Dual Alignment for Partial Domain Adaptation

Lusi Li, Zhiqiang Wan, *Student Member, IEEE*, and Haibo He, *Fellow, IEEE*

Abstract—**Partial domain adaptation (PDA) aims to transfer knowledge from a label-rich source domain to a label-scarce target domain based on an assumption that the source label space subsumes the target label space. The major challenge is to promote positive transfer in the shared label space and circumvent negative transfer caused by the large mismatch across different label spaces. In this article, we propose a dual alignment approach for PDA (DAPDA), including three components: 1) a feature extractor extracts source and target features by the Siamese network; 2) a reweighting network produces "hard" labels, class-level weights for source features and "soft" labels, instance-level weights for target features; 3) a dual alignment network aligns intra domain and interdomain distributions. Specifically, the intra domain alignment aims to minimize the intraclass variances to enhance the intraclass compactness in both domains, and interdomain alignment attempts to reduce the discrepancies across domains by domain-wise and class-wise adaptations. The negative transfer can be alleviated by down-weighting source features with nonshared labels. The positive transfer can be enhanced by upweighting source features with shared labels. The adaptation can be achieved by minimizing the discrepancies based on class-weighted source data with hard labels and instance-weighed target data with soft labels. The effectiveness of our method has been demonstrated by outperforming state-of-the-art PDA methods on several benchmark datasets.**

Index Terms—**Dual alignment, partial domain adaptation (PDA), reweighting network, Siamese network.**

I. Introduction

DEEP neural networks significantly improve classification accuracy, which are trained via representation learning on large-scale labeled training data and tested on the data with similar distribution. However, acquiring numerous labeled training data is a time consuming and expensive task for various applications [1], [2]. Hence, in order to alleviate the labeling time and cost, domain adaptation is proposed to address this problem by leveraging label-rich data (i.e., source domain) to related label-scarce data (i.e., target domain) [3], [4].

Most of the existing domain adaptation methods generally assume source and target domains are related by sharing identical label space but separated by different data distributions (i.e., distribution shift), resulting in weak generalization ability of models on target data [5], [6]. They focus on matching marginal distributions since the differences might seem small due to the same label space [7]. One effective strategy is to estimate important weights of source samples related to target samples such that their shared similar distribution can be obtained [8]. Another successful strategy bridges different domains by learning domain-invariant features to reduce data distribution divergence [9], [10]. Recent studies [11], [12] have shown that more transferable representative features can be learned by deep neural networks [13], [14]. Extracting domain-invariant representations by embedding deep representation learning in the pipeline of domain adaptation has achieved certain latest advances [15], [16].

Partial domain adaptation (PDA), as a more practical and challenging problem, assumes the target label space is subsumed into the source label space. In an unsupervised scenario, the target domain only has non-labeled data and the shared label space across domains is unknown. Thus, PDA has another technical challenge: how to alleviate the negative transfer caused by the outlier source classes. Recently, there are four related methods, including the importance weighted adversarial nets (IWANs) [17], selective adversarial network (SAN) [18], partial adversarial domain adaptation (PADA) [19], and example transfer network (ETN) [20]. They had some success in addressing the PDA by weighing each sample in the domain-adversarial networks and matching either marginal or conditional distributions to align the source domain as well as the target domain. However, they do not explore the role of each sample, ignore the joint distributions, and do not consider latent structures underlying distributions. In this article, we propose reweighting all source and target samples and then match the marginal distributions together with the conditional distributions, which can better determine the outlier source classes and align source and target domains.

A dual alignment approach for PDA (DAPDA) is presented, which improves the previous works [17]–[20] by jointly exploring the contribution of each sample, matching joint distributions, and learning latent structures underlying distributions. Our proposed DAPDA method consists of a feature extractor, a reweighting network, and a dual alignment network (shown in Fig. 1). The feature extractor M implemented by the Siamese network embeds input samples from source and target domains into latent feature representations. Therein, the Siamese network has two identical subnetworks, where the weights θ_g are shared. This special characteristic of the Siamese network makes it possible to discover the discrepancies between the source and target domains. The discrepancies will be minimized during the training process such that the data from the source and target domains can be mapped

Manuscript received July 31, 2019; revised December 23, 2019 and February 18, 2020; accepted March 11, 2020. This work was supported in part by the National Science Foundation under Grant ECCS 1917275. This article was recommended by Associate Editor Y. S. Ong. *(Corresponding author: Haibo He.)*

The authors are with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: lli@ele.uri.edu; zwan@ele.uri.edu; haibohe@uri.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.2983337

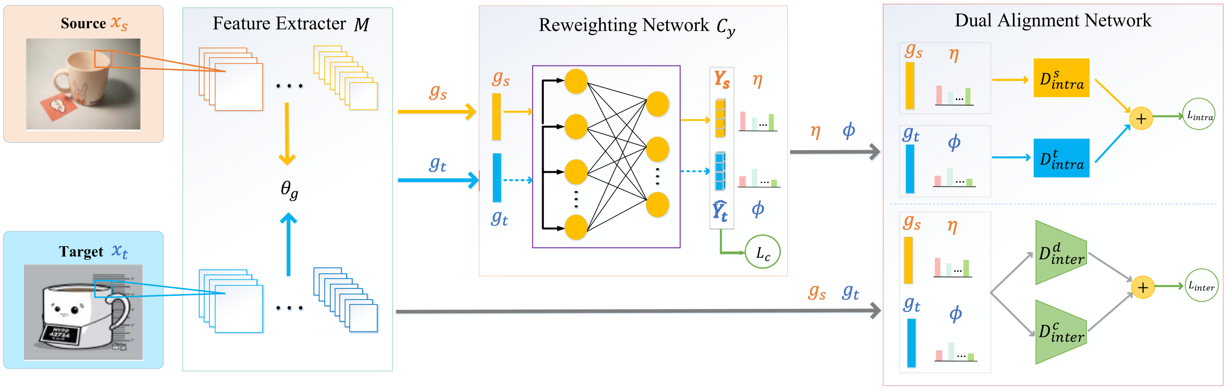


Fig. 1. Framework of our proposed DAPDA method.

into the same latent space, that is, an intermediate domain. With the features extracted by the Siamese network (g_s and g_t), we train a reweighting network C_y with labeled source data. Then, the trained reweighting network is used to generate hard labels Y_s, class-level weights η for source features and soft labels $\widehat{Y}_t$, and instance-level weights ϕ for target features. Note that DAPDA improves the reweighting quality over ETN [20] by further learning the contributions of target samples to the domain alignment. The dual alignment network aims to match intradomain and inter domain distributions based on the Wasserstein distance, which is a metric measuring the difference between distributions. Specifically, intradomain alignments (D^s_{intra} and D^t_{intra}) would like to minimize the intra class distances in both source domain and target domain, and interdomain alignments (D^d_{inter} and D^c_{inter}) attempt to reduce the discrepancies across domains and that within the same classes from different domains. For our proposed DAPDA method, given source and target domains, we match joint distributions to obtain an intermediate domain, where the learned features from both domains would be class discriminative and domain invariant. In such a way, the negative transfer can be alleviated by class-wisely downweighting source features with nonshared labels; positive transfer can be enhanced by class-wisely upweighting source features with shared labels as well as simultaneously aligning intradomain and inter domain distributions; domain-invariant feature representations can be learned through the Siamese network in the shared label space.

The main contributions are highlighted as follows.

1) We design a reweighting network in DAPDA to give class-level weights to source features and instance-level weights as well as soft labels to target features. The outlier source classes can be downweighted based on the low class-level weights. The more the target instances are similar to the source domain, the higher the instance-level weights they can have. With each iteration of our proposed method, the errors introduced from wrongly predicted target labels can be reduced.
2) We propose a dual alignment network in DAPDA to match joint distributions between domains. It minimizes the intra class variances in the source domain based on labeled source data, the intraclass variances in the target domain based on the weighted target data with soft labels, the distances across domains, and the discrepancies within the same class from different domains.
3) DAPDA combines both the domain-shared and domain-specific information to learn domain-invariant and class-discriminative feature representations. The proposed model outperforms the existing PDA approaches. Good adaptation is achieved in simulations.

The remainder of this article is organized as follows. Section II reviews background on domain adaptation and discrepancy metrics. In Section III, we give the details of the proposed method. Section IV presents the experiments on real-world datasets. In Section V, we provide the conclusion.

II. Related Work

Supervised learning has superiority in representation learning. However, the large labeling time and cost hinders its development [21]–[23]. Unsupervised learning can discover the hidden patterns without labels [24], [25]. Reinforcement learning performs a certain goal by interacting with a dynamic environment [26], [27]. In this article, we aim to use unsupervised learning techniques to find the shared space between source and target domains.

A. Domain Adaptation

Recent studies focus on transferring feature representations learned by deep neural networks from a labeled source domain to an unlabeled target domain. One effective strategy is to map the features from two different domains into a common latent space, in which the corresponding feature distributions are close [28]. The maximum mean discrepancy (MMD) [29] has been used in several approaches for this purpose. It can measure the divergence between two distribution means in reproducing kernel Hilbert space (RKHS). In residual transfer network (RTN) [30], the MMD criterion is used to match distributions for feature adaptation, where the features are fused by the output of multiple layers with the tensor product.

An adversarial objective is also used to minimize domain discrepancy. Tzeng *et al.* [31] presented a generalized framework, adversarial discriminative domain adaptation (ADDA), including a domain discriminator, target weight sharing, and an adversarial loss.

Another class of divergences between two data distributions is optimal transport (OT) [32], [33], where the Wasserstein distance induced by OT has been successfully applied to domain adaptation due to its generalization [34]–[36]. Shen *et al.* [14] proposed the Wasserstein distance guided representation learning (WDGRL) to learn domain-invariant feature representations by evaluating and minimizing improved Wasserstein distance across domains. However, all these methods aim to match marginal distributions in the identical label space.

For PDA problem, four existing methods are IWAN [17], SAN [18], PADA [19], and ETN [20]. They address the PDA by weighing each sample in the domain-adversarial networks and matching either marginal or conditional distributions to align the source domain as well as the target domain.

Specifically, IWAN trains the first domain classifier to reweight source domain samples, and matches marginal distributions via a feature extractor as well as the second domain classifier in an adversarial manner. Our proposed method differs from this article.

1) For the source domain, the class-level weights are not considered. It may raise the problem of not completely selecting out the outlier source samples, resulting in performance degradation. In contrast, we weight the source domain samples with the average class probabilities over all target samples, which are given by the reweighting network, such that domain-invariant features in the shared label space can be learned.
2) For the target domain, the instance-level weights are not taken into account. The shared-label source samples may be forcefully aligned to the noise target samples. Our DAPDA method estimates the instance-level weights to target samples reducing the negative effect of noise target samples in the dual alignment network.
3) Conditional distributions are also not considered in IWAN. Instead, we not only finely grained align source and target domains by category to capture latent structures underlying conditional distributions but also enhance the intraclass compactness in both domains by minimizing the intra class variances.

The other three works are proposed by Cao *et al.* SAN matches conditional distributions across domains by training multiple domain classifiers and down-weighting outlier source classes with both class-level and instance-level weights. PADA and ETN focus on matching marginal contributions by training one whole domain classifier and down-weighing source outlier classes only with source sample weights, where ETN can automatically obtain source sample weights based on their similarities to the target domain and use the obtained weights in the source classifier as well as domain-adversarial network.

Our proposed method is different from these three works.

1) We match joint distributions, starting with the interdomain alignment and then capturing latent structures by the intradomain alignment, instead of only matching either marginal or conditional contributions. Specifically, the inter domain alignment, including domain-wise and class-wise alignments contributes to matching both domains in all aspect view.
2) We use both class- and instance-level weights and hence are capable of selecting out outlier source samples as well as dealing with imbalanced, noise target data. Since if class-level weights are not applied, outlier source samples may not be picked out completely leading to certain negative transfer. In addition, if instance-level weights are not applied, the target samples in negative class (i.e., the class with less samples) may not be classified well for the imbalanced issue; the noise target samples will be forcefully aligned to source samples for the noise issue. However, instance-level weights are not considered in PADA and ETN.
3) We assign target samples with soft labels and instance-level weights according to maximum class probabilities of softmax output offered by the reweighting network. Then, the distribution discrepancy for each class can be measured by the Wasserstein distance between the weighted source and target features. While SAN does not assign labels to target samples and enables all target samples to participate in the domain classification in each domain classifier, which makes SAN hardly scalable to a large source data and increases computational complexity.

B. Discrepancy Metric

In this article, we use the Wasserstein distance as the discrepancy metric of distributions, which is also known as the Kantorovich–Monge–Rubinstein metric on a given metric space $\mathcal{M}$ and arises from the idea of OT.

Let $X \sim \mathbb{P}$ and $Y \sim \mathbb{Q}$. We assume that $X, Y \in \mathbb{R}^d$. The Wasserstein distance of order σ between two Borel probability distribution measures $\mathbb{P}$ and $\mathbb{Q}$ on $\mathcal{M}$ is defined as

$$\begin{aligned} W_\sigma(\mathbb{P}, \mathbb{Q}) &= \left(\inf_{\mu \in \Omega(\mathbb{P}, \mathbb{Q})} \int \tau(x, y)^\sigma \, d\,\mu(x, y) \right)^{1/\sigma} \\ &= \inf_{\mu \in \Omega(\mathbb{P}, \mathbb{Q})} \left(\mathbb{E}_{x \sim \mathbb{P}, y \sim \mathbb{Q}} \, \tau(x, y)^\sigma \right)^{1/\sigma} \end{aligned} \tag{1}$$

where $\sigma \geq 1$; $\mathbb{P}, \mathbb{Q} \in \{\mathbb{P} : \int \tau(x, y)^\sigma \, d\,\mathbb{P}(x) < \infty \quad \forall y \in M\}$; $\Omega(\mathbb{P}, \mathbb{Q})$ denotes all joint distributions μ for (X, Y) with marginal distributions $\mathbb{P}$ and $\mathbb{Q}$; τ is a distance and $\tau(x, y)^\sigma$ is the corresponding unit cost function; $\mu(x, y)$ can be viewed as a joint probability measure in $\Omega(\mathbb{P}, \mathbb{Q})$, and indicates that how much "mass" would be transported from a random location x to another one y on $\mathcal{M}$ such that $\mathbb{P}$ can be transformed into $\mathbb{Q}$. From the above, given a unit cost $\tau(x, y)^\sigma$, we can effectively transform $\mathbb{P}$ into $\mathbb{Q}$ at the minimum expected transport cost $W_\sigma(\mathbb{P}, \mathbb{Q})$.

When $\mathcal{M}$ is separable and $\sigma = 1$, (1) is also called Earth-mover distance [37]. It can be written as follows:

$$W_1(\mathbb{P}, \mathbb{Q}) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}}\big[f(x)\big] - \mathbb{E}_{y \sim \mathbb{Q}}\big[f(y)\big] \tag{2}$$

where f denotes all maps from $\mathbb{R}^d$ to $\mathbb{R}$; $\tau(x, y) \geq |f(x) - f(y)|$ for all x, y; the Lipschitz seminorm is thus defined as $\|f\|_L = \sup|f(x) - f(y)|/\tau(x, y)$. To enforce the Lipschitz constraint, Arjovsky *et al.* [36] proposed to use clipped weights within a compact space $[-c, c]$ after updating gradient. While Gulrajani *et al.* [38] pointed out the strategy of weight clipping

TABLE I
MAIN NOTATIONS AND DEFINITIONS IN THE PROPOSED METHOD

Notations	Definitions
D^s	Source domain
X_s	Source instances
Y_s	Source labels (i.e., “hard” labels)
C^s	Source label space
g_s	Source features
K	Source class number
D^t	Target Domain
X_t	Target instances
$\widehat{Y_t}$	Predicted target labels (i.e., “soft” labels)
C^t	Target label space
g_t	Target features
M	Siamese network
C_y	Reweighting network
H	Dual alignment network
$\widehat{q_t}$	Class distribution matrix of target data over C^s
η_{c_k}	Normalized class-level weight for source class c_k
ϕ_j	Instance-level weight for j-th target instance

would induce the issues of gradient vanishing or exploding without carefully tuning the threshold c. Thus, they use gradient penalty to enforce a soft constraint on the gradient norm for random samples $z \sim \mathbb{Z}$

$$W_1(\mathbb{P}, \mathbb{Q}) = \sup_{\|f\|_L \le 1} \mathbb{E}_{x\sim\mathbb{P}}\big[f(x)\big] - \mathbb{E}_{y\sim\mathbb{Q}}\big[f(y)\big] + \lambda \mathbb{E}_{z\sim\mathbb{Z}}\Big[(\|\nabla_z f(z)\|_2 - 1)^2\Big] \tag{3}$$

where z is sampled uniformly along the straight lines between the pairs of points x and y; λ is a balancing coefficient. The gradients are penalized at z. For simplicity, WD is the Wasserstein distance of order 1 in this article.

III. PROPOSED METHOD

For the problem of PDA in unsupervised scenario [17]–[20], we are given a sufficient labeled source dataset $(X_s, Y_s) = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ with n_s samples and known class set Y_s from source domain D^s, and an unlabeled target dataset $(X_t, Y_t) = \{x_t^j, y_t^j\}_{j=1}^{n^t}$ with n_t samples and unknown class set Y_t from target domain D^t. D^s and D^t share identical feature space but the label space of D^t is a subspace of that of D^s, that is, $C^t \subseteq C^s$. C^s can be splitted into source domain-specific (outlier) label space and source domain-invariant label space. In addition, D^s and D^t are, respectively, sampled from joint probability distributions $P(X_s, Y_s)$ and $Q(X_t, Y_t)$, where $P(X_s, Y_s) \neq Q(X_t, Y_t)$, furthermore, $P(X_s) \neq Q(X_t)$ and $P(X_s|Y_s) \neq Q(X_t|Y_t)$. We assume D^s have total K known classes $|C^s| = K$, and $|C^t|$ is unknown but $|C^t| \le |C^s|$. To describe our proposed method better, Table I shows the summary of main notations used in this article.

Our proposed DAPDA method attempts to use a feature extractor M, a reweighting network C_y, and a dual alignment network H to learn domain-invariant and class-discriminative feature representations as well as reduce joint distribution gaps across domains, such that the target risk $\Pr_{(x,y)\sim Q}[C_y(M(x_t)) \neq y_t]$ in the intermediate domain can be minimized based on the intra and interdomain alignments.

A. Feature Extractor

Siamese network, as the feature extractor, is used to extract domain-invariant features g, including two identical subnetworks: one for D^s and the other for D^t. *Identical* here indicates both subnetworks have the same parameters and weights. In the meantime, parameter updating is mirrored across them. Siamese network has superiority since: 1) fewer parameters are to be trained which in turn means less data are required and less tendency is overfitted and 2) similar model is used to process similar inputs if the inputs are of the same distribution, making feature representations with similar semantics and easier to compare. These special characteristics of the Siamese network make it possible to discover the discrepancy between source and target domains. This discrepancy will be minimized during the training process such that the data from the source domain and target domain can be mapped into the same latent space [18], [19]. Through the Siamese network, source features $g_s = M(x_s)$ and target features $g_t = M(x_t)$ are obtained, where each sample is mapped from an m- to a d-dimensional representation with the same parameter θ_g. θ_g can be optimized to enable the Siamese network to learn domain-invariant feature representations by feature mapping M, such that positive transfer can be promoted and negative transfer can be alleviated.

B. Reweighting Network

With the extracted source and target features from the Siamese network, the reweighting network trained with labeled source feature representations g_s can be applied to target feature representations g_t to predict their labels. Furthermore, the reweighting layer is added to give class-level weights to source features and instance-level weights to target features.

First, we train C_y to classify the source samples using the following supervised loss function [14], [18]:

$$\mathcal{L}_c(x_s, y_s) = \frac{1}{n_s}\sum_{i=1}^{n_s} L\big(C_y\big(g_s^i\big), y_s^i\big) \tag{4}$$

where L is the cross-entropy loss function. By minimizing softmax cross entropy to, respectively, learn the parameters θ_c of C_y, the objective function can be attained

$$\min_{\theta_c} \mathcal{L}_c. \tag{5}$$

Second, for source samples at the reweighting layer, we up-weight source samples in the shared label space and down-weight source samples in the nonshared label space. We call the class in the shared label space as shared class and the class in the nonshared label space as outlier class. In our proposed DAPDA method, we use C_y to determine whether a class is a shared one or not. Especially, we apply C_y to the target data X_t to obtain the predicted, that is, soft target labels $\widehat{Y_t}$

$$\widehat{Y_t} = C_y(g_t) \tag{6}$$

where $g_t = M(x_t)$ with parameter θ_g. In the meanwhile, C_y also gives a class probability distribution $\widehat{q_t^j}$ over source label space C^s for the jth target sample x_t^j. Thus, the distribution matrix of target data $\widehat{q_t}$ can be obtained as follows:

$$\widehat{q_t} = \begin{bmatrix} \widehat{q_t^1} \\ \widehat{q_t^2} \\ \vdots \\ \widehat{q_t^{n_t}} \end{bmatrix} = \begin{bmatrix} \widehat{q_{t,c_1}^1} & \widehat{q_{t,c_2}^1} & \cdots & \widehat{q_{t,c_K}^1} \\ \widehat{q_{t,c_1}^2} & \widehat{q_{t,c_2}^2} & \cdots & \widehat{q_{t,c_K}^2} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{q_{t,c_1}^{n_t}} & \widehat{q_{t,c_2}^{n_t}} & \cdots & \widehat{q_{t,c_K}^{n_t}} \end{bmatrix} \tag{7}$$

where $\widehat{q_t^j}$ represents the probability of assigning x_t^j to each of K classes and $j = 1, 2, \ldots, n_t$. There should be high probabilities of assigning target samples to the source shared classes. On the contrary, there should be low probabilities of assigning target samples to the source outlier classes. In order to identify the outlier classes, we average the class probabilities $\widehat{q_t}$ over all target samples to obtain the source class-level weights. The class with high weight is likely to be shared class while the class with low weight is likely to be outlier class. We then normalize these weights as follows:

$$\eta_{c_k} = \frac{\frac{1}{n_t}\sum_{j=1}^{n_t} \widehat{q_{t,c_k}^j}}{\frac{1}{n_t}\sum_{c_k \in C^s}\sum_{j=1}^{n_t} \widehat{q_{t,c_k}^j}} \tag{8}$$

where η_{c_k} is normalized source kth source class weight. $\eta = \{\eta_{c_1}, \eta_{c_2}, \ldots, \eta_{c_K}\}$ represent source class-level weights.

Third, for target samples at the reweighting layer, we assign a soft label to each target feature g_t^j and weight it with its corresponding maximum probability in $\widehat{q_t^j}$. That is to say, the jth target instance has the soft label $\widehat{y_t^j} \in C^s$ and the weight

$$\phi_j = \max\left(\widehat{q_t^j}\right). \tag{9}$$

The target instance-level weights are denoted as $\phi = \{\phi_j\}_{j=1}^{n_t}$. The higher the weight of the target feature, the more likely its soft label is to be true. If each target sample is labeled with one specific class (hard label) and weighted by a constant (i.e., 1), it may raise the problem of false alignment, since the reweighting network may make a mistake predicting some samples due to large domain shift. Especially, when some target samples lie in the overlapping area of two classes of distributions, assigning hard labels to these samples and weighting them by a constant would destroy target data structures.

Although the obtained source class-level weights in (8) and target instance-level weights in (9) can contribute to transferring knowledge from shared source classes and alleviating the negative impact of outlier source classes, these weights highly rely on the probabilities $\widehat{q_t}$. Hence, inspired by [18], [20], and [39], we employ the entropy minimization principle to refine C_y. This principle encourages low-density separation between classes such that C_y can improve itself to better evaluate target unlabeled instances and achieve more accurate probabilities $\widehat{q_t}$ with minimal prediction uncertainty. For each target instance, it can be implemented by minimizing the entropy loss

$$H\left(C_y\left(g_t^j\right)\right) = -\sum_{k=1}^{K} \widehat{q_{t,c_k}^j} \log \widehat{q_{t,c_k}^j}. \tag{10}$$

Thus, plugging (10) into (4), we can have the following loss function to train C_y instead of (4):

$$\mathcal{L}_c(x_s, y_s, x_t) = \frac{1}{n_s}\sum_{i=1}^{n_s} L\left(C_y\left(g_s^i\right), y_s^i\right) + \frac{1}{n_t}\sum_{j=1}^{n_t} H\left(C_y\left(g_t^j\right)\right). \tag{11}$$

C. Dual Alignment Network

With the reweighted source and target features from the reweighting network, source outlier samples would be downweighted and source shared samples would be upweighted. The dual alignment network aims to match joint distributions across domains. The intra domain alignment attempts to minimize the distance between each instance and its corresponding intraclass centroid for source and target domains, respectively. The inter domain alignment includes domain-wise alignment and class-wise alignment.

For intradomain alignment, our goal is to make the learned features in the intermediate domain preserve the intrinsic data structure and the class constraints. That is to say, the features with the same label should be close to the corresponding cluster centroid for both domains. To develop an effective loss term, we first need to determine the centroids for all source and target classes, respectively. Then, for source domain, the loss using the hard labels can be formulated as

$$\mathcal{L}_{\text{intra}}^s = \sum_{k=1}^{K} \frac{1}{n_s^k} \sum_{y_s^i = c_k} \left\| \eta_{c_k} g_s^i - O_s^k \right\|^2 \tag{12}$$

where

$$O_s^k = \frac{1}{n_s^k} \sum_{y_s^i = c_k} \eta_{c_k} g_s^i. \tag{13}$$

Notably, O_s^k is the cluster centroid of source class c_k calculated by mean value, and $1/n_s^k$, as the penalty coefficient, is associated on the distances to balance the effects of different classes. If this coefficient is not involved, the classes overrepresented by enough training instances would play a more important role than that under-represented by only a few. The raised imbalance problem usually results in the degradation of transfer performance on the target domain. Therefore, we attempt to address this problem when the source and target data are imbalanced. For the target domain, the loss using the soft labels can be obtained by

$$\mathcal{L}_{\text{intra}}^t = \sum_{k=1}^{K} \frac{1}{\widehat{n_t^k}} \sum_{\widehat{y_t^i} = c_k} \left\| \phi_i \widehat{g_t^i} - \widehat{O_t^k} \right\|^2 \tag{14}$$

where

$$\widehat{O_t^k} = \frac{1}{\widehat{n_t^k}} \sum_{\widehat{y_t^i} = c_k} \phi_i \widehat{g_t^i}. \tag{15}$$

For the source domain, we aim to minimize the discrepancies between weighted instances and the centroid in the same class for all classes, which can make the classes discriminative and alleviate the effects of source-specific instances. Similar to the source domain, we embed the instance-level weights such that the target instances with high weights would have more important contributions to enhance the intra class compactness and the errors introduced from wrongly predicted target labels can be reduced. The intradomain alignment loss term of both domains can be denoted as

$$\mathcal{L}_{\text{intra}} = \mathcal{L}_{\text{intra}}^{s} + \mathcal{L}_{\text{intra}}^{t}. \tag{16}$$

Clearly, by minimizing $\mathcal{L}_{\text{intra}}$, the instances with the same label would form compact clusters for both the source and target domains.

For inter domain alignment, the domain-wise alignment network maps a d-dimensional representation to a real number with parameter θ_{dw}, that is, h_{dw}: $\mathbb{R}^d \rightarrow \mathbb{R}$. Given $g_s = M(x_s)$ and $g_t = M(x_t)$, the WD with gradient penalty weighted by η between two representation distributions, that is, P_{g_s} and Q_{g_t} can be calculated using (3)

$$\begin{aligned} W_1\left(P_{g_s}, Q_{g_t}\right) = & \sup_{\|h_{dw}\|\leq 1} \mathbb{E}_{x\sim P_{g_s}}[F_s] - \mathbb{E}_{x\sim Q_{g_t}}[F_t] \\ & + \lambda \mathbb{E}_{z_g\sim\mathbb{Z}}\left[\left(\left\|\nabla_{z_g} F_z\right\|_2 - 1\right)^2\right] \end{aligned} \tag{17}$$

where $F_s = h_{dw}(\eta\, g_s)$; $F_t = h_{dw}(\phi g_t)$; $F_z = h_{dw}(z_d)$; source sample x_s^i with label c_k have a corresponding class-level weight η_{c_k}; and z_d are random feature representations sampled along the straight line between pairs of g_s and g_t. If the parameter of the domain-wise alignment network θ_{h_g} is 1-Lipschitz, the WD can be estimated by maximizing the global alignment loss $\mathcal{L}_{dw}$ with parameter θ_{dw} and balancing coefficient λ_{dw}

$$\begin{aligned} \mathcal{L}_{dw}(x_s, x_t) = & \frac{1}{n_s}\sum_{x_s\in D^s} h_{dw}(\eta\, g_s) - \frac{1}{n_t}\sum_{x_t\in D^t} h_{dw}(\phi g_t) \\ & + \lambda_{dw}\left(\left\|\nabla_{z_g} h_{dw}\left(z_g\right)\right\|_2 - 1\right)^2 \end{aligned} \tag{18}$$

where λ is a balancing coefficient.

However, reducing the discrepancy at the domain level cannot guarantee that the same classes from different domains are pulled close together [18]. To address this problem, it would be necessary to do the class-wise alignment. The class-wise alignment network focuses on matching conditional distributions to further explore diverse structures hidden in class characteristics. We have source feature representations with class-level weights and hard labels, as well as target feature representations with instance-level weights and soft labels. If the parameter of the class-wise alignment network θ_{cw} is 1-Lipschitz, the WD can be estimated by maximizing the class-wise alignment loss $\mathcal{L}_{cw}$ with parameter $\theta_{cw} = \{\theta_{cw}^k\}_{k=1}^K$ and balancing coefficients $\{\lambda_{cw}^k\}_{k=1}^K$

$$\begin{aligned} \mathcal{L}_{cw}(x_s, x_t) = \sum_{k=1}^{K} \Bigg\{ & \frac{1}{n_s^k}\sum_{x_s\in D_k^s} h_{cw}^k\left(\eta_{c_k}\, g_s^k\right) - \frac{1}{\widehat{n_t^k}}\sum_{x_t\in \widehat{D_k^t}} h_{cw}^k\left(\widehat{\phi^k}\widehat{g_t^k}\right) \\ & + \lambda_{cw}^k\left(\|\nabla_{z_c^k} h_{cw}^k\left(z_c^k\right)\|_2 - 1\right)^2 \Bigg\} \end{aligned} \tag{19}$$

where n_s^k is the number of source instances with hard label k; D_k^s indicates all the source instances with hard label k; $\widehat{n_t^k}$ is the number of target instances with soft label k; $\widehat{D_k^t}$ represents all the target instances with soft label k; $\widehat{g_t^k}$ includes the target features with soft label k and their corresponding instance-level weights $\widehat{\phi^k}$; z_c^k are random feature representations sampled along the straight line between pairs of g_s^k and $\widehat{g_t^k}$.

Algorithm 1 Dual Alignment for PDA

```
Require: source data $(X_s, Y_s)$, target data $X_t$, $K$ source classes
    $c_k \in C^s, k = 1, \ldots, K$, the minibatch size $m$, training step of
    reweighting network $T$, training step of dual alignment networks $A$,
    balancing coefficients $\alpha = 1$ and $\beta = 1$, learning rate for
    reweighting network and Siamese network $\gamma_1$, learning rate for
    dual alignment networks $\gamma_2$.
Initialize Siamese network, reweighting network, and dual alignment network
    with random parameters $\theta_g$, $\theta_c$, and $\theta_{inter}$
repeat
    Sample minibatch $\{(x_s^i, y_s^i)\}_{i=1}^m$ from $(X_s, Y_s)$
    Sample minibatch $\{x_t^i\}_{i=1}^m$ from $X_t$
    for $t = 1, 2, \ldots, T$ do
        $\theta_c \leftarrow \theta_c - \gamma_1 \nabla_{\theta_c}\mathcal{L}_c$
    end for
    for $a = 1, 2, \ldots, A$ do
        $g_s \leftarrow M(x_s)$, $g_t \leftarrow M(x_t)$, $(\widehat{y_t}, \widehat{q_t}) \leftarrow C(g_t)$
        Sample $z_g$ as the random representations between
    pairs of $g_s$ and $g_t$
        $\theta_{inter} \leftarrow \theta_{inter} + \gamma_2 \nabla_{\theta_{inter}}\mathcal{L}_{inter}$
    end for
    $\theta_g \leftarrow \theta_g - \gamma_1 \nabla_{\theta_g}[\mathcal{L}_c + \alpha\mathcal{L}_{intra} + \beta\mathcal{L}_{inter}]$
until $\theta_c$, $\theta_g$, and $\theta_{inter}$ converge
```

We denote the loss term for the interdomain alignment as $\mathcal{L}_{\text{inter}} = \mathcal{L}_{dw} + \mathcal{L}_{cw}$ with the parameters $\theta_{\text{inter}} = \{\theta_{dw}, \theta_{cw}\}$. The alignment can be achieved by solving the problem

$$\max_{\theta_{\text{inter}}} \mathcal{L}_{\text{inter}} \tag{20}$$

where balancing coefficients λ_{dw} and $\{\lambda_{cw}^k\}_{k=1}^K$ should be set to 0 at the end of each iteration of optimizing the maximum. It is because the gradient penalty ought not to guide other learning procedures.

D. Overall Objective

The overall objective of our proposed DAPDA method is as follows:

$$\min_{\theta_g, \theta_c}\left\{\mathcal{L}_c + \alpha\mathcal{L}_{\text{intra}} + \beta\max_{\theta_{\text{inter}}}\mathcal{L}_{\text{inter}}\right\} \tag{21}$$

where α and β are two balancing coefficients. Algorithm 1 shows the detailed training and testing procedures of our proposed method. The overall objective can be achieved by the standard backpropagation training approach with a two-step iteration. We first train the reweighting network, which can be optimized through minimizing the classification loss with labeled source samples. Second, we apply the trained classifier to predict soft labels of target data such that outlier source samples can be downweighted as well as the

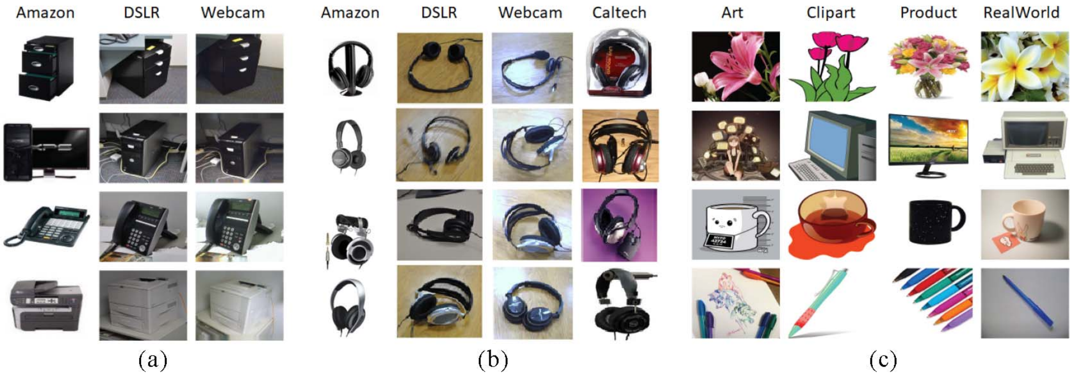


Fig. 2. Illustration of several image examples for (a) Office-31 dataset [40], (b) Office-Caltech dataset [41], and (c) Office-Home dataset [42].

soft labels and instance weights for target samples can be obtained. Then, the inter domain alignment network can be optimized by maximizing the estimators of WDs with gradient penalty via gradient ascent. After this iteration, the balancing coefficients in θ_{inter} are set to 0. The Siamese network is finally updated by the combination of the minimized classification loss, the minimized intradomain alignment loss, and the maximized estimated WDs. The learned feature representations can be domain invariant and class discriminative based on dual alignment.

IV. Experiments

A. Experimental Settings

Datasets: Our proposed DAPDA method is evaluated on three widely used real-world datasets: 1) Office-31 [40]; 2) Office-Caltech [41]; and 3) Office-Home [42]. Fig. 2 shows several image examples for these three datasets.

We first validate DAPDA on the Office-31 dataset, which is a standard benchmark for domain adaptation and composed of 4652 images and 31 classes. Three distinct domains are involved: 1) Amazon (A); 2) Webcam (W); and 3) DSLR (D), which include images downloaded from amazon.com, taken with Web camera, and picked up by digital SLR camera, respectively. We follow the experimental settings of [17] and [19], taking one domain with 31 classes as the source domain and another domain with ten classes (which are shared by Office-31 and Caltech-256) as the target domain to enable adaptation. Hence, six transfer tasks across three domains are conducted: A31 → W10, A31 → D10, D31 → A10, D31 → W10, W31 → A10, and W31 → D10.

Second, we validate DAPDA on the Office-Caltech dataset released by [41], which consists of ten common classes shared by Office-31 and Caltech-256 datasets. The experimental settings of [17] are also applied, taking one domain with ten classes as the source domain and another domain with the first five classes as target domain to enable adaptation. For the partial adaptation of Office-Caltech, we perform 12 tasks across four domains: A10 → C5, A10 → D5, A10 → W5, C10 → A5, C10 → D5, C10 → W5, D10 → A5, D10 → C5, D10 → W5, W10 → A5, W10 → C5, and W10 → D5, in which the numbers of image samples from Amazon (A), Caltech (C), DSLR (D), and Webcam (W) are 958, 1123, 157, and 295, respectively. Thereinto, A, D, and W domains are from Office-31, and C domain comes from Caltech-256. In addition, we further conduct experiments on the Office-Caltech dataset in the standard full protocol.

To evaluate it on a large-scale dataset, we design several transfer tasks on the Office-Home dataset, which contains 15 500 images crawled via a few search engines and online image directories. This dataset has four domains: 1) Artistic (Ar); 2) Clipart (Cl); 3) Product (Pr); and 4) Real-World (Rw) images, where each domain includes images from 65 object classes. In each transfer task, one domain with all 65 classes can be considered as the source domain, and another domain with the first 25 classes can be taken as the target domain. Thus, for the partial adaptation of Office-Home, 12 transfer tasks can be performed: Ar65 → Pr25, Ar65 → Cl25, Ar65 → Rw25, Cl65 →Pr25, Cl65 → Ar25, Cl65 → Rw25, Pr65 → Ar25, Pr65 → Cl25, Pr65 →Rw25, Rw65 →Pr25, Rw65 →Ar25, and Rw65 → Cl25.

Benchmark Methods: For PDA, we compare our proposed DAPDA with the baseline that fine-tuning the CNN (e.g., AlexNet [43] and ResNet-50 [44]), and several deep domain adaptation methods: WDGRL [14], ADDA [31], Reverse Gradient (RevGrad) [28], RTN [30], IWANs [17], SAN [18], domain adversarial neural network (DANN) [45], deep adaptation network (DAN) [13], joint adaptation network (JAN) [46], PADA [19], and ETN [20].

As we all know, in the dual alignment network, the proposed DAPDA with the intra domain alignment only and without the interdomain alignment would perform worse than vice-versa since the distribution gap is not reduced. Moreover, we also would like to explore the effect of the entropy minimization principle for DAPDA. Therefore, to further demonstrate the effectiveness of DAPDA with respect to the inter domain alignment and the entropy minimization principle, three variants are evaluated by ablation study: 1) DAPDA-CW is the variant with the entropy minimization principle and class-wise alignment only, and without domain-wise alignment; 2) DAPDA-DW is the variant with the entropy minimization principle and domain-wise alignment only, and without class-wise alignment; and 3) DAPDA-N-EN is the variant with class-wise and domain-wise alignments, without the entropy minimization principle.

For full-domain adaptation, the compared methods are DANN, DAN, WDGRL, and distribution matching machines

TABLE II
AVERAGE ACCURACY (%) OF PDA TASKS ON THE OFFICE-31 DATASET (ALEXNET)

Methods	A31→D10	A31→W10	D31→A10	D31→W10	W31→A10	W31→D10	AVG
AlexNet	71.97	62.03	68.27	95.25	62.94	97.45	76.32
WDGRL	57.14	47.66	52.06	78.08	73.87	85.66	65.75
ADDA	72.92	70.39	74.46	96.31	76.46	98.72	81.54
RevGrad	57.32	56.59	57.62	75.59	63.15	89.17	66.64
RTN	69.43	68.14	68.27	91.53	77.35	98.09	78.80
IWAN	78.98	76.27	89.46	98.98	81.73	**100.0**	87.57
SAN	81.28	80.02	80.58	98.64	83.09	**100.0**	87.27
DAPDA-CW	88.72	89.73	88.69	98.99	92.13	**100.0**	93.04
DAPDA-DW	85.66	88.37	87.13	97.67	90.93	99.32	91.51
DAPDA-N-EN	85.82	88.84	90.81	99.43	96.32	100.0	93.54
DAPDA	**89.30**	**92.61**	**93.18**	**100.0**	**96.83**	**100.0**	**94.99**

TABLE III
AVERAGE ACCURACY (%) OF PDA TASKS ON THE OFFICE-CALTECH DATASET (ALEXNET)

Methods	A10 →C5	A10 →D5	A10 →W5	C10 →A5	C10 →D5	C10 →W5	D10 →A5	D10 →C5	D10 →W5	W10 →A5	W10 →C5	W10 →D5	AVG
AlexNet	85.27	85.29	76.30	93.58	91.18	83.70	89.51	80.82	98.52	87.37	74.14	**100.0**	87.14
WDGRL	75.53	61.36	56.82	91.30	70.45	76.14	65.34	54.68	84.09	70.63	61.38	90.91	71.55
RevGrad	77.57	80.88	65.93	91.86	83.82	82.22	77.09	69.35	80.74	80.30	72.60	95.59	79.83
RTN	80.99	70.59	69.63	91.86	80.88	93.99	70.02	59.08	91.11	74.73	59.08	**100.0**	78.44
IWAN	89.90	88.24	87.41	94.22	98.53	97.78	94.43	91.61	98.52	95.29	90.24	**100.0**	93.85
PADA	92.05	98.76	87.33	95.25	97.59	96.00	96.39	**95.80**	97.87	96.14	**96.85**	**100.0**	95.70
DAPDA-CW	92.32	92.07	91.22	95.51	95.83	92.52	95.62	91.75	97.92	93.91	92.13	**100.0**	94.23
DAPDA-DW	91.59	95.92	90.31	96.63	91.67	90.86	96.60	92.25	96.05	92.89	91.77	**100.0**	93.88
DAPDA-N-EN	91.80	95.83	91.67	98.48	100.0	97.44	95.51	90.16	98.92	96.13	88.73	100.0	95.39
DAPDA	**93.05**	**98.83**	**93.46**	**98.66**	**100.0**	**97.93**	**97.02**	94.62	**99.10**	**96.62**	94.35	**100.0**	**96.97**

(DMM) [47], our proposed method with the metric of MMD (Ours-MMD) [29], our proposed method with the metric of correlation alignment (Ours-CORAL) [48], in which Ours-MMD indicates the DAPDA with the MMD metric and Ours-CORAL represents the DAPDA with the metric in CORAL (i.e., the second-order statistics).

Implementation Details: Following the standard protocols, we use all labeled source data and unlabeled target data for unsupervised domain adaptation. All our models are implemented using TensorFlow and trained by Adam optimizer. For a fair comparison, we fine-tune the AlexNet and ResNet-50, respectively, which are both pretrained on the ImageNet dataset similar to previous domain adaptation approaches [17], [19]. For DAPDA, we fine-tune the two subnetworks of the Siamese network, which is the standard multiplayer perceptron network designed with two hidden layers of 500 and 100 nodes for all datasets. The reweighting network is built with one hidden layer of 100 nodes, relu activation function, and softmax output function. Our interdomain alignment network is designed with a hidden layer of 100 nodes. The training steps of reweighting network T and dual alignment networks A are 10 and 5, respectively. The learning rates γ_1 and γ_2 are 10^{-4}. The gradients are penalized at source, target, and random representations. The balancing coefficients of gradient penalty λ_{dw} and $\{\lambda_{cw}^k\}_{k=1}^K$ are all set to 10 as suggested in [38].

For each method, the batch size of each domain is set to be 64, and a fixed learning rate is 10^{-4}. We report the average classification accuracy results of each transfer task over three random experiments. The values of hyperparameters are selected based on their original papers.

B. Results

Tables II–V show the average results of the compared and our proposed methods for PDA, where the results of IWAN, SAN, PADA, and ETN are copied directly from their corresponding original papers. The best results are marked in bold. To a large extent, our proposed DAPDA method performs better than previous domain adaptation methods, such as AlexNet, WDGRL, ADDA, RevGrad, and RTN. Furthermore, it can be comparable to some state-of-the-art PDA approaches, such as IWAN, SAN, PADA, and ETN on most datasets.

Table II shows the detailed comparison results of these methods using AlexNet as the baseline on the Office-31 dataset. Our proposed DAPDA method outperforms all the other methods. Specifically, the average classification accuracy of DAPDA is 94.99%, and DAPDA achieves significant performance enhancements of 7.72% and 13.45% compared to the best PDA method IWAN as well as the best full domain adaptation method ADDA, respectively. We note that the PDA methods perform better than the full-domain adaptation methods. The average results of our proposed two variants of DAPDA also are better than most of the compared methods. DAPDA-CW has a slightly better performance than DAPDA-DW since the local characteristics associated with the categories of the Office-31 dataset can greatly contribute to distribution alignment.

TABLE IV
AVERAGE ACCURACY (%) OF PDA TASKS ON THE OFFICE-31 DATASET (RESNET-50)

Methods	A31→D10	A31→W10	D31→A10	D31→W10	W31→A10	W31→D10	AVG
ResNet	65.61	54.52	73.17	94.57	71.71	94.27	75.64
WDGRL	39.03	35.24	40.61	43.29	41.06	37.30	39.42
ADDA	43.66	43.65	42.76	46.48	45.95	40.12	43.77
DANN	41.36	41.35	41.34	46.78	44.68	38.85	42.39
DAN	42.68	46.44	65.66	53.56	65.34	58.60	55.38
RTN	66.88	75.25	85.59	97.12	85.70	98.32	84.81
JAN	35.67	43.39	51.04	53.56	51.57	41.40	46.11
PADA	82.17	86.54	92.69	99.32	95.41	**100.0**	92.69
IWAN	90.45	89.15	95.62	99.32	94.26	99.36	94.69
SAN	90.70	83.39	87.16	99.32	91.85	100.0	92.07
ETN	**95.03**	94.52	**96.21**	100.0	94.64	100.0	**96.73**
DAPDA-N-EN	84.78	90.12	93.65	99.57	97.34	100.0	94.24
DAPDA	92.15	**95.06**	95.13	**100.0**	**97.40**	**100.0**	96.62

TABLE V
AVERAGE ACCURACY (%) OF PDA TASKS ON THE OFFICE-HOME DATASET (RESNET-50)

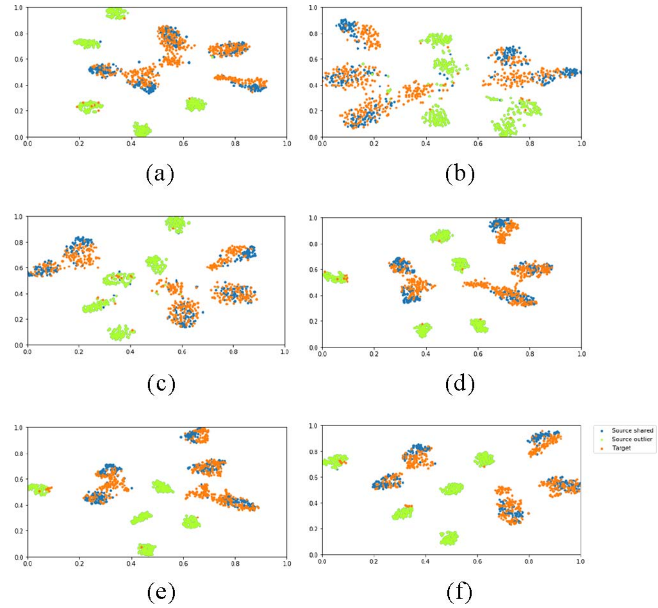
Methods	Ar65	Ar65	Ar65	Cl65	Cl65	Cl65	Pr65	Pr65	Pr65	Rw65	Rw65	Rw65	AVG
	→Pr25	→Cl25	→Rw25	→Pr25	→Ar25	→Rw25	→Ar25	→Cl25	→Rw25	→Pr25	→Ar25	→Cl25	
ResNet	60.78	38.57	75.21	48.12	39.94	52.90	49.68	30.91	70.79	70.42	65.38	41.79	53.71
DANN	54.06	44.89	68.97	34.34	36.27	45.22	44.08	38.03	68.69	46.50	52.98	34.68	47.39
DAN	61.79	44.36	74.49	45.21	41.78	54.11	46.92	38.14	68.42	68.85	64.37	45.37	54.48
RTN	64.33	49.37	76.19	51.74	47.56	57.67	50.38	41.45	75.53	74.78	70.17	51.82	59.25
PADA	67.00	51.59	78.74	53.78	52.16	59.03	52.61	43.22	78.79	77.09	73.73	56.60	62.06
IWAN	54.45	53.94	78.12	47.95	61.31	63.32	54.17	52.02	81.28	82.90	76.46	56.75	63.56
SAN	68.68	44.42	74.60	64.99	**67.49**	**77.80**	59.78	44.72	80.07	78.66	72.18	50.21	65.30
ETN	77.03	**59.24**	79.54	65.73	62.92	75.01	68.29	55.37	84.37	84.54	75.72	57.66	70.45
DAPDA-N-EN	72.49	55.31	77.79	70.68	59.04	68.22	58.93	51.41	80.15	79.28	70.20	59.79	66.94
DAPDA	**77.56**	56.49	**80.29**	**71.52**	65.73	77.28	**66.53**	**55.96**	**85.65**	**84.82**	**77.02**	**60.82**	**71.64**

Similar to Table II, Table III shows the average classification accuracies of several methods using AlexNet as the baseline for 12 partial transfer tasks on the Office-Caltech dataset. DAPDA achieves better performance than most compared methods in 10 out of 12 partial transfer tasks, and it also achieves the second highest accuracy in the remaining two tasks. It can be noted that the performance of the baseline is better than the full-domain adaptation methods, including WDGRL, RevGrad, and RTN. This phenomenon indicates the importance of picking out the source outlier classes such that the errors of misalignment can be alleviated.

In Table IV, we use the ResNet-50 as the baseline on the Office-31 dataset. DAPDA performs slightly worse than ETN but better than the other methods in the average accuracy. No notable degradation is observed compared to state-of-the-art methods. It can be noted that the worst performing method is WDGRL shown in these three tables. That is because WDGRL first aligns the distributions of source and target domains based on the WD criterion and then predicts the labels of target samples, which is not suitable for partial adaptation problem. Starting with alignment without considering the influence of source outliers could easily cause domain confusion and faulty alignment. This also happens to RevGrad based on adversarial nets and RTN based on the MMD metric. WDGRL and RevGrad perform worse than standard AlexNet since negative transfer caused by source outliers is not considered. They try to match source and target domains, including matching the source outliers and target data to predict labels of target samples in outlier classes as much as possible. ADDA, IWAN, SAN, PADA, and ETN are all adversarial nets-based methods, playing minimax game between feature extractors and domain classifiers by the GRL layer. ADDA is an unweighted version of IWAN and thus performs worse than IWAN in detecting source outlier samples. IWAN and PADA focus on selecting out outliers with class-level weights but both of them match different domains without considering latent structures hidden in each class. While SAN uses separate domain classifier for each class to explore the latent structures ignoring intra class compactness. ETN performs well by embedding the source sample weights in the source classifier and the domain-adversarial network.

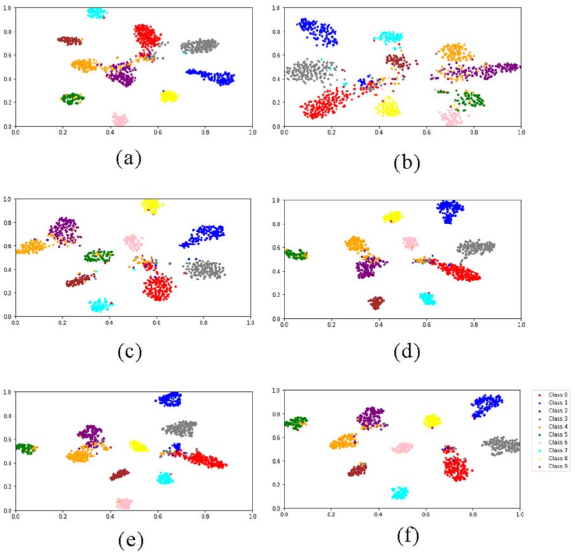
The average classification accuracies of DAPDA and other comparisons on the Office-Home dataset are listed in Table V. It is worth noting that DAPDA obtains the best performance in 9 out of 12 transfer tasks. We observe that DAPDA enhances the average performance by huge margins not only on Office-Caltech and Office-31 datasets both with small domain gaps but also on Office-Home with large domain gaps. The dual alignment strategy makes DAPDA generalize well on the unlabeled target data.

To further illustrate the effectiveness of our proposed DAPDA, as an example, we visualize the learned feature representations using AlexNet as the baseline in A10→C5 task on the Office-Caltech dataset in Table II. In Figs. 3 and 4, the ten classes are labeled as 0–9 and five shared classes are 0–5 classes. In Fig. 3, the blue dots indicate source samples in shared classes and the green dots represent source samples in source outlier classes. The orange dots belong to target



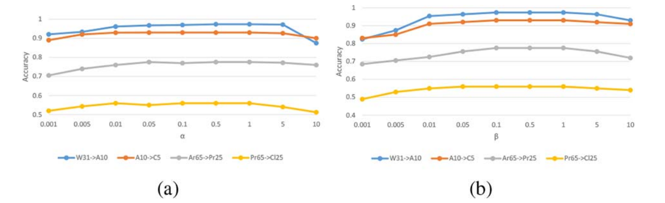
(a) (b) (c) (d) (e) (f)

Fig. 3. t-SNE visualization of learned features of compared and proposed methods for A10→C5 task on the Office-Caltech dataset. The blue, green, and orange dots represent the source shared samples, source outlier samples, and target samples. The orange dots are expected to be aligned with the blue dots. (a) AlexNet. (b) RTN. (c) IWAN. (d) DAPDA-CW. (e) DAPDA-DW. (f) DAPDA.



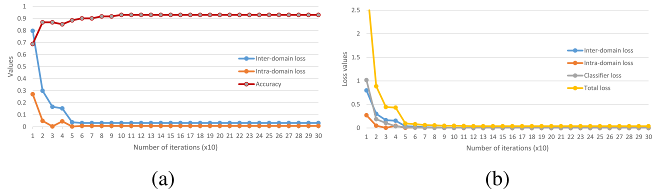
(a) (b) (c) (d) (e) (f)

Fig. 4. t-SNE visualization of learned features of compared and proposed methods for A10→C5 task on the Office-Caltech dataset. Each color represents a class. The dots with the same color are expected to be aligned. (a) AlexNet. (b) RTN. (c) IWAN. (d) DAPDA-CW. (e) DAPDA-DW. (f) DAPDA.



(a) (b)

Fig. 5. Parameter sensitivity on the task W31 → A10, A10 → C5, Ar65 → Pr25, and Pr65 → Cl25. (a) Parameter α. (b) Parameter β.



(a) (b)

Fig. 6. For task A10 → C5. (a) Loss and accuracy values with respect to the number of iterations (x10). (b) Loss values with respect to the number of iterations (x10).

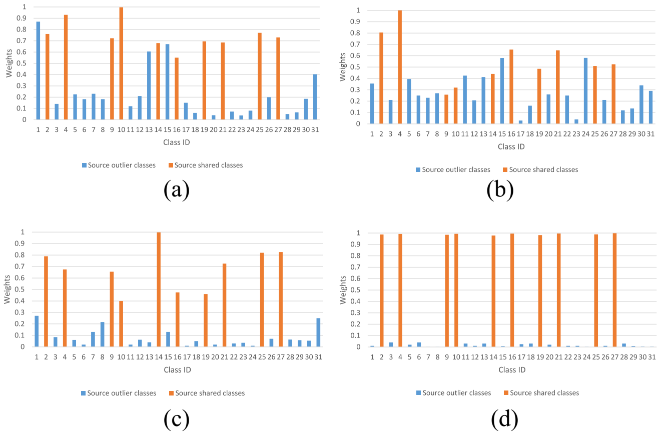
domain. The adaptation is achieved if the orange dots are scarcely aligned with green dots but well aligned with blue dots. In Fig. 4, the dots with the same color belong to the same class for both domains. The alignment is performed effectively if intra class dots have the same color and interclass dots have different colors. To preserve the target domain structure, RTN and IWAN both use target domain entropy minimization strategy to encourage low-density separation among classes. Figs. 3(b) and (c) and 4(b) and (c) show the target samples are not spread to all classes but RTN cannot effectively ensure the intraclass compactness and IWAN misses more source outlier samples. The variants of our proposed methods only consider one style of alignment, that is, DAPDA-CW focuses on class-wise alignment shown in Figs. 3(d) and 4(d) and DAPDA-DW focuses on the domain-wise alignment shown in Figs. 3(e) and 4(e). Figs. 3(f) and 4(f) verify that our proposed DAPDA method selects out most source outlier samples and simultaneously ensures the interclass separation as well as intra class compactness.

For parameter sensitivity, there are two tunable parameters: 1) α and 2) β, where α controls the balance between the intra domain loss and the other losses, and β controls the balance between the inter domain loss and the other losses. We have conducted parameter sensitivity analysis on four transfer tasks: 1) W31 → A10; 2) A10 → C5; 3) Ar65 → Pr25; and 4) Pr65 → Cl25. From Fig. 5, it can be seen that DAPDA could achieve good performance under a range of parameter values. First, we run DAPDA as α varies from 0.001 to 10 when $\beta = 1$. From Fig. 5(a), we can observe that the small α values would contribute to improving the accuracy and the too large α values would degrade the performance. That is because it will weaken the effects of interdomain loss term. Next, from Fig. 5(b), we evaluate DAPDA by varying β from 0.001 to 10 with α fixed to 1. It can be observed that small β values would result in poor performance but reasonable β values would enhance the accuracy. Note that DAPDA is more sensitive to β than α, and this guides us to determine $\alpha \in [0.01\ 5]$ and $\beta \in [0.1\ 5]$. In the experiments, we empirically set $\alpha = \beta = 1$.

For convergence analysis, we give an example on the transfer task A10 → C5. Fig. 6(a) shows the changing trends of interdomain loss, intra domain loss, and the accuracy of soft labels with respect to the number of iterations. Our proposed DAPDA method converges fast on the test target data. We also plot the changing trend of the classifier loss, interdomain loss, intradomain loss, and the total loss with respect to the number

TABLE VI
AVERAGE ACCURACY (%) OF FULL-DOMAIN ADAPTATION TASKS ON THE OFFICE-CALTECH DATASET (ALEXNET)

Methods	A→C	A→D	A→W	C→A	C→D	C→W	D→A	D→C	D→W	W→A	W→C	W→D	AVG
DANN	87.81	91.20	81.11	92.43	91.22	89.50	87.93	82.91	98.90	82.31	85.64	100.0	89.25
DAN	84.06	91.71	91.82	92.00	89.27	90.57	89.98	80.27	98.51	92.08	81.19	100.0	90.67
DMM	83.30	93.00	92.20	92.60	91.70	90.50	93.20	84.30	99.70	92.50	85.80	100.0	92.70
WDGRL	86.99	93.68	89.47	93.54	94.74	91.58	91.69	**90.24**	97.89	93.67	89.43	100.0	92.74
Ours-MMD	90.16	96.53	90.24	93.72	93.63	90.51	93.88	88.10	97.60	93.55	87.92	100.0	92.99
Ours-CORAL	86.35	91.19	91.05	92.89	90.04	92.73	86.03	85.73	97.88	89.06	88.70	100.0	90.97
DAPDA-N-EN	90.50	98.23	98.91	94.34	94.71	94.71	94.30	90.16	99.02	96.63	89.54	100.0	95.08
DAPDA	**92.27**	**98.19**	**99.25**	**94.42**	**94.83**	**94.76**	**95.00**	90.33	**99.10**	**96.40**	**90.24**	**100.0**	**95.40**



(a) (b)

(c) (d)

Fig. 7. For the task A31 → W10, histograms of class weights learned by (a) ResNet-50, (b) DANN, (c) PADA, and (d) DAPDA.

of iterations in Fig. 6(b). As we can see, DAPDA converges to the lowest test error within 80 iterations.

To demonstrate if the trained classifier C_y can correctly reweight source classes according to whether they are in the shared label space, we give an example in Fig. 7 of this response. It shows the histograms of class weights learned using ResNet-50, DANN, PADA, and DAPDA on the task A31 → W10. The blue bins indicate source outlier classes, and the orange bins represent source shared classes in the shared label space. From Fig. 7(a) and (b), note that ResNet-50 and DANN can hardly select out outlier classes since there is not a sharp distinction between weights for outlier classes and that for shared classes. As shown in Fig. 7(c), we know that PADA can better distinguish the source outlier and shared classes. However, some weights for the shared classes are still below 0.5, such as Classes 10, 16, and 19. In the meanwhile, some outlier weights are higher than expected, such as Classes 1, 7, 8, and 31. To compare these methods, Fig. 7(d) shows the mean value of learned weights, that is, $\{(1/n_t)\sum_{j=1}^{n_t}\widehat{q^j_{t,c_k}}\}_{k=1}^{K}$, which are the source class-level weights before normalization in DAPDA. We can note that the weights for the shared classes are almost up to 1 and the weights for the outlier classes are close to 0. The low weights for source outlier classes can greatly alleviate the raised negative transfer.

To evaluate the influence of the number of shared classes on performance, we also conduct the experiments to compare classification accuracies by varying the number of target classes. Fig. 8 shows that our proposed three methods outperform the AlexNet and IWAN to a large extent on A31→W10 transfer task using AlexNet as the baseline. As the number of target classes get smaller, the performances have relatively better improvements. Our proposed DAPDA method shows an "even" best performance.

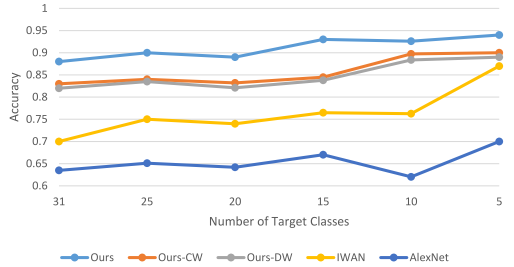
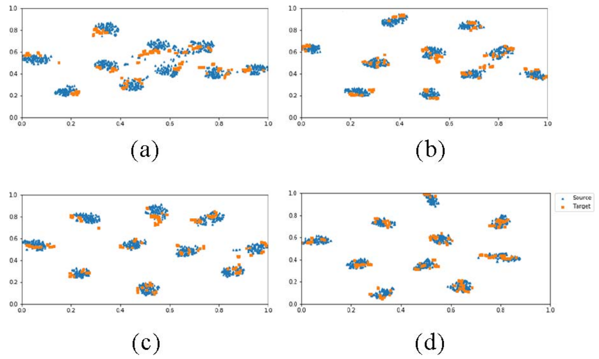


Fig. 8. Accuracy curve of varying the number of target classes for A31→W10 task using AlexNet as the baseline in Table II.

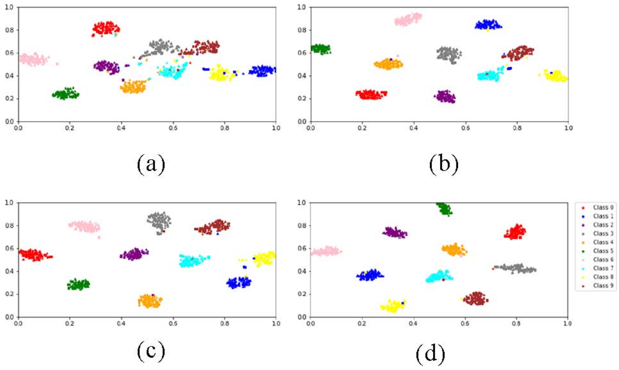


(a) (b)

(c) (d)

Fig. 9. t-SNE visualization of learned features of compared and proposed methods for A→C task on the Office-Caltech dataset. The blue and orange dots represent source and target samples, respectively. The orange dots are expected to be aligned with the blue dots. (a) DANN. (b) Ours-MMD. (c) Ours-CORAL. (d) DAPDA.

To verify our proposed DAPDA on the full-domain adaptation setting, we further conduct experiments on the Office-Caltech dataset. The average classification accuracies are shown in Table VI, which manifests that DAPDA still has superiority in the standard full protocol. That is because we design DAPDA to not only align the source and target domains as much as possible but also explore the class-invariant information in the shared label space of both domains. The results also show that DAPDA outperforms the Ours-MMD and Ours-CORAL such that the effectiveness of WD can be demonstrated. Moreover, we visualize the learned feature representations of several methods for A→C transfer task in Table VI. From Figs. 9 and 10, it can be observed that DAPDA



(a) (b)

(c) (d)

Fig. 10. t-SNE visualization of learned features of compared and proposed methods for A→C task on the Office-Caltech dataset. Each color represents a class. The dots with the same color are expected to be aligned. (a) DANN. (b) Ours-MMD. (c) Ours-CORAL. (d) DAPDA.

better aligns source and target domains than other compared methods.

V. Conclusion

In this article, we proposed a novel DAPDA. DAPDA can effectively select out source outlier samples and learn domain-invariant feature representations to explore latent structures under different data distributions across domains. The data distributions can be aligned as domain-wise and class-wise manners. The domain discrepancy can be effectively reduced using the metric of Wasserstein distance with the gradient penalty. The experimental results on some datasets demonstrate that DAPDA outperforms several state-of-the-art PDA methods. From feature visualization results, the great learning capability of DAPDA is manifested in capturing domain-invariant and target-discriminative representations. From the accuracy curve result, DAPDA shows even high performance. For future work, we will integrate DAPDA into existing domain adaptation frameworks, and investigate architectures for tasks in more complex scenarios.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Transfer independently together: A generalized framework for domain adaptation," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2144–2155, Jun. 2019.

[3] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 137–144.

[4] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, 2010.

[5] M. Uzair and A. Mian, "Blind domain adaptation with augmented extreme learning machine features," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 651–660, Mar. 2016.

[6] L. Li, H. He, J. Li, and G. Yang, "Adversarial domain adaptation via category transfer," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2019, pp. 1–8.

[7] C.-X. Ren, X.-L. Xu, and H. Yan, "Generalized conditional domain adaptation: A causal perspective with low-rank translators," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 821–834, Feb. 2020.

[8] S. Khalighi, B. Ribeiro, and U. J. Nunes, "Importance weighted import vector machine for unsupervised domain adaptation," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3280–3292, Oct. 2017.

[9] K. Yan, L. Kou, and D. Zhang, "Learning domain-invariant subspace using domain features and independence maximization," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 288–299, Jan. 2018.

[10] Z. Ding and Y. Fu, "Deep transfer low-rank coding for cross-domain learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1768–1779, Jun. 2019.

[11] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2019.

[12] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, "Locality preserving joint transfer for domain adaptation," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6103–6115, Dec. 2019.

[13] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[14] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–10.

[15] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 4068–4076.

[16] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 801–814, Apr. 2019.

[17] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8156–8164.

[18] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2724–2732.

[19] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 135–150.

[20] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2985–2994.

[21] L. Li, H. He, J. Li, and W. Li, "EDOS: Entropy difference-based oversampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2018, pp. 1–8.

[22] Z. Wan and H. He, "AnswerNet: Learning to answer questions," *IEEE Trans. Big Data*, vol. 5, no. 4, pp. 540–549, Dec. 2019.

[23] Z. Wan, H. He, and B. Tang, "A generative model for sparse hyperparameter determination," *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 2–10, Mar. 2018.

[24] S. Li, L. Li, J. Yan, and H. He, "SDE: A novel clustering framework based on sparsity-density entropy," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1575–1587, Aug. 2018.

[25] L. Li, H. He, and J. Li, "Entropy-based sampling approaches for multi-class imbalanced problems," *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 30, 2019, doi: 10.1109/TKDE.2019.2913859.

[26] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.

[27] Z. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, and H. He, "Robot-assisted pedestrian regulation based on deep reinforcement learning," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1669–1682, Apr. 2020.

[28] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[29] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.

[30] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[31] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.

[32] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 3730–3739.

[33] B. Bhushan Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty, "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 447–463.

[34] H. Narayanan and S. Mitter, "Sample complexity of testing the manifold hypothesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1786–1794.

[35] I. Redko, A. Habrard, and M. Sebban, "Theoretical analysis of domain adaptation with optimal transport," in *Machine Learning and Knowledge Discovery in Databases*. Cham, Switzerland: Springer, 2017, pp. 737–753.

[36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017. [Online]. Available: https://arxiv.org/abs/1701.07875

[37] C. Villani, *Optimal Transport: Old and New*, vol. 338. Heidelberg, Germany: Springer-Verlag, 2008.

[38] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[39] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 529–536.

[40] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

[41] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2066–2073.

[42] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5018–5027.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[45] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," 2014. [Online]. Available: https://arxiv.org/abs/1412.4446

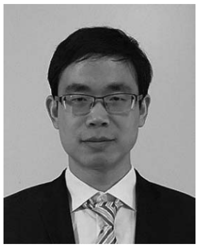
[46] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[47] Y. Cao, M. Long, and J. Wang, "Unsupervised domain adaptation with distribution matching machines," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2795–2802.

[48] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.

Lusi Li received the B.S. and M.S. degrees in computer science from the Zhongnan University of Economics and Law, Wuhan, China, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree in electrical engineering with the University of Rhode Island, Kingston, RI, USA.

Her research interests include machine learning, data mining, and transfer learning.



Zhiqiang Wan (Student Member, IEEE) received the B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2012, and the M.S. degree from the School of Electrical and Electronics Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Electrical, Computer and Biomedical Engineering, University of Rhode Island (URI), Kingston, RI, USA.

His current research interests include deep learning, deep reinforcement learning, and cyber-physical systems, with a particular interest in smart grid applications.

Mr. Wan was a recipient of the URI Graduate Student Research and Scholarship Excellence Award in the Life Sciences, Physical Sciences, and Engineering in 2019; the Best Paper Award in the IEEE Power & Energy Society General Meeting in 2018; and the Best Paper Award in the IEEE 11th International Conference on Power Electronics and Drive Systems in 2015.

Haibo He (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University, Athens, OH, USA, in 2006.

He is currently the Robert Haas Endowed Chair Professor with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA. His current research interests include computational intelligence, machine learning, data mining, and various applications.

Dr. He received the IEEE International Conference on Communications Best Paper Award in 2014, the IEEE CIS Outstanding Early Career Award in 2014, and the National Science Foundation CAREER Award in 2011. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.