on the transfer tasks with all domain combinations: $\mathbf{I} \to \mathbf{P}$, $\mathbf{P} \to \mathbf{I}$, $\mathbf{I} \to \mathbf{C}$, $\mathbf{C} \to \mathbf{I}$, $\mathbf{C} \to \mathbf{P}$ and $\mathbf{P} \to \mathbf{C}$. The other dataset is *Office-31* [29], which consists of totally $4,110$ images from 31 categories. All the images are collected from three different domains: *Amazon* (**A**), which are downloaded from amazon.com, *DSLR* (**D**), which are taken by digital SLR camera and *Webcam* (**W**), which are recorded with a simple webcam. This dataset with images from different photographical settings represent visual domain shifts. We evaluate our method in terms of classification accuracy on all the six transfer tasks $\mathbf{A} \to \mathbf{W}$, $\mathbf{D} \to \mathbf{W}$, $\mathbf{W} \to \mathbf{D}$, $\mathbf{W} \to \mathbf{D}$, $\mathbf{A} \to \mathbf{D}$ and $\mathbf{D} \to \mathbf{A}$.

Impementation details The training and testing are implemented by **Pytorch**. Among all the transfer tasks, we use stochastic gradient descent (SGD) with momentum of 0.9 [31] for minimizing the loss function given by equation (11). In *Office-31*, we adopt balanced sampling between classes to increase the chance that samples from each category can be drawn in each batch. The learning rate is initialized with 0.0005 for all the CNN layers and 0.005 for all the fully connected layers, and then exponentially decayed during SGD by a factor $(1+\alpha p)^{\beta}$, where $\alpha = 10$, $\beta = 0.75$ and $p \in [0,1]$ is the training progress measured by epoch numbers [12]. All weights and biases are regularized by a weight decay with L_2 penalty multiplier set to 0.0005. λ_{adv} for adversarial training is fixed with 1, decayed with a factor $\frac{1-exp(-10p)}{1+exp(-10p)}$ through the training process, while λ_R is fixed with 0.01 across all the experiments [28]. We implement our method based on the *ResNet-50* [19] pre-trained on the ImageNet dataset as is done in the compared deep learning methods [9, 24, 28].

[6] http://imageclef.org/2014/adaptation
[7] Reimplementation.

Table 1. Mean accuracy (%) on *ImageCLEF-DA* for UDA (*ResNet-50*)

Method	I→P	P→I	I→C	C→I	C→P	P→C	Average
ResNet [19]	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN [23]	75.0	86.2	93.3	84.1	69.8	91.3	83.3
RTN [25]	75.6	86.8	95.3	86.9	72.7	92.2	84.9
DANN [12]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN [26]	76.8	88.0	94.7	89.5	74.2	91.7	85.8
CAN [36]	78.2	87.5	94.2	89.5	75.8	89.2	85.7
MADA [28]	75.0	87.9	96.0	88.8	75.2	92.2	85.8
RADA$_{y\to d}$	78.8	92.1	97.3	90.9	76.4	94.6	88.4
RADA$_{d\to y}$	**79.2**	**92.4**	**97.5**	**91.1**	**76.6**	**95.3**	**88.7**

Table 2. Mean accuracy (%) on *Office-31* for UDA (*ResNet-50*)

Method	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet [19]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
TCA [27]	74.7	96.7	99.6	76.1	63.7	62.9	79.3
GFK [15]	74.8	95.0	98.2	76.5	65.4	63.0	78.8
DDC [34]	75.8	95.0	98.2	77.5	67.4	64.0	79.7
DAN [23]	83.8	96.8	99.5	78.4	66.7	62.7	81.3
RTN [25]	84.5	96.8	99.4	77.5	66.2	64.8	81.6
DANN [12]	82.0	96.9	99.1	79.7	68.2	67.4	82.2
ADDA [33]	86.2	96.2	98.4	77.8	69.5	68.9	82.9
JAN [26]	85.4	97.4	99.8	84.7	68.6	70.0	84.3
JDDA [6]	82.6	95.2	99.7	79.8	57.4	66.7	80.2
CAN [36]	81.5	98.2	99.7	85.5	65.9	63.4	82.4
MADA [28]	90.0	97.4	99.6	87.8	70.3	66.4	85.2
RADA$_{y\to d}$	**91.5**	**99.0**	**100.0**	90.3	**71.5**	70.1	87.1
RADA$_{d\to y}$	**91.5**	98.9	**100.0**	**90.7**	**71.5**	**71.3**	**87.3**

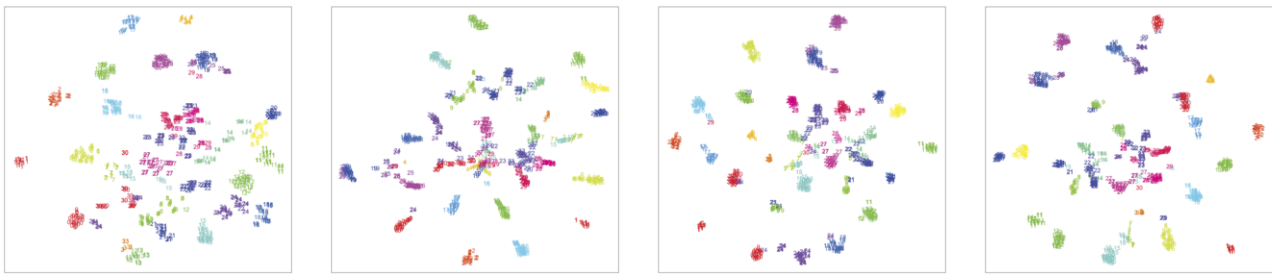
Table 3. Mean accuracy (%) on *Office-31* for PDA from 31 classes to 10 classes (*ResNet-50*)

Method	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet [19]	54.5	94.6	94.3	65.6	73.2	71.7	75.6
DAN [23]	46.4	53.6	58.6	42.7	65.7	65.3	55.4
ADDA [33]	43.7	46.5	40.1	43.7	42.8	46.0	43.8
RTN [25]	75.3	97.1	98.3	66.9	85.6	85.7	84.8
JAN [26]	43.4	53.6	41.4	35.7	51.0	51.6	46.1
DANN [12]	41.4	46.8	38.9	41.4	41.3	44.7	42.4
MADA [28] [7]	63.5	85.1	**99.7**	67.7	59.1	63.9	73.2
RADA$_{y\to d}$	**83.0**	**97.4**	97.2	**87.4**	86.0	85.5	89.4
RADA$_{d\to y}$	82.8	**97.4**	97.6	86.8	**86.6**	**86.3**	**89.6**

Baselines We follow standard evaluation protocols for UDA using all labeled source samples and all unlabeled target samples, and report the mean classification accuracy over three random experiments [24, 28]. We compare RADA with recent state-of-the-art deep transfer learning methods based on *ResNet-50*: Deep Domain Confusion (DDC) [34], Deep Adaptation Network (DAN) [23], Residual Transfer Network (RTN) [25], DANN [12], Adversarial Discriminative Domain Adaptation (ADDA) [33], Joint Adaptation Network (JAN) [26], MADA [28], Collaborative and Adversarial Network (CAN) [36], and Joint Discriminative Domain Adaptation (JDDA) [6]; and traditional machine learning methods: Transfer Component Analysis (TCA) [27], Geodesic Flow Kernel (GFK) [15].

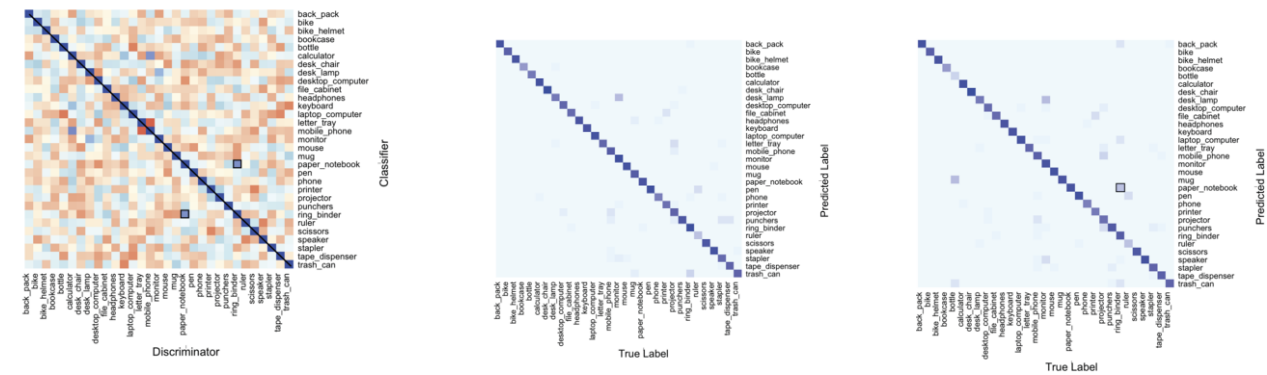
4.2 Main results

The mean classification accuracy on *ImageCLEF-DA* and *Office-31* is reported in Table 1 and Table 2. The results of baseline methods are reprinted from previous literatures [28, 24, 6, 36, 4]. RADA$_{d\to y}$ and RADA$_{y\to d}$ are our methods trained with $D_{KL}(\mathbf{\Omega_y}||\mathbf{\Omega_d})$ and $D_{KL}(\mathbf{\Omega_d}||\mathbf{\Omega_y})$. As shown in Table 1, RADA$_{d\to y}$ and RADA$_{y\to d}$ both outperform baseline methods across all the transfer tasks for both *ImageCLEF-DA* and *Office-31*. RADA$_{d\to y}$ slightly outperforms RADA$_{y\to d}$. This very similar performance suggests the choice of the target matrix does not make much difference in spite of the asymmetry of the adopted discrepancy metric, thus either of these two forms can be used in the RADA algorithm. The number of parameters used by RADA is also reduced to a large extent, compared with the multiple discriminators method (e.g. MADA). For *ImageCLEF-DA* and



(a) DANN (b) MADA (c) RADA$_{y\to d}$ (d) RADA$_{d\to y}$

Figure 3. The t-SNE visulization of embedded features from target domain.



(a) Heatmap of $\hat{\rho}_{ij}$ (blue: ≥ 0, red: < 0) (b) Confusion matrix of RADA$_{d\to y}$ (c) Confusion matrix of MADA

Figure 4. Visualization for characterized class relationships and confusion matrix for task $\mathbf{A} \to \mathbf{W}$

Office31 datasets, the adoption of 12 and 31 independent two-layer discriminators ($x \to 1024 \to 1024 \to 1$) generates more than 10^7 parameters, whereas RADA only generates $\sim 10^6$ parameters. This reduction is important in practice, especially when the number of classes is very large (e.g. > 100). The improved performance with even simpler network highlights the significance of incorporating class relationships into the adversarial training process. The alignment of class relationships between label predictor and domain discriminator introduces more structure information from the label space to the adversarial training process, and efficiently promotes the learning of transferable representations for feature extractor. We also include an ablation study of our method on different modules in the appendix.

We additionally provide evaluations for *partial domain adaptation* (PDA) problem, where the target label space is a subset of source label space. It is a new technical bottleneck, which is more challenging and practical than the standard domain adaptation, considering the outlier classes in the source domain can cause negative transfer when discriminating the target classes [4, 28]. To show the robustness of our method against PDA, we implement the evaluation in a benchmark experimental setup. From *Office-31*, we use all the categories for the source domain and choose the ten categories shared with *Caltech256* [17] for the target domain. Among all the transfer tasks, the source domain contains 31 classes and the target domain has 10 classes. From Table 3, we can observe that RADA outperforms *ResNet* and other general domain adaptation methods, especially on the tasks $\mathbf{A} \to \mathbf{W}$, $\mathbf{A} \to \mathbf{D}$, $\mathbf{W} \to \mathbf{A}$ and $\mathbf{D} \to \mathbf{A}$, which suggests it successfully avoids the negative transfer trap.

4.3 Empirical analysis

Feature visualization In order to visualize the embedded data, we use t-SNE to project the feature representations after *pool5* in *ResNet-50* that are respectively trained with DANN, MADA and RADA to lower dimensional space. The two-dimensional map of embedded data in target domain from the transfer task $\mathbf{A} \to \mathbf{W}$ is visualized in Figure 3, where the class information is also given by assigning data points with different colors and numerical labels in the plot. We observe that the embedded features from different classes are better separated in RADA and MADA when compared with DANN. Although MADA is able to separate most of the data points according to their class labels, several classes around the center are still mixed up, while RADA can better separate those points. By integrating the information of class relationships, RADA can better extract the features uniquely belonging to each class and capture the modes of the data distribution.

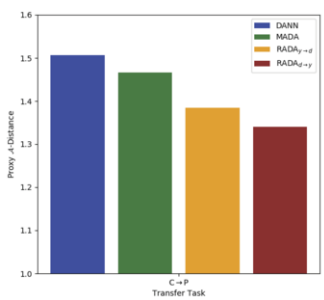


Figure 5. Proxy $\mathcal{A}$-Distance

Class relationships Partial correlation is a symmetric measure of association between two variables while controlling the effect of other variables. It is commonly used to model the conditional dependencies among a group of variables. The partial correlation between class i and j can be calculated by $\rho_{ij} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}$ from the element ω_{ij} of precision matrix Ω. With the estimated $\hat{\mathbf{\Omega}}_\mathbf{y}$ and $\hat{\mathbf{\Omega}}_\mathbf{d}$ from the transfer task $\mathbf{A} \rightarrow \mathbf{W}$ using RADA$_{d\rightarrow y}$, we calculate and visualize the partial correlations among all the classes in Figure 4, where $\hat{\rho}_{ij}$ estimated from label predictor is displayed on the upper triangular part and that from the discriminator is on the lower triangular part. The symmetry of the heatmap in Figure 4a indicates our regularization successfully encourages the class relationships to be consistent between G_d^* and G_y. Some class relationships are interesting and intuitive. For example, not surprisingly, the class *paper notebook* is found to be positive associated with *ring binder* from both label predictor and discriminator with RADA (black framed cell in Figure 4a). Aware of such class relationships, our method avoids miss-classifying several images (Figure 4b) of *ring binder* as *paper notebook* compared to MADA (black framed cell in Figure 4c).

Distribution discrepancy Proxy $\mathcal{A}$-Distance (PAD) [2, 12] is a widely used metric to measure the feature distributional discrepancy between source and target domains. PAD is defined as $d_\mathcal{A} = 2(1-2\epsilon)$, where ϵ is the classification error (e.g. mean absolute error) of a domain classifier (e.g. SVM). Generally, a lower PAD indicates a better generalization ability. As shown in Figure 5, on the transfer task $\mathbf{C} \rightarrow \mathbf{P}$, RADA outperforms DANN and MADA. This indicates RADA can better extract domain-invariant features. In addition, $d_\mathcal{A}$ of RADA$_{d\rightarrow y}$ are slightly lower than RADA$_{y\rightarrow d}$, showing that RADA$_{d\rightarrow y}$ has a better generalization ability.

5 CONCLUSION

We present a novel approach to domain adaptation through revealing the structure information from the label space for aligning complicated data distributions during adversarial training. We propose a new design of multi-class domain discriminator and a novel regularizer to align the inter-class dependencies respectively characterized from label predictor and domain discriminator. Experiments show considering class relationship information can substantially improve the transfer learning performance. In this work, we model the class relationships in terms of symmetric relationships, and we will consider extending to a model that allows the class relationships to be asymmetric in our future study.

6 APPENDIX

6.1 Detailed derivation of solution to equation (5)

With the cyclic property of the trace, we have an exponential form of the objective function as:

$$\begin{aligned} &\min_{\mathbf{\Omega_y}} -d_y \log\det(\mathbf{\Omega_y}) + \mathrm{Tr}(\mathbf{W_y}^{[L]}\mathbf{\Omega}{\mathbf{W_y}^{[L]}}^T) \\ \Leftrightarrow &\min_{\mathbf{\Omega_y}} -\det(\mathbf{\Omega_y})^{d_y}\exp(-\mathrm{Tr}(\mathbf{\Omega_y}{\mathbf{W_y}^{[L]}}^T\mathbf{W_y}^{[L]})) \end{aligned} \tag{12}$$

Let us denote $\mathbf{H} = {\mathbf{W_y}^{[\mathbf{L}]}}^T\mathbf{W_y}^{[L]}$, which is assumed as a positive definite matrix, we have:

$$\min_{\mathbf{\Omega_y}} -\det(\mathbf{\Omega_y})^{d_y}\exp(-\mathrm{Tr}(\mathbf{\Omega_y}\mathbf{H})) \tag{13}$$

According to the spectral theorem in linear algebra, when $\mathbf{H}$ is a positive-definite symmetric matrix, then it has precisely one positive-definite symmetric square root which we denote as $\mathbf{H}^{\frac{1}{2}}$. Thus with the cyclic property of the trace, equation (13) can be rewritten to:

$$\min_{\mathbf{\Omega_y}} -\det(\mathbf{\Omega_y})^{d_y}\exp(-\mathrm{Tr}(\mathbf{H}^{\frac{1}{2}}\mathbf{\Omega_y}\mathbf{H}^{\frac{1}{2}})) \tag{14}$$

Let $\mathbf{U} = \mathbf{H}^{\frac{1}{2}}\mathbf{\Omega_y}\mathbf{H}^{\frac{1}{2}}$ and solve:

$$\begin{aligned} &\min_{\mathbf{U}} -\det(\mathbf{H})^{-d_y}\det(\mathbf{U})^{d_y}\exp(-\mathrm{Tr}(\mathbf{U})) \\ \Leftrightarrow &\min_{\mathbf{U}} -\det(\mathbf{U})^{d_y}\exp(-\mathrm{Tr}(\mathbf{U})) \end{aligned} \tag{15}$$

Let $\eta_1, \cdots, \eta_K \geq 0$ be the eigenvalues of matrix $\mathbf{U}$, then we have $\det(\mathbf{U}) = \prod_i^K \eta_i$ and $\mathrm{Tr}(\mathbf{U}) = \sum_i^K \eta_i$. Equation (15) reduces to the problem of finding η_i that:

$$\min_{\eta_i} -{\eta_i}^{d_y}\exp(-\eta_i), \forall i \tag{16}$$

With calculus, it is easy to get $\forall i$, $\eta_i = d_y$. Assuming $\mathbf{V}$ is the matrix of eigen vectors of $\mathbf{U}$, we have:
$\mathbf{U} = \mathbf{V}(d_y\mathbf{I}_K)\mathbf{V}^{-1} = d_y\mathbf{I}_K$ Finally we get:

$$\begin{aligned} &\mathbf{H}^{\frac{1}{2}}\mathbf{\Omega_y}\mathbf{H}^{\frac{1}{2}} = d_y\mathbf{I}_K \\ \Rightarrow &\mathbf{\Omega_y} = (\mathbf{H}^{-1})^{\frac{1}{2}}d_y\mathbf{I}_K(\mathbf{H}^{-1})^{\frac{1}{2}} \\ \Rightarrow &\mathbf{\Omega_y} = d_y\mathbf{H}^{-1} = d_y({\mathbf{W_y}^{[L]}}^T\mathbf{W_y}^{[L]})^{-1} \end{aligned} \tag{17}$$

6.2 Ablation study

In order to provide an ablation study of our models, we conducted experiments with the same experimental setting on *Office-31* that dropped the regularization item L_R from our method to examine the effect of aligning inter-class dependencies in addition to using the single multi-class domain discriminator. The ablation study is reported in Table 4, where the use of the single multi-class domain discriminator without the regularization item L_R is noted as *Only Multi-class Discriminator*. We find that the only use of a multi-class discriminator is better than MADA, and both RADA$_{d\rightarrow y}$ and RADA$_{y\rightarrow d}$ outperform the utilization of only a multi-class discriminator, implying that the alignment of class relationships can effectively improve the adaptation performance.

Table 4. Ablation study: mean accuracy (%) on *Office-31* for UDA (*ResNet-50*)

Method	MADA	*Only Multi-class* Discriminator	RADA$_{y\rightarrow d}$	RADA$_{d\rightarrow y}$
A→W	90.0	90.0	91.5	91.5
D→W	97.4	99.0	99.0	98.9
W→D	99.6	100.0	100.0	100.0
A→D	87.8	88.9	90.3	90.7
D→A	70.3	70.5	71.5	71.5
W→A	66.4	68.5	70.1	71.3
Average	85.2	86.2	87.1	87.3

REFERENCES

[1] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang, 'Generalization and equilibrium in generative adversarial nets (gans)', in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 224–232. JMLR. org, (2017).

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira, 'Analysis of representations for domain adaptation', in *Advances in Neural Information Processing Systems*, pp. 137–144, (2007).

[3] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao, 'Learning disentangled semantic representation for domain adaptation', in *IJCAI: proceedings of the conference*, volume 2019, p. 2060. NIH Public Access, (2019).

[4] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang, 'Partial adversarial domain adaptation', in *European Conference on Computer Vision*, pp. 135–150, (2018).

[5] Rich Caruana, 'Multitask learning', *Machine Learning*, **28**(1), 41–75, (1997).

[6] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin, 'Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation', in *AAAI Conference on Artificial Intelligence*, (2019).

[7] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun, 'No more discrimination: Cross city adaptation of road scene segmenters', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1992–2001, (2017).

[8] Xiangzhao Cui, Chun Li, Jine Zhao, Li Zeng, Defei Zhang, and Jianxin Pan, 'Covariance structure regularization via frobenius-norm discrepancy', *Linear Algebra and its Applications*, **510**, 124–145, (2016).

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, 'Imagenet: A large-scale hierarchical image database', in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, (2009).

[10] Nanqing Dong, Michael Kampffmeyer, Xiaodan Liang, Zeya Wang, Wei Dai, and Eric Xing, 'Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio', in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 544–552. Springer, (2018).

[11] Yaroslav Ganin and Victor Lempitsky, 'Unsupervised domain adaptation by backpropagation', in *International Conference on Machine Learning*, pp. 1180–1189, (2015).

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, 'Domain-adversarial training of neural networks', *Journal of Machine Learning Research*, **17**(1), 2096–2030, (2016).

[13] André R Gonçalves, Puja Das, Soumyadeep Chatterjee, Vidyashankar Sivakumar, Fernando J Von Zuben, and Arindam Banerjee, 'Multi-task sparse structure learning', in *International Conference on Information and Knowledge Management*, pp. 451–460. ACM, (2014).

[14] André R Gonçalves, Fernando J Von Zuben, and Arindam Banerjee, 'Multi-task sparse structure learning with gaussian copula models', *Journal of Machine Learning Research*, **17**(1), 1205–1234, (2016).

[15] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman, 'Geodesic flow kernel for unsupervised domain adaptation', in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073. IEEE, (2012).

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, 'Generative adversarial nets', in *Advances in Neural Information Processing Systems*, pp. 2672–2680, (2014).

[17] Gregory Griffin, Alex Holub, and Pietro Perona, 'Caltech-256 object category dataset', (2007).

[18] Lan-Zhe Guo and Yu-Feng Li, 'A general formulation for safely exploiting weakly supervised data', in *Thirty-Second AAAI Conference on Artificial Intelligence*, (2018).

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, (2016).

[20] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang, 'Exploiting feature and class relationships in video categorization with regularized deep neural networks', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**(2), 352–364, (2018).

[21] Baoyu Jing, Chenwei Lu, Deqing Wang, Fuzhen Zhuang, and Cheng Niu, 'Cross-domain labeled lda for cross-domain text classification', in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 187–196. IEEE, (2018).

[22] Yu-Feng Li, Lan-Zhe Guo, and Zhi-Hua Zhou, 'Towards safe weakly supervised learning', *IEEE transactions on pattern analysis and machine intelligence*, (2019).

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan, 'Learning transferable features with deep adaptation networks', in *International Conference on Machine Learning*, pp. 97–105, (2015).

[24] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan, 'Conditional adversarial domain adaptation', in *Advances in Neural Information Processing Systems*, pp. 1640–1650, (2018).

[25] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan, 'Unsupervised domain adaptation with residual transfer networks', in *Advances in Neural Information Processing Systems*, pp. 136–144, (2016).

[26] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan, 'Deep transfer learning with joint adaptation networks', in *International Conference on Machine Learning*, pp. 2208–2217. JMLR. org, (2017).

[27] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang, 'Domain adaptation via transfer component analysis', *IEEE Transactions on Neural Networks*, **22**(2), 199–210, (2011).

[28] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang, 'Multi-adversarial domain adaptation', in *AAAI Conference on Artificial Intelligence*, (2018).

[29] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell, 'Adapting visual category models to new domains', in *European Conference on Computer Vision*, pp. 213–226. Springer, (2010).

[30] Baochen Sun, Jiashi Feng, and Kate Saenko, 'Return of frustratingly easy domain adaptation', in *Thirtieth AAAI Conference on Artificial Intelligence*, (2016).

[31] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, 'On the importance of initialization and momentum in deep learning', in *International Conference on Machine Learning*, pp. 1139–1147, (2013).

[32] A Torralba and AA Efros, 'Unbiased look at dataset bias', in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1521–1528. IEEE Computer Society, (2011).

[33] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, 'Adversarial discriminative domain adaptation', in *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, p. 4, (2017).

[34] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell, 'Deep domain confusion: Maximizing for domain invariance', *arXiv preprint arXiv:1412.3474*, (2014).

[35] Tong Wei, Lan-Zhe Guo, Yu-Feng Li, and Wei Gao, 'Learning safe multi-label prediction for weakly labeled data', *Machine Learning*, **107**(4), 703–725, (2018).

[36] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu, 'Collaborative and adversarial network for unsupervised domain adaptation', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3801–3809, (2018).

[37] Yu Zhang and Dit-Yan Yeung, 'A convex formulation for learning task relationships in multi-task learning', in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pp. 733–742, (2010).