Bridging the Theoretical Bound and Deep Algorithms for Open Set Domain Adaptation

Li Zhong, Zhen Fang, *Member, IEEE*, Feng Liu, *Member, IEEE*, Bo Yuan, *Senior Member, IEEE*, Guangquan Zhang, and Jie Lu, *Fellow, IEEE*

***Abstract*— In the unsupervised open set domain adaptation (UOSDA), the target domain contains unknown classes that are not observed in the source domain. Researchers in this area aim to train a classifier to accurately: 1) recognize *unknown target data* (data with unknown classes) and 2) classify other target data. To achieve this aim, a previous study has proven an upper bound of the target-domain risk, and the *open set difference*, as an important term in the upper bound, is used to measure the risk on unknown target data. By minimizing the upper bound, a *shallow* classifier can be trained to achieve the aim. However, if the classifier is very flexible [e.g., deep neural networks (DNNs)], the open set difference will converge to a negative value when minimizing the upper bound, which causes an issue where most target data are recognized as unknown data. To address this issue, we propose a new upper bound of target-domain risk for UOSDA, which includes four terms: source-domain risk, ϵ-*open set difference* (Δ_ϵ), distributional discrepancy between domains, and a constant. Compared with the open set difference, Δ_ϵ is more robust against the issue when it is being minimized, and thus we are able to use very flexible classifiers (i.e., DNNs). Then, we propose a new principle-guided *deep* UOSDA method that trains DNNs via minimizing the new upper bound. Specifically, source-domain risk and Δ_ϵ are minimized by gradient descent, and the distributional discrepancy is minimized via a novel open set conditional adversarial training strategy. Finally, compared with the existing shallow and deep UOSDA methods, our method shows the state-of-the-art performance on several benchmark datasets, including digit recognition [*modified National Institute of Standards and Technology database* (*MNIST*), *the Street View House Number* (*SVHN*), *U.S. Postal Service* (*USPS*)], object recognition (*Office-31*, *Office-Home*), and face recognition [*pose, illumination, and expression* (*PIE*)].**

***Index Terms*— Domain adaptation (DA), machine learning, open set recognition, transfer learning.**

Manuscript received April 30, 2020; revised December 28, 2020; accepted October 1, 2021. This work was supported by the Australian Research Council (ARC) under Grant FL190100149 and Grant DP170101632. The work of Li Zhong was supported by the Australian Artificial Intelligence Institute, University of Technology Sydney (UTS-AAII). *(Li Zhong, Zhen Fang, and Feng Liu contributed equally to this work.) (Corresponding author: Jie Lu.)*

Li Zhong is with the Faulty of Engineering and Information Technology, Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW 2007, Australia, and also with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China (e-mail: allenzhong95@gmail.com).

Zhen Fang, Feng Liu, Guangquan Zhang, and Jie Lu are with the Faulty of Engineering and Information Technology, Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: zhen.fang@uts.edu.au; feng.liu@uts.edu.au; guangquan.zhang@uts.edu.au; jie.lu@uts.edu.au).

Bo Yuan is with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China (e-mail: boyuan@ieee.org).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3119965.

Digital Object Identifier 10.1109/TNNLS.2021.3119965

I. Introduction

DOMAIN adaptation (DA) methods aim to train a target-domain classifier with data in source and target domains [1]. Based on the variety of data in the target domain (i.e., fully labeled, partially labeled, and unlabeled), DA consists of three categories: supervised DA [2]–[4], semi-supervised DA [5]–[7], and unsupervised DA (UDA) [8]–[10]. In practice, UDA methods have been deployed to solve diverse real-world problems, such as object recognition [11]–[13], cross-domain recommendation [14], [15], and sentiment analysis [16], [17].

There are two common settings in UDA: unsupervised closed set DA (UCSDA) and unsupervised open set DA (UOSDA). UCSDA is a classical scenario in which the source and target domains share the same label sets. In contrast, in UOSDA, the target domain contains some unknown classes that are not observed in the source domain, and the data with unknown classes are called *unknown target data*. In Fig. 1, the source domain contains four known classes (i.e., monitor, mug, staple, and calculator), but the target domain contains some unknown classes in addition to the classes in the source domain.

UOSDA is more general than UCSDA, since the label sets are usually not consistent between the source and target domains in a real-world scenario. Namely, the target domain may contain classes that are not observed in the source domain. For example, a classifier trained with images of various kinds of cats is likely to encounter the image of a dog or another animal in reality. In this case, the UCSDA methods are unable to distinguish the unseen animals (i.e., unknown classes). The UOSDA methods, however, can establish a boundary between known classes and unknown classes.

Panareda Busto and Gall [18] are the first to propose the setting of UOSDA, but the source domain also contains some unknown classes in Panareda Busto and Gall's article. Since it is expensive and prohibitive to obtain data labeled by unknown classes in the source domain, Saito *et al.* [19] propose a new UOSDA setting where the source domain only contains known classes. In this article, we focus on the same setting as Saito *et al.*'s article, which is more realistic [19], [20].

In UOSDA, we aim to train a target-domain classifier with labeled data in the source domain and unlabeled data in the target domain. The trained classifier is expected to accurately 1) recognize unknown target data and 2) classify other target data. The existing UOSDA methods can be divided into two groups: shallow methods and deep methods.

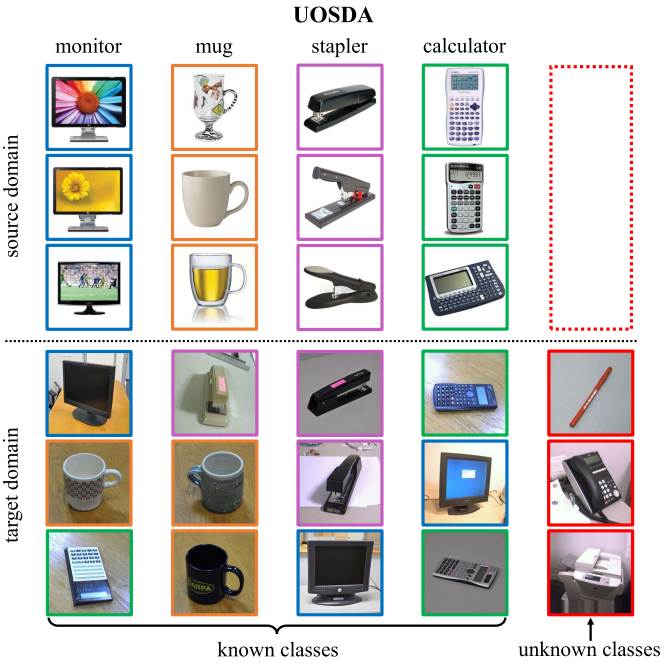

Fig. 1. UOSDA. When the target domain does not contain unknown classes, UOSDA will degenerate into the UCSDA.

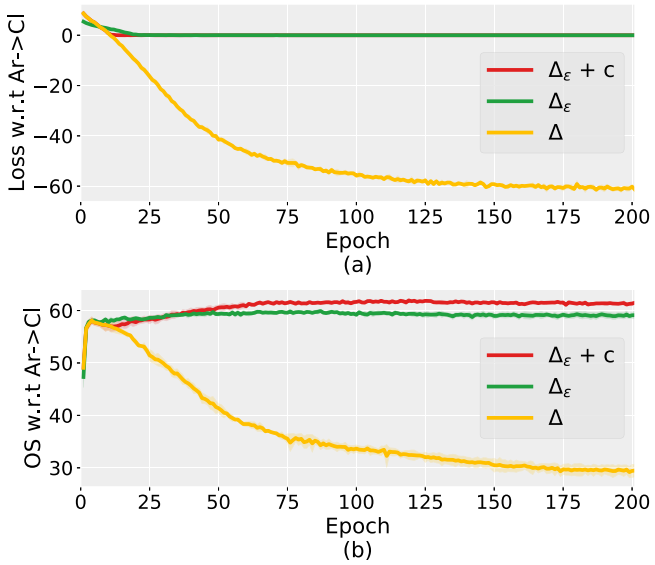

Fig. 2. Accuracy of OS and loss w.r.t. the task Ar → Cl. "c" denotes the conditional adversarial training strategy. Δ_ϵ is the ϵ-open set difference proposed in this article and Δ is the open set difference proposed in [20]. The loss in (a) is the value of Δ or Δ_ϵ. It is worth noting that the green line and the red line in (b) are partially coincident. Here, ϵ is set as 0.

For shallow methods, a recent work [20] proved an upper bound of target-domain risk, which can provide a theoretical guarantee for the design of a shallow UOSDA method. For deep methods, since [21]–[24] have shown that deep neural networks (DNNs) can learn more transferable features, researchers presented DNN-based methods to address the UOSDA problem [19], [25], [26]. Nevertheless, these deep UOSDA methods lack theoretical guarantees. Thus, how to bridge theoretical bound and deep algorithms is both necessary and important for addressing the UOSDA problem.

To train an effective target-domain classifier, Fang *et al.* [20] have proven an upper bound of target-domain risk [see (14)] for the UOSDA problem and propose a *shallow* UOSDA method. Specifically, the bound consists of four terms: source-domain risk, distributional discrepancy between domains, *open set difference* (Δ), and a constant. Open set difference, as an important term in upper bound, is leveraged to measure the risk of a classifier on unknown target data. The shallow method in [20] trains a target-domain classifier by minimizing the empirical estimation of the upper bound.

However, the theoretical bound presented in [20] is not adaptable to flexible classifiers (i.e., DNNs). In Fig. 2, we show that if the classifier is a DNN, the accuracy [OS in Fig. 2(b)] in the target domain will drop significantly [yellow line in Fig. 2(b)] when minimizing the empirical estimates of the upper bound. This phenomenon confirms that we cannot simply combine the existing theoretical bound and deep algorithms to address the UOSDA problem.

To reveal the nature of this phenomenon, we investigate that the lower bound of the distributional discrepancy is the negative value of open set difference. Since DNNs are very flexible and the empirical open set difference can be a negative value, empirical open set difference will be quickly minimized to a very negative value [yellow line in Fig. 2(a)]. Based on the lower bound of the distributional discrepancy, if the empirical open set difference is a very small negative number, the distributional discrepancy is greater than a very large positive number. Consequently, we fail to align the distributions of the two domains, resulting in a very low accuracy on the target domain [yellow line in Fig. 2(b)].

In this article, we propose a new upper bound of target-domain risk for UOSDA [see (20)], which includes four terms: source-domain risk, *ϵ-open set difference* (Δ_ϵ), conditional distributional discrepancy between domains, and a constant. ϵ is the lower bound of open set difference and we construct a new risk estimator Δ_ϵ that limits the descent of the open set difference by ϵ. Δ_ϵ can ensure prompt prevention of the lower bound of the distributional discrepancy between two domains from significantly increasing. Fig. 2 shows that minimizing the empirical estimates of the new upper bound achieves higher accuracy [green line in Fig. 2(b)].

Then, we propose a new principle-guided *deep* UOSDA method that trains DNNs via minimizing empirical estimates of the new upper bound. The network structure is shown in Fig. 3. We use a generator ($\boldsymbol{G}$) to extract the feature of input data, a classifier ($\boldsymbol{C}$) to classify input data, and a domain discriminator ($\boldsymbol{D}$) to assist distribution alignment. The overall object function consists of source classification loss, binary adversarial loss, domain adversarial loss, and empirical Δ_ϵ. Specifically, the source classification loss and empirical Δ_ϵ are minimized by gradient descent, and a gradient reverse layer is adopted for adversarial losses.

To effectively align distributions between data with known classes, we propose a novel open set conditional adversarial training strategy based on the tensor product between feature representation and label prediction to capture the multimodal structure of distribution. According to [27], [28], it is significant to capture the multimodal

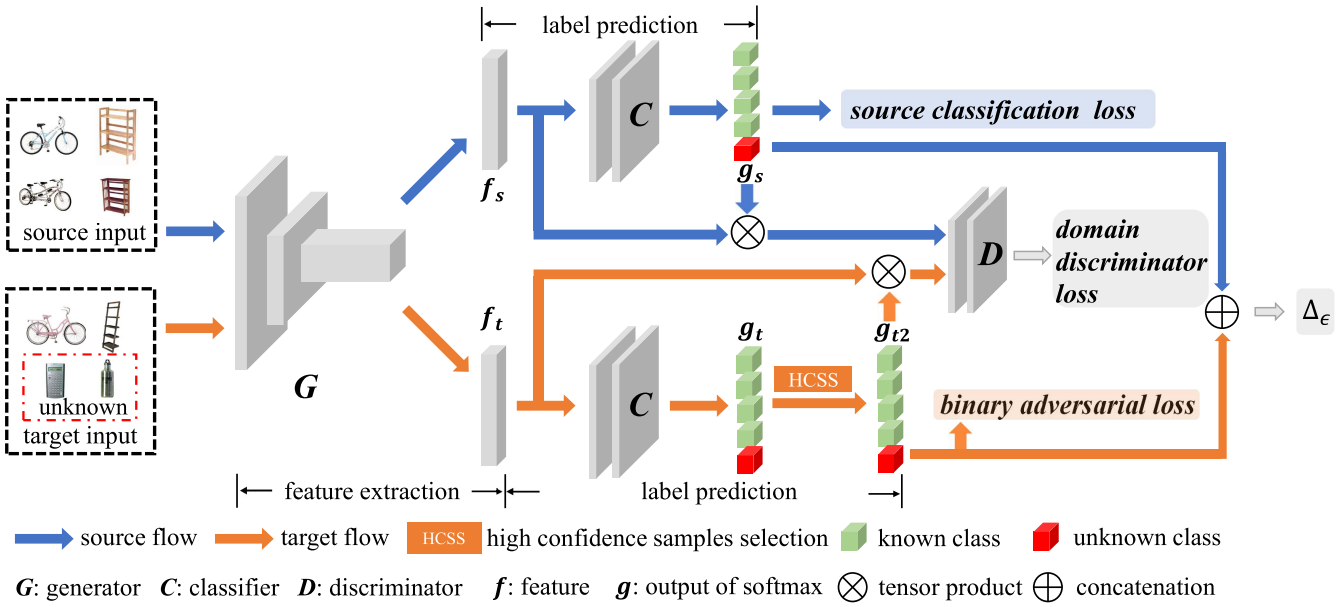

Fig. 3. Framework of the proposed method. The generator ($\boldsymbol{G}$) aims to extract the feature (f) of the input data and feed it to the classifier ($\boldsymbol{C}$) to predict its label ($\hat{y}$). This whole framework consists of three parts. 1) BADA, which is made of source classification loss and binary adversarial loss. The classifier can find a rough boundary between known data and unknown data. 2) ϵ-open set difference (Δ_ϵ). We proposed the amended risk estimator to more properly estimate the risk of the classifier on unknown data. and 3) CDAN, which aims to capture multimodal structure of distribution for distribution alignment. In summary, our method can achieve better performance by accurately estimating risk on unknown target data and aligning distribution more adequately.

structures of distributions using cross-covariance dependency between the features and classes. However, the existing deep UOSDA methods align distributions by either the binary adversarial net [19], [25] or the multibinary classifier [26], which is not adequate for distributions with multimodal structure. Furthermore, this novel training strategy also pushes unknown target data away from data with known classes via $\boldsymbol{D}$. As shown in Fig. 2(b), the novel distribution alignment strategy can further boost the performance of the classifier.

To validate the efficacy of the proposed method, we conduct extensive experiments on several standard benchmark datasets containing 41 transfer tasks. Compared with the existing shallow and deep UOSDA methods, our method shows state-of-the-art performance on digit recognition [*modified National Institute of Standards and Technology database* (*MNIST*), *the Street View House Number* (*SVHN*), *U.S. Postal Service* (*USPS*)], object recognition (*Office-31*, *Office-Home*), and face recognition [*pose, illumination, and expression* (*PIE*)]. The main contributions of this article are:

1) A new theoretical bound of target-domain risk for UOSDA is proposed. It is essential since the existing bound does not apply to flexible classifiers (i.e., DNNs). Thus, this work can bridge the gap between the existing theoretical bound and deep algorithms for the UOSDA problem.
2) A UOSDA method based on DNNs is proposed under the guidance of the proposed theoretical bound. The method can better estimate the risk of the classifier on unknown data than the existing deep methods with the theoretical guarantee.
3) A novel open set conditional adversarial training strategy is proposed to ensure that our method can align the distributions of two domains better than the existing UOSDA methods.
4) Experiments on Digits, Office-31, Office-Home, and PIE show that the accuracy of the OS of our method significantly outperforms all baselines, which shows that our method achieves state-of-the-art performance.

This article is organized as follows. Section II reviews the works related to UCSDA, open set recognition, and UOSDA. Section III introduces the definitions of notations and our problem. Section IV demonstrates the motivation of this article. Theoretical results and the proposed method are shown in Section V. The experimental results and analyses are provided in Section VI. Finally, Section VII concludes this article.

II. Related Work

UOSDA is a combination of UCSDA and open set recognition. In this section, we present a systematic review of related studies.

A. Closed Set Domain Adaption

In [29], a theoretical bound for UCSDA is given, which indicates that minimizing the source risk and distributional discrepancy is the key to the UCSDA problem. Based on this point, there are two kinds of methods for UCSDA: one is to use a distributional discrepancy measurer to measure the domain gap [30]; the other is the adversarial training strategy [28].

Transfer component analysis (TCA) [30] uses maximum mean discrepancy (MMD) [31], [32] to learn a domain invariant feature by aligning marginal distribution. Meanwhile, joint distribution adaptation (JDA) [33] aligns marginal distribution and conditional distribution simultaneously. To simplify the training of a classifier, easy transfer learning (EasyTL) [34]

exploits the intradomain information to get a nonparametric feature and the classifier. CORrelation alignment (CORAL) [35] aligns the second-order statistics of the source and target domains to minimize domain divergence. Manifold embedded distribution alignment (MEDA) [36] performs a dynamic distribution alignment in a Grassmann manifold subspace.

Meanwhile, DNNs have also been introduced into DA and achieved competitive performance in UCSDA. Deep adaptation networks (DANs) [37] use the multikernel MMD (MK-MMD) to align the feature of 6–8 layers in Alexnet. Deep CORAL Correlation is the extension of shallow method CORAL in DNNs. Wasserstein distance guided representation learning (WDGRL) [38] uses the Wasserstein distance to learn an invariant representation in DNNs.

Representative adversarial-training-based methods are domain-adversarial training of neural networks (DANNs) [39] and conditional adversarial domain adaptation (CDAN) [28]. DANN uses a domain discriminator to recognize which domain data come from and deceive the domain discriminator by changing features so that an invariant representation can be learned during the adversarial procession. Furthermore, CDAN uses the tensor product between the feature and classifier prediction to grasp the multimodal information and an entropy condition to control the uncertainty of the classifier. However, these methods can only cope with the UCSDA problem and are unable to address the UOSDA problem.

B. Open Set Recognition

This setting allows some unknown classes to be shown in the target domain, but there is no distributional discrepancy between domains [40]. Open set support vector machine (SVM) [41] rejects the unknown classes via a fixed threshold. Open set nearest neighbor (OSNN) [42] extends the nearest neighbor to recognize unknown classes. Bendale and Boult [43] introduce a layer named OpenMAX to estimate the probability that the input data are recognized as unknown classes in DNNs. However, these methods do not consider distributional discrepancy. They are also unable to address the UOSDA problem.

C. Open Set DA

Panareda Busto and Gall [18] were the first to propose the setting of UOSDA. They used a method named assign-and-transform-iterately (ATI) to assign labels to target data using a distance matrix between the target data and source class centers and aligned distributions through a mapping matrix. In the setting of this article, however, the source domain contains some unknown classes to assist the classifier to recognize unknown data. Since obtaining unknown samples of the source domain is expensive and time-consuming, open set backpropagation (OSBP) [19] assumes a more realistic scenario that the source domain has no unknown classes, which is more challenging. An adversarial network is used to recognize unknown samples and align distribution during backpropagation.

Based on OSBP, Feng *et al.* [25] proposed a method named SCI_SCM, which uses semantic structure among data to align the distribution of known classes and push unknown classes away from known classes. Separate to adapt (STA) [26] uses a coarse-to-fine weight mechanism to separate unknown samples from the target domain. In distribution alignment with open difference (DAOD) [20], a theoretical bound is proposed for UOSDA and a risk estimator is used to recognize unknown target data.

However, the existing deep UOSDA methods lack the theoretical guidance and the upper bound in [20] is not applicable to DNNs, which causes a large distributional discrepancy (details are shown in Section IV). Obviously, for UOSDA, there is a gap between the existing theoretical bound and deep algorithms. In this article, we aim to fill this gap.

III. Preliminary and Notations

The definitions of the UOSDA problem and some important concepts are introduced in this section. The notations used in this article are summarized in Table I.

A. Definitions and Problem Setting

Important definitions are presented as follows.

Definition 1 (Domain [20]): Given a feature space $\mathcal{X} \subset \mathbb{R}^d$ and a label space $\mathcal{Y}$, a *domain* is a joint distribution $P(X, Y)$, where the random variables $X \in \mathcal{X}$, $Y \in \mathcal{Y}$.

In Definition 1, $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ mean that the spaces $\mathcal{X}$ and $\mathcal{Y}$ contain the image sets of X and Y, respectively. In the article, we name the random variable X as feature vector and the random variable Y as label. Based on this definition, we have:

Definition 2 (Domains for Open Set DA [20]): Given a feature space $\mathcal{X} \subset \mathbb{R}^d$ and the label spaces $\mathcal{Y}^s, \mathcal{Y}^t$, the source and target domains have different joint distributions $P(X^s, Y^s)$ and $P(X^t, Y^t)$, where the random variables $X^s, X^t \in \mathcal{X}$, $Y^s \in \mathcal{Y}^s$, $Y^t \in \mathcal{Y}^t$, and the label space $\mathcal{Y}^s \subset \mathcal{Y}^t$.

From the definitions above, we can note that: 1) This article focuses on homogeneous situations. Thus, X^s and X^t belong to the same space and 2) $\mathcal{Y}^t$ contains $\mathcal{Y}^s$. It is the *unknown target classes* that are the classes from $\mathcal{Y}^t \backslash \mathcal{Y}^s$. It is the *known classes* that are the classes from $\mathcal{Y}^s$. Thus, the UOSDA problem is

Problem 1 (UOSDA [20]): Given labeled samples $\mathcal{S}$ drawn from the joint distribution of the source domain $P(X^s, Y^s)$ i.i.d and unlabeled samples $\mathcal{T}_X$ drawn from the marginal distribution of the target domain $P(X^t)$ i.i.d. The aim of UOSDA is to find a target classifier $\boldsymbol{c}^t : \mathcal{X} \to \mathcal{Y}^t$ such that

1) $\boldsymbol{c}^t$ classifies the known target samples into the correct known classes;
2) $\boldsymbol{c}^t$ recognizes the unknown target samples as unknown.

According to the definition of the problem, the target-domain classifier only needs to recognize unknown target data as unknown and classify other target data. It is not necessary to classify unknown target data, and all unknown target data are recognized as the "unknown class." In general, we assume that $\mathcal{Y}^s = \{\mathbf{y}_k\}_{k=1}^K$, $\mathcal{Y}^t = \{\mathbf{y}_k\}_{k=1}^{K+1}$, where the label $\mathbf{y}_{K+1}$ denotes the unknown class and the label $\mathbf{y}_k \in \mathbb{R}^{(K+1)\times 1}$ is a one-hot vector. The label $\mathbf{y}_k$ denotes the kth class.

TABLE I
NOTATIONS AND THEIR DESCRIPTIONS

Notation	Description	Notation	Description
$\mathcal{X}$	feature space	$P_{X^sY^s},\ P_{X^tY^t}$	source, target joint distributions
$\mathcal{Y}^s, \mathcal{Y}^t$	source, target label sets $\{\mathbf{y}_c\}_{c=1}^{K}, \{\mathbf{y}_c\}_{c=1}^{K+1}$	$P_{X^s},\ P_{X^t}$	source, target marginal distributions
X^s, X^t	random variables on the feature space	Δ	open set difference
$Y^s,\ Y^t$	random variables on the label spaces	$P_{X^t\|\mathcal{Y}^s}$	$P(X^t\|Y^t \in \mathcal{Y}^s)$
$L^s(\cdot), L^t(\cdot)$	source, target risks	$L^t_*(\cdot)$	partial risk on known target classes
$\mathbf{y}_c$	one-hot vector (class c)	$L^t_{K+1}(\cdot)$	partial risk on unknown target classes
$\boldsymbol{G}, \boldsymbol{C}$	feature transformation , classifier over $\boldsymbol{G}(\mathcal{X})$	$L^s_{u,K+1}, L^t_{u,K+1}$	risks that samples regarded as unknown
$\mathcal{H}_{\boldsymbol{G}}$	hypothesis space, set of classifiers $\boldsymbol{C}$	π^t_{K+1}	class-prior probability for unknown class
$\mathcal{X}_{\boldsymbol{G}}, \mathbf{x}_{\boldsymbol{G}}$	$\boldsymbol{G}(\mathcal{X})$, sample from $\boldsymbol{G}(\mathcal{X})$	$\widehat{P}, \widehat{L}(\cdot)$	empirical distribution, empirical risk
$d^{\ell}_{\mathcal{H}_{\boldsymbol{G}}}(\cdot,\cdot)$	$\mathcal{H}\Delta\mathcal{H}$ distance	$d^{\ell}_{\Delta_{\boldsymbol{C},\boldsymbol{G}}}(\cdot,\cdot)$	tensor discrepancy distance

B. Concepts and Notations

It is necessary to introduce some important concepts and notations before demonstrating our main results. Unless otherwise specified, all the following notations are used consistently throughout this article without further explanations.

1) Notations for Distributions: For simplicity, we denote the joint distributions $P(X^s, Y^s)$ and $P(X^t, Y^t)$ by the notations $P_{X^sY^s}$ and $P_{X^tY^t}$, respectively. Similarly, we use P_{X^s} and P_{X^t} to denote the marginal distributions $P(X^s)$ and $P(X^t)$, respectively.

$P_{X^t|\mathcal{Y}^s}$ denotes the target conditional distribution for the known classes, while $P_{X^t|\mathbf{y}_{K+1}}$ denotes the target conditional distribution for the unknown classes. $\pi^t_{K+1} = P(Y^t = \mathbf{y}_{K+1})$ denotes the class-prior probability for the unknown target classes.

Given a feature transformation

$$\begin{aligned} \boldsymbol{G} : \mathcal{X} &\to \mathcal{X}_{\boldsymbol{G}} := \boldsymbol{G}(\mathcal{X}) \\ \mathbf{x} &\to \mathbf{x}_{\boldsymbol{G}} := \boldsymbol{G}(\mathbf{x}) \end{aligned} \tag{1}$$

the induced distributions related to P_{X^s} and $P_{X^t|\mathcal{Y}^t}$ are

$$\begin{aligned} \boldsymbol{G}_{\#}P_{X^s} &:= P(\boldsymbol{G}(X^s)) \\ \boldsymbol{G}_{\#}P_{X^t|\mathcal{Y}^s} &:= P(\boldsymbol{G}(X^t)|Y^t \in \mathcal{Y}^s). \end{aligned} \tag{2}$$

Finally, the notation $\widehat{P}$ denotes the corresponding empirical distribution to any distribution P. For example, $\widehat{P}_{X^sY^s}$ represents the empirical distribution corresponding to $P_{X^sY^s}$.

2) Risks and Partial Risks: In learning theory, risks and partial risks are two important concepts, which are briefly explained below.

Following the notations in [44], consider a multiclass classification task with a *hypothesis space* $\mathcal{H}_{\boldsymbol{G}}$ of the classifiers:

$$\begin{aligned} \boldsymbol{C} :\ \mathcal{X}_{\boldsymbol{G}} &\to \mathcal{Y}^t \\ \mathbf{x} &\to [C_1(\mathbf{x}), \ldots, C_{K+1}(\mathbf{x})]^T. \end{aligned} \tag{3}$$

Let

$$\begin{aligned} \ell : \mathbb{R}^{K+1} \times \mathbb{R}^{K+1} &\to \mathbb{R}_{\geq 0} \\ (\mathbf{y}, \tilde{\mathbf{y}}) &\to \ell(\mathbf{y}, \tilde{\mathbf{y}}) \end{aligned} \tag{4}$$

be the loss function. For convenience, we also require ℓ to satisfy the following conditions in Theorem 1.

1) ℓ is symmetric and satisfies triangle inequality.
2) $\ell(\mathbf{y}, \tilde{\mathbf{y}}) = 0$ iff $\mathbf{y} = \tilde{\mathbf{y}}$.
3) $\ell(\mathbf{y}, \tilde{\mathbf{y}}) \equiv 1$ if $\mathbf{y} \neq \tilde{\mathbf{y}}$ and $\mathbf{y}, \tilde{\mathbf{y}}$ are the one-hot vectors.

We can check many losses satisfying the above conditions such as (0)–(1) loss $1_{\mathbf{y}\neq\tilde{\mathbf{y}}}$ and ℓ_2 loss $(1/2)\|\mathbf{y} - \tilde{\mathbf{y}}\|_2^2$.

Then, the *risks* of $\boldsymbol{C} \in \mathcal{H}_{\boldsymbol{G}}$ w.r.t. ℓ under $\boldsymbol{G}_{\#}P_{X^sY^s}$ and $\boldsymbol{G}_{\#}P_{X^tY^t}$ are given by

$$\begin{aligned} L^s(\boldsymbol{C} \circ \boldsymbol{G}) &:= \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim P_{X^sY^s}} \ell(\boldsymbol{C} \circ \boldsymbol{G}(\mathbf{x}), \mathbf{y}) \\ L^t(\boldsymbol{C} \circ \boldsymbol{G}) &:= \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim P_{X^tY^t}} \ell(\boldsymbol{C} \circ \boldsymbol{G}(\mathbf{x}), \mathbf{y}). \end{aligned} \tag{5}$$

The *partial risk* of $\boldsymbol{C} \in \mathcal{H}_{\boldsymbol{G}}$ for the known target classes is

$$L^t_*(\boldsymbol{C} \circ \boldsymbol{G}) := \frac{1}{1 - \pi^t_{K+1}} \int_{\mathcal{X}\times\mathcal{Y}^s} \ell(\boldsymbol{C} \circ \boldsymbol{G}(\mathbf{x}), \mathbf{y}) \mathrm{d}P_{X^tY^t} \tag{6}$$

and the *partial risk* of $\boldsymbol{C} \in \mathcal{H}_{\boldsymbol{G}}$ for the unknown target classes is

$$L^t_{K+1}(\boldsymbol{C} \circ \boldsymbol{G}) : = \mathop{\mathbb{E}}_{\mathbf{x}\sim P_{X^t|\mathbf{y}_{K+1}}} \ell(\boldsymbol{C} \circ \boldsymbol{G}(\mathbf{x}), \mathbf{y}_{K+1}). \tag{7}$$

Finally, we denote

$$\begin{aligned} L^s_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G}) &:= \mathop{\mathbb{E}}_{\mathbf{x}\sim P_{X^s}} \ell(\boldsymbol{C} \circ \boldsymbol{G}(\mathbf{x}), \mathbf{y}_{K+1}) \\ L^t_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G}) &:= \mathop{\mathbb{E}}_{\mathbf{x}\sim P_{X^t}} \ell(\boldsymbol{C} \circ \boldsymbol{G}(\mathbf{x}), \mathbf{y}_{K+1}) \end{aligned} \tag{8}$$

as the *risks* that the samples are regarded as the unknown classes.

Given a risk $L(\boldsymbol{C} \circ \boldsymbol{G})$, it is convenient to use notation $\widehat{L}(\boldsymbol{C} \circ \boldsymbol{G})$ as the empirical risk that corresponds to $L(\boldsymbol{C} \circ \boldsymbol{G})$.

3) Discrepancy Distance: How to measure the difference between domains plays a critical role in DA. To achieve this, a famous distribution distance has been proposed as the measures of the distribution difference.

Definition 3 (Distributional Discrepancy [45]): Given a hypothesis space $\mathcal{H}_{\boldsymbol{G}}$ containing a set of functions defined in a feature space $\mathcal{X}_{\boldsymbol{G}}$. Let ℓ be a loss function, and P_1, P_2 be distributions on space $\mathcal{X}_{\boldsymbol{G}}$. The $\mathcal{H}\Delta\mathcal{H}$ distance $d^{\ell}_{\mathcal{H}_{\boldsymbol{G}}}(P_1, P_2)$ between distributions P_1 and P_2 over $\mathcal{X}_G$ is

$$\sup_{\boldsymbol{C},\boldsymbol{C}^*\in\mathcal{H}_{\boldsymbol{G}}} \left| \mathop{\mathbb{E}}_{\mathbf{x}\sim P_1} \ell(\boldsymbol{C}(\mathbf{x}), \boldsymbol{C}^*(\mathbf{x})) - \mathop{\mathbb{E}}_{\mathbf{x}\sim P_2} \ell(\boldsymbol{C}(\mathbf{x}), \boldsymbol{C}^*(\mathbf{x})) \right|.$$

In this article, we have used a tighter distance named tensor discrepancy distance, which is first proposed by

Long *et al.* [28]. The tensor discrepancy distance can extract the multimodal structure of distributions to make sure the knowledge related to learned classifier and pseudo labels can be used during the distribution aligning process.

We consider the following tensor mapping:

$$\begin{aligned} \otimes_{\boldsymbol{C}} : \mathcal{X}_G &\to \mathcal{X}_{\boldsymbol{G}} \otimes \mathcal{Y}^t \\ \mathbf{x}_{\boldsymbol{G}} &\to \mathbf{x}_{\boldsymbol{G}} \otimes \boldsymbol{C}(\mathbf{x}_{\boldsymbol{G}}). \end{aligned} \tag{9}$$

Then, we induce two importance distributions

$$\begin{aligned} \otimes_{\boldsymbol{C}\#} P_{X^s} &:= P(\otimes_{\boldsymbol{C}}(\boldsymbol{G}(X^s))) \\ \otimes_{\boldsymbol{C}\#} P_{X^t|\mathcal{Y}^s} &:= P(\otimes_{\boldsymbol{C}}(\boldsymbol{G}(X^t))|Y^t \in \mathcal{Y}^s). \end{aligned} \tag{10}$$

Using $\mathcal{H}_{\boldsymbol{G}} \subset \{\overline{\boldsymbol{C}} : \mathcal{X}_{\boldsymbol{G}} \to \mathcal{Y}^t\}$, we reconstruct a new hypothetical set

$$\Delta_{\boldsymbol{C},\boldsymbol{G}} := \{\delta_{\overline{\boldsymbol{C}}} : \mathcal{X}_{\boldsymbol{G}} \otimes \mathcal{Y}^t \to \mathbb{R} : \overline{\boldsymbol{C}} \in \mathcal{H}_{\boldsymbol{G}}\} \tag{11}$$

where $\delta_{\overline{\boldsymbol{C}}}(\mathbf{x}_{\boldsymbol{G}} \otimes \mathbf{y}) = |\otimes_{\boldsymbol{C}}(\mathbf{x}_{\boldsymbol{G}}) - \otimes_{\overline{\boldsymbol{C}}}(\mathbf{x}_{\boldsymbol{G}})|$. Then, the distance between $\otimes_{\boldsymbol{C}\#} P_{X^s}$ and $\otimes_{\boldsymbol{C}\#} P_{X^t|\mathcal{Y}^s}$ is

$$\begin{aligned} &d^{\ell}_{\Delta_{\boldsymbol{C},\boldsymbol{G}}}(\otimes_{\boldsymbol{C}\#} P_{X^s}, \otimes_{\boldsymbol{C}\#} P_{X^t|\mathcal{Y}^s}) \\ &= \sup_{\delta \in \Delta_{\boldsymbol{C},\boldsymbol{G}}} \left| \mathop{\mathbb{E}}_{\mathbf{z} \sim \otimes_{\boldsymbol{C}\#} P_{X^s}} \operatorname{sgn} \circ \delta(\mathbf{z}) - \mathop{\mathbb{E}}_{\mathbf{z} \sim \otimes_{\boldsymbol{C}\#} P_{X^t|\mathcal{Y}^s}} \operatorname{sgn} \circ \delta(\mathbf{z}) \right| \end{aligned} \tag{12}$$

where sgn is the sign function.

It is easy to prove that under the conditions (1)–(3) for loss ℓ and for any $\boldsymbol{C} \in \mathcal{H}_{\boldsymbol{G}}$, we have

$$d^{\ell}_{\Delta_{\boldsymbol{C},\boldsymbol{G}}}(\otimes_{\boldsymbol{C}\#} P_{X^s}, \otimes_{\boldsymbol{C}\#} P_{X^t|\mathcal{Y}^s}) \le d^{\ell}_{\mathcal{H}_{\boldsymbol{G}}}(\boldsymbol{G}_{\#} P_{X^s}, \boldsymbol{G}_{\#} P_{X^t|\mathcal{Y}^s}). \tag{13}$$

4) Existing Theoretical Bound: Fang *et al.* [20] first proposed a theoretical bound for UOSDA

$$\begin{aligned} \frac{L^t(\boldsymbol{C} \circ \boldsymbol{G})}{1 - \pi^t_{K+1}} \le \overbrace{L^s(\boldsymbol{C} \circ \boldsymbol{G})}^{\text{Source Risk}} + \overbrace{2 d^{\ell}_{\mathcal{H}_{\boldsymbol{G}}}(\boldsymbol{G}_{\#} P_{X^s}, \boldsymbol{G}_{\#} P_{X^t|\mathcal{Y}^s})}^{\text{distributional discrepancy}} + \Lambda \\ + \underbrace{\frac{L^t_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G})}{1 - \pi^t_{K+1}} - L^s_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G})}_{\text{Open Set Difference } \Delta}. \end{aligned} \tag{14}$$

There are four main terms: source risk, distributional discrepancy, a constant Λ, and open set difference. The fourth term, open set difference, is designed to estimate the risk of classifier on unknown data.

IV. Motivation

In UOSDA, the target-domain classifier aims to accurately recognize unknown target data and classify the other target data. Since the knowledge about unknown classes is missing, the classifier is likely to be confused about the boundary between known and unknown target data. Thus, recognizing unknown target data plays a critical role in addressing the UOSDA problem.

To obtain an effective target-domain classifier, Fang *et al.* [20] have proven an upper [see (14)] bound for UOSDA and proposed a *shallow* method based on the bound. It consists of four terms: source-domain risk, distributional discrepancy, *open set difference* (Δ), and a constant. Particularly, open set difference, as an important term, is leveraged to estimate the risk of the classifier on unknown target data.

To verify whether open set difference works in DNNs, we introduced open set difference into DNNs and conducted a group of experiments on the task $Ar \to Cl$ in *Office-Home*. The classifier consists of backbone (ResNet50), generator (two linear layers), and classifier (one linear layer). It is evident that the classifier is very flexible. As shown in Fig. 2, the empirical open set difference converges to a negative value [refer to the yellow line in Fig. 2(a)] and the accuracy of OS, average accuracy among all classes that include unknown classes [see (29)], significantly decreases when empirical open set difference converges to a negative value.

To reveal the nature of this phenomenon, first we investigate the distributional discrepancy and find that the distributional discrepancy has a lower bound. Specifically, the distributional discrepancy is greater than the negative value of open set difference [see (18)]. Based on the lower bound, if the value of the open set difference is a large negative number, then the distributional discrepancy is greater than a large positive number. Hence, we may fail to align the distributional discrepancy. In fact, experiments have shown that the empirical open set difference may converge to a large negative value if we introduce the open set difference into DNNs.

Clearly, there is a gap between the existing theoretical bound and DNNs. To bridge theoretical bound and deep algorithms, in this article, we propose a new practical upper bound [see (20)] for UOSDA that applies to DNNs. The term ϵ-open set difference in the new bound can effectively overcome the defect of open set difference. As shown in Fig. 2, ϵ-open set difference guarantees that the risk of the classifier on unknown data is always greater than the lower bound of open set difference by ϵ [refer to the green line in Fig. 2(a)]. Furthermore, the ϵ-open set difference significantly outperforms the open set difference [refer to the green line in Fig. 2(b)].

To sum up, the existing upper bound is not compatible with DNNs. That is why we propose a new upper bound that contains an amended risk estimator, ϵ-open set difference (Δ_ϵ). Details of the new upper bound and Δ_ϵ are shown in Section V.

V. Proposed Method

In this section, we first propose a theoretical bound that applies to DNNs for UOSDA. Under the guidance of the bound, we then propose a UOSDA method based on DNNs.

A. Theoretical Results

1) Analysis for Open Set Difference: Equation (15) is the open set difference

$$\Delta = \frac{L^t_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G})}{1 - \pi^t_{K+1}} - L^s_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G}) \tag{15}$$

where $L^t_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G})$ and $L^s_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G})$ are defined in (8). The positive term $L^t_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G})$ is used to recognize unknown data and the negative term $L^s_{u,K+1}(\boldsymbol{C} \circ \boldsymbol{G})$ is designed to prevent known data from being classified as unknown classes. By combining these two terms, the classifier can recognize unknown target samples. According to [20], the open set

difference Δ satisfies the following inequality:

$$\begin{aligned}\Delta &= \frac{L^t_{u,K+1}(\boldsymbol{C}\circ\boldsymbol{G})}{(1-\pi^t_{K+1})} - L^s_{u,K+1}(\boldsymbol{C}\circ\boldsymbol{G}) \\ &\geq \frac{\pi^t_{K+1}}{\left(1-\pi^t_{K+1}\right)} L^t_{K+1}(\boldsymbol{C}\circ\boldsymbol{G}) - d^{\ell}_{\mathcal{H}_{\boldsymbol{G}}}(\boldsymbol{G}_{\#}P_{X^s}, \boldsymbol{G}_{\#}P_{X^t|\mathcal{Y}^s}).\end{aligned} \tag{16}$$

The proof of (16) can be found in Appendix A in the supplementary material. Proposition 1. Note that

$$\frac{\pi^t_{K+1}}{\left(1-\pi^t_{K+1}\right)} L^t_{K+1}(\boldsymbol{C}\circ\boldsymbol{G}) \geq 0 \tag{17}$$

hence, the distributional discrepancy is greater than the negative open set difference

$$d^{\ell}_{\mathcal{H}_{\boldsymbol{G}}}(\boldsymbol{G}_{\#}P_{X^s}, \boldsymbol{G}_{\#}P_{X^t|\mathcal{Y}^s}) \geq -\Delta. \tag{18}$$

Theoretically, we hope that the optimized open set difference should not be a large negative value. Otherwise, it is impossible to eliminate the distributional discrepancy. However, in fact, the empirical open set difference $\widehat{\Delta}$ may converge to a large negative value (see Fig. 2). This results in that the distributional discrepancy may still be large.

2) ϵ-Open Set Difference: Based on the analyses above, we try to correct the open set difference to avoid the problem mentioned above. According to (18), the open set difference is lower bounded. We denoted the lower bound of the open set difference by ϵ. A potentiality is to limit the lower bound of the open set difference by a small negative constant $-\epsilon$. Hence, we propose an amended risk estimator, ϵ-open set difference (Δ_ϵ), to overcome the existing defect in the open set difference

$$\Delta_\epsilon = \max\{-\epsilon, \frac{L^t_{u,K+1}(\boldsymbol{C}\circ\boldsymbol{G})}{1-\pi^t_{K+1}} - L^s_{u,K+1}(\boldsymbol{C}\circ\boldsymbol{G})\}. \tag{19}$$

If we optimize the empirical ϵ-open set difference, we can guarantee that the empirical ϵ-open set difference is always larger than $-\epsilon$. Finally, combining (12) and (13) with (19), we develop a new theoretical bound for UOSDA.

Theorem 1: Given a feature transformation $\boldsymbol{G}: \mathcal{X} \to \mathcal{X}_{\boldsymbol{G}}$, a loss function ℓ satisfying conditions (1)–(3) introduced in Section III-B.2, a nonegative constant ϵ, and a hypothesis $\mathcal{H}_{\boldsymbol{G}} \subset \{\boldsymbol{C}: \mathcal{X}_{\boldsymbol{G}} \to \mathcal{Y}^t\}$ with a mild condition that the constant vector value function $\widetilde{\boldsymbol{C}} := \mathbf{y}_{C+1} \in \mathcal{H}_{\boldsymbol{G}}$, then for any $\boldsymbol{C} \in \mathcal{H}_{\boldsymbol{G}}$, we have

$$\begin{aligned}&\frac{L^t(\boldsymbol{C}\circ\boldsymbol{G})}{1-\pi^t_{K+1}} \\ &\leq \overbrace{L^s(\boldsymbol{C}\circ\boldsymbol{G})}^{\text{Source Risk}} + \overbrace{2d^{\ell}_{\Delta_{\boldsymbol{C},\boldsymbol{G}}}(\otimes_{\boldsymbol{C}\#}P_{X^s}, \otimes_{\boldsymbol{C}\#}P_{X^t|\mathcal{Y}^s})}^{\text{Tensor distributional discrepancy}} \\ &\quad + \underbrace{\max\left\{-\epsilon, \frac{L^t_{u,K+1}(\boldsymbol{C}\circ\boldsymbol{G})}{1-\pi^t_{K+1}} - L^s_{u,K+1}(\boldsymbol{C}\circ\boldsymbol{G})\right\}}_{\epsilon\text{-Open Set Difference } \Delta_\epsilon} + \Lambda\end{aligned} \tag{20}$$

where $L^s(\boldsymbol{C}\circ\boldsymbol{G})$ and $L^t(\boldsymbol{C}\circ\boldsymbol{G})$ are the risks defined in (5), $L^s_{u,K+1}(\boldsymbol{C}\circ\boldsymbol{G})$ and $L^t_{u,K+1}(\boldsymbol{C}\circ\boldsymbol{G})$ are the risks defined in (8), $L^t_*(\boldsymbol{C}\circ\boldsymbol{G})$ is the partial risk defined in (6), and $\Lambda = \min_{\boldsymbol{C}\in\mathcal{H}_{\boldsymbol{G}}} L^s(\boldsymbol{C}\circ\boldsymbol{G}) + L^t_*(\boldsymbol{C}\circ\boldsymbol{G})$.

TABLE II
NOTATIONS AND THEIR DESCRIPTIONS

Notation	Description
ℓ_{ce}, ℓ_{mse}	cross entropy, mean square error loss function
$\mathcal{T}^*_u$	set of predicted unknown target data with high confidence
$\mathcal{T}^*_K$	set of predicted known target data with high confidence
n^s	number of source data
n^t	number of target data
n^*_K	number of $\mathcal{T}^*_K$
n^*_u	number of $\mathcal{T}_u$
$\mathbf{x}^s_i$	source data
$\mathbf{x}^t_i$	target data

Proof: The proof is given in Appendix A in the supplementary material. □

It is notable that the theoretical bound introduced in Theorem 1 has two main differences from the learning bound introduced by Fang *et al.* [20]. The first one is the ϵ-open set difference. As mentioned before, ϵ-open set difference is designed to eliminate the distributional discrepancy caused by open set difference when the module is based on DNNs. The other difference is that we use the tensor distributional discrepancy to estimate the domain difference. There are two advantages for the tensor distributional discrepancy compared with the distributional discrepancy (Definition 3): 1) the tensor distributional discrepancy is tighter than the distributional discrepancy [see (13)] and 2) the tensor distributional discrepancy can extract the multimodal structure of distributions to make sure the knowledge related to the learned classifier and pseudo labels can be used during the process of distribution alignment [28].

B. Method Description

According to Theorem 1, we formally present our method (see Fig. 3), which consists of three parts. Part 1) Binary adversarial DA (BADA). Following [19], we yse a binary adversarial module to find a rough boundary between the class-known data (*known data*) and the class-unknown data (*unknown data*), and thus this module can provide target samples with high confidence for other modules. Part 2) ϵ-open set difference (Δ_ϵ). The Δ_ϵ is leveraged to estimate the risk of the classifier on unknown data such that the classifier can accurately recognize the unknown target data. Part 3) CDAN. The existing deep UOSDA methods ignore the importance of the multimodal structure of distribution while aligning distributions for known classes. According to the tensor distributional discrepancy, we design a novel open set conditional adversarial strategy to align distributions for known classes. The notations used in this section are summarized in Table II.

1) Binary Adversarial DA: According to our theoretical bound, the first term is the source risk. For the source domain, the label is available. We use a cross-entropy for the

classification of source samples

$$\widehat{L}^s_{cls} = \frac{1}{n^s}\sum_{i=1}^{n^s} \ell_{ce}\big(\boldsymbol{C}\circ\boldsymbol{G}\big(\mathbf{x}^s_i\big), \mathbf{y}^s_i\big). \tag{21}$$

For the target domain, it is imperative to recognize the unknown target data before aligning distribution. Following [19], we use a binary cross-entropy and a gradient reverse layer between generator and classifier to find a boundary between the known data and the unknown data

$$\widehat{L}_{badv} = -\frac{1}{2n^t}\sum_{i=1}^{n^t} \log\big(\big(C_{K+1}\circ\boldsymbol{G}\big(\mathbf{x}^t_i\big)\big)\big)\big(1-\big(C_{K+1}\circ\boldsymbol{G}\big(\mathbf{x}^t_i\big)\big)\big) \tag{22}$$

where C_{K+1} is the $K+1$th value of hypothesis function $\boldsymbol{C}$.

The minimax game is shown in Section V-C. During the process of adversarial training, the classifier attempts to minimize $\widehat{L}_{badv}$, but the generator attempts to maximize $\widehat{L}_{badv}$. Therefore, recognizing unknown data is achieved during the process of adversarial training.

However, this module can only find a coarse boundary between the known data and the unknown data, which cannot accurately recognize the unknown target data. Table VII verifies that only BADA cannot achieve satisfactory performance. Therefore, we use the ϵ-open set difference for recognizing unknown target data more appropriately and the open set conditional adversarial strategy to further align distribution.

2) ϵ-Open Set Difference: The principle of the ϵ-open set difference (Δ_ϵ) is adequately demonstrated in Sections IV and V-A. Then we introduce Δ_ϵ to recognize unknown target data. According to (19), (23), we can calculate the empirical ϵ-open set difference $\widehat{\Delta}_\epsilon$ by

$$\max\Bigg\{-\epsilon, \frac{\alpha}{n^t}\sum_{i=1}^{n^t}\ell_{mse}\big(\boldsymbol{C}\circ\boldsymbol{G}\big(\mathbf{x}^t_i\big), \mathbf{y}_{K+1}\big) - \frac{1}{n^s}\sum_{i=1}^{n^s}\ell_{mse}\big(\boldsymbol{C}\circ\boldsymbol{G}\big(\mathbf{x}^s_i\big), \mathbf{y}_{K+1}\big)\Bigg\}. \tag{23}$$

Without more label information, π^t_{K+1} in (19) is impossible to be evaluated accurately, and thus, we introduce a parameter, α, to replace it. The analysis of α is discussed in Section VI.

3) Conditional Adversarial DA: Here, we use the tensor distributional discrepancy to align the distribution between the known classes. First, the empirical representations of $\otimes_{\boldsymbol{C}\#}\widehat{P}_{X^s}$ and $\otimes_{\boldsymbol{C}\#}\widehat{P}_{X^t|\mathcal{Y}^s}$ can be written as follows:

$$\begin{aligned}\otimes_{\boldsymbol{C}\#}\widehat{P}_{X^s} &= \frac{1}{n^s}\sum_{i=1}^{n^s} 1_{\boldsymbol{G}(\mathbf{x}^s_i)\otimes\boldsymbol{C}\circ\boldsymbol{G}(\mathbf{x}^s_i)}\\ \otimes_{\boldsymbol{C}\#}\widehat{P}_{X^t|\mathcal{Y}^s} &= \frac{1}{|\mathcal{T}_K|}\sum_{\mathbf{x}\in\mathcal{T}_X} 1_{\boldsymbol{G}(\mathbf{x})\otimes\boldsymbol{C}\circ\boldsymbol{G}(\mathbf{x})}\end{aligned} \tag{24}$$

where $\mathcal{T}_K$ is the set of target data from the known classes and $1_{\boldsymbol{G}(\mathbf{x})\otimes\boldsymbol{C}\circ\boldsymbol{G}(\mathbf{x})}$ is the Dirac measure.

Then, motivated by DANN [39] and CDAN [28], we can reformulate the tensor distributional discrepancy between the known classes as follows:

$$-\frac{1}{n^s}\sum_{i=1}^{n^s}\log\big(\boldsymbol{D}\big(\boldsymbol{G}\big(\mathbf{x}^s_i\big)\otimes\boldsymbol{C}\circ\boldsymbol{G}\big(\mathbf{x}^s_i\big)\big)\big) - \frac{1}{|\mathcal{T}_K|}\sum_{\mathbf{x}\in\mathcal{T}_K}(1-\boldsymbol{D}(\log(\boldsymbol{G}(\mathbf{x})\otimes\boldsymbol{C}\circ\boldsymbol{G}(\mathbf{x})))) \tag{25}$$

where $\boldsymbol{D}$ is the domain discriminator designed to classify domains.

Since the target data are unlabeled, (25) cannot be directly calculated. Thanks to the pseudo labels provided by BADA, we leverage it to replace the true label. Since these pseudo labels are not completely accurate, we only select the samples with a confidence of 0.9. We then formulate the domain adversarial loss function below

$$\widehat{L}_{dadv} = -\frac{1}{n^s}\sum_{i=1}^{n^s}\log\big(\boldsymbol{D}\big(\boldsymbol{G}\big(\mathbf{x}^s_i\big)\otimes\boldsymbol{C}\circ\boldsymbol{G}\big(\mathbf{x}^s_i\big)\big)\big) - \frac{1}{n^*_K}\sum_{\mathbf{x}\in\mathcal{T}^*_K}\big(1-\boldsymbol{D}(\log(\boldsymbol{G}(\mathbf{x})\otimes\boldsymbol{C}\circ\boldsymbol{G}(\mathbf{x})))\big) \tag{26}$$

where $\mathcal{T}^*_K$ denotes the set of samples from known classes with high confidence in the target domain, and $n^*_K = |\mathcal{T}^*_K|$.

Domain adversary loss aims to minimize over $\boldsymbol{D}$ and maximize over $\boldsymbol{G}$. The gradient reverse layer between $\boldsymbol{G}$ and $\boldsymbol{D}$ results in $\boldsymbol{D}$ becoming confused about the source data and the target data. The minimax game is shown in Section V-C. The classifier aims to identify what input data belong to which domain, but the generator aims to deceive the classifier by changing the features of the input data. Distribution alignment can be achieved during this process.

Furthermore, the unknown data may distract distribution alignment of the known data. Thus, the unknown data should be pushed away from known data to prevent them from affecting distribution alignment. We construct the loss function below. It is worth noting that there is no gradient reverse between $\boldsymbol{D}$ and $\boldsymbol{G}$ during the process of backpropagation

$$\widehat{L}_d = -\frac{1}{n^s}\sum_{i=1}^{n^s}\log\big(\boldsymbol{D}\big(\boldsymbol{G}\big(\mathbf{x}^s_i\big)\otimes\boldsymbol{C}\circ\boldsymbol{G}\big(\mathbf{x}^s_i\big)\big)\big) - \frac{1}{n^*_u}\sum_{\mathbf{x}\in\mathcal{T}^*_u}\big(1-\boldsymbol{D}(\log(\boldsymbol{G}(\mathbf{x})\otimes\boldsymbol{C}\circ\boldsymbol{G}(\mathbf{x})))\big) \tag{27}$$

where $\mathcal{T}^*_u$ is the unknown target samples with high confidence and $n^*_u = |\mathcal{T}^*_u|$.

In this section, we construct a domain discriminator ($\boldsymbol{D}$) to align the distributions for the known data by a tensor product, which can capture the multimodal structure of distribution. Furthermore, we construct a loss function to push the unknown data away from the known data to prevent the unknown data affecting distribution alignment.

Algorithm 1 Training Procedure of Our Method

Input: source samples $\{\mathbf{x}_i^s, \mathbf{y}_i^s\}_{i=1}^{n^s}$, target samples $\{\mathbf{x}_i^t\}_{i=1}^{n^t}$.
Parameter: learning rate γ, batch size m, the number of iterations T, network parameters $\theta_{\boldsymbol{G}}$, $\theta_{\boldsymbol{C}}$, $\theta_{\boldsymbol{D}}$.
Output: predicted target label $\widehat{\mathbf{y}}_t$.

1: Initialize $\theta_{\boldsymbol{G}}$, $\theta_{\boldsymbol{C}}$, $\theta_{\boldsymbol{D}}$
2: t=0
3: **while** $t < T$ **do**
4: sample source minibatch $\{(\mathbf{x}_{i_1}^s, \mathbf{y}_{i_1}^s), \ldots, (\mathbf{x}_{i_m}^s, \mathbf{y}_{i_m}^s)\}$.
5: sample target minibatch $\{\mathbf{x}_{i_1}^t, \ldots, \mathbf{x}_{i_m}^t\}$.
6: calculate $\widehat{L}_s$, $\widehat{L}_{badv}$ according to Eqs. (21) and (22).
7: calculate $\widehat{\Delta}_\epsilon$ according to Eq. (23).
8: select high confidence target samples according to the output of softmax g_t.
9: calculate L_{dadv}, L_d according to Eqs. (26) and (27) by leveraging high confidence target samples.
10: update parameter:
$\theta_{\boldsymbol{G}} = \theta_{\boldsymbol{G}} - \gamma \bigtriangledown_{\theta_{\boldsymbol{G}}} (\widehat{L}_{cls}^s - \widehat{L}_{adv} + \widehat{\Delta}_\epsilon - \widehat{L}_{dadv} + \widehat{L}_d)$
$\theta_{\boldsymbol{C}} = \theta_{\boldsymbol{C}} - \gamma \bigtriangledown_{\theta_{\boldsymbol{C}}} (\widehat{L}_{cls}^s + \widehat{L}_{adv} + \widehat{\Delta}_\epsilon)$
$\theta_{\boldsymbol{D}} = \theta_{\boldsymbol{D}} - \gamma \bigtriangledown_{\theta_{\boldsymbol{D}}} (\widehat{L}_{dadv} + \widehat{L}_d)$.
11: $t = t + 1$
12: **end while**

C. Training Procedure

Combining (21)–(23), (26), and (27), we solve the UOSDA problem by the following minimax game:

$$\begin{aligned} &\min_{\boldsymbol{G}} \widehat{L}_{cls}^s - \widehat{L}_{badv} + \widehat{\Delta}_\epsilon - \widehat{L}_{dadv} + \widehat{L}_d \\ &\min_{\boldsymbol{C}} \widehat{L}_{cls}^s + \widehat{L}_{badv} + \widehat{\Delta}_\epsilon \\ &\min_{\boldsymbol{D}} \widehat{L}_{dadv} + \widehat{L}_d. \end{aligned} \tag{28}$$

We introduce the gradient reverse layer for adversary learning. The whole training procedure is shown in Algorithm 1. First, we initialize the parameters of the generator ($\boldsymbol{G}$), the classifier ($\boldsymbol{C}$), and the domain discriminator ($\boldsymbol{D}$) (line 1). In each epoch, we divide data into multiminibatches (lines 4 and 5). Then we calculate source risk ($\widehat{L}_{cls}^s$), binary adversarial loss ($\widehat{L}_{badv}$), and Δ_ϵ according to (21)–(23) (lines 6 and 7). After selecting target samples with high confidence ($\geq$ 0.9) (line 8), we calculate $\widehat{L}_{dadv}$ and $\widehat{L}_d$ according to (26) and (27) (line 9). Finally, the parameters are updated via the stochastic gradient descent (SGD) optimizer (line 10).

With the proposed method, in BADA ($\widehat{L}_{cls}^s$, $\widehat{L}_{badv}$), a coarse boundary between known data and unknown data can be found. Furthermore, ϵ-open set difference ($\widehat{\Delta}_\epsilon$) can adequately estimate the risk of the classifier on unknown data, which is effective for the classifier to accurately recognize unknown target data. Then, we further align distributions of known data ($\widehat{L}_{dadv}$) and push unknown data away from known data ($\widehat{L}_d$) using a domain discriminator. Finally, combining these three modules, we can adequately solve the UOSDA problem.

VI. Experiments and Evaluations

In this section, we conducted extensive experiments on 6 standard benchmark datasets (including 41 transfer tasks) to demonstrate the effectiveness of our method. Several state-of-the-art UOSDA methods such ATI-λ [18], OSBP [19], SCI_SCM [25], STA [26], and DAOD [20] are used as our baselines.

A. Datasets

Digits contains three digit datasets: *MNIST* (*M*) [46], *SVHN* (*S*) [47], and *USPS* (*U*) [48]. We construct three open set DA tasks as previous works [19]: $S \rightarrow M$, $M \rightarrow U$, and $U \rightarrow M$. Following the protocol of [19], we select classes 0–4 as the known classes and classes 5–9 as the unknown classes of the target domain.

Office-31 [49] is an object recognition dataset with 4110 imges, which consists of three domains with slight distributional discrepancy: *amazon* (*A*), *dslr* (*D*), and *webcam* (*W*). Each domain contains 31 kinds of object. So there are 6 open set DA tasks on *Office-31*: $A \rightarrow D$, $A \rightarrow W$, $D \rightarrow A$, $D \rightarrow W$, $W \rightarrow A$, and $W \rightarrow D$. We follow the open set protocol of [19], selecting the first ten classes in alphabetical order as the known classes and classes 21–31 as the unknown classes of the target domain.

Office-Home [50] is an object recognition dataset with 15 500 image, which contains four domains with more obvious distributional discrepancy than *Office-31*. These domains are *Artistic* (*Ar*), *Clipart* (*Cl*), *Product* (*Pr*), and *Real-World* (*Rw*). Each domain contains 65 kinds of objects. So there are 12 open set DA tasks on *Office-Home*: $Ar \rightarrow Cl$, $Ar \rightarrow Pr$, $Ar \rightarrow Rw, \ldots, Rw \rightarrow Pr$. Following the standard protocol, we chose the first 25 classes as the known classes and 26–65 classes as the unknown classes of the target domain.

PIE [51] is a face recognition dataset, containing 41 368 images of 68 people with multifarious pose, illumination, and expression, following the protocol of [20], We performed open set DA among 5 out of 13 poses and selected classes 1–20 as the known classes and classes 21–68 as the unknown classes of the target domain: *PIE1* (left pose), *PIE2* (upward pose), *PIE3* (downward pose), *PIE4* (frontal pose), and *PIE5* (right pose). We construct 20 open set DA tasks, i.e., $PIE1 \rightarrow PIE2$, $PIE1 \rightarrow PIE3, \ldots, PIE5 \rightarrow PIE4$.

ImageNet-Caltech is constructed from ImageNet-1K [52] and Caltech-256.[1] Note that the validation set of ImageNet-1K is adopt instead of the training set to avoid the effect of the pretrained model on ImageNet. The validation set of ImageNet-1K contains 50 000 images that consist of 1000 classes. Caltech-256 contains 30 607 images that consist of 256 classes. They share 84 common classes. Two transfer tasks can be constructed: ImageNet-1K $\rightarrow$ Caltech-84 ($I \rightarrow C$) and Caltech-256 $\rightarrow$ ImageNet-84 ($C \rightarrow I$).

B. Implementation

1) Network Structure: For *Digits*, we use the similar convolution neural network as [19], [53] for $S \rightarrow M$ and other tasks, respectively, and train the DNNs from scratch. For *Office-31*, we use a pre-trained convolutional neural network from Visual Geometry Group (VGGNet) [54] as backbone to extract features of images. We use two fully connected layers as the generator and one fully connected layer as the classifier. For *Office-Home* and *ImageNet-Caltech*, we use

[1] https://authors.library.caltech.edu/7694/

TABLE III
ACC(OS*) AND ACC(OS) (%) ON *Digits*

Dataset	ATI-λ		OSBP		SCA_SCM		STA		DAOD		OURS	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
$S \rightarrow M$	67.6	66.5	63.1	59.1	68.6	65.5	76.9	75.4	-	-	**82.9**	**82.6**
$M \rightarrow U$	86.8	89.6	92.1	94.9	91.3	92.0	93.0	**94.9**	-	-	**93.4**	94.6
$U \rightarrow M$	82.4	81.5	92.3	91.2	**93.1**	**95.2**	92.2	91.3	-	-	90.7	92.7
Average	78.9	79.2	82.4	81.7	84.3	84.2	87.3	87.2	-	-	**89.0**	**90.0**

ResNet-50 [55] as backbone to extract features of images. The network structure of the generator and the classifier is the same as *Office-31*. PIE provides the available features of all images. Therefore, convolutional neural network (CNN) is not necessary, and we adopted a similar generator and classifier as *Office-31*. Details about the network can be found in Appendix B in the supplementary material. In the same manner as [19], [25], we do not update the parameters of the backbone during the training process.

2) Parameter Setting: In the proposed method, there are two important parameters: α and ϵ. We set ϵ as 0 in all experiments, which is because distributional discrepancy is gradually approaching 0 during the process of DA and Δ_ϵ should be greater than or equal to 0 when the distributional discrepancy is 0. Besides, we set α as 1.25 for *Office-31* and *ImageNet-Caltech*, 1.1 for *Digit* and *Office-Home*, and 1.0 for *PIE*. When the distributional discrepancy is relatively large, we advise that α should be smaller for steady training. All the experimental results are the accuracy averaged over three independent runs.

C. Baselines

We compare our method with five UOSDA methods: ATI-λ, OSBP [19], SCA_SCM [25], STA [26], and DAOD [20]. We briefly introduce these baselines in the following.

1) ATI-λ [18] uses an integer programming to assign the label for the target domain and a mapping matrix to align distribution.
2) OSBP [19] uses a classifier to align distributions between data (with known classes) in both the source and target domains and an adversarial net to reject unknown samples through the probability of samples in the target domain.
3) SCA_SCM [25] aligns the centroids between the source and the target and pushes unknown samples away from known classes to achieve a good performance.
4) STA [26] uses a coarse-to-fine weight mechanism to separate unknown samples from the target domain and achieves distribution alignment simultaneously.
5) DAOD [20] trains a target-domain classifier via minimizing (14). The term open set difference is used to estimate the risk of the classifier on unknown classes.

D. Evaluation Metrics

Following previous works [18]–[20], we use the two metrics below to evaluate our method. **OS**: average accuracy among all classes that include unknown classes. **OS***: average accuracy among known classes

$$\text{Acc(OS}^*) = \frac{1}{K}\sum_{c=1}^{K}\frac{\left|\mathbf{x} \in \mathcal{T}_c \bigwedge \mathcal{C}^t(\mathbf{x}) = \mathbf{y}_c\right|}{|\mathcal{T}_c|}$$
$$\text{Acc(OS)} = \frac{1}{K+1}\sum_{c=1}^{K+1}\frac{\left|\mathbf{x} \in \mathcal{T}_c \bigwedge \mathcal{C}^t(\mathbf{x}) = \mathbf{y}_c\right|}{|\mathcal{T}_c|} \quad (29)$$

where $\mathcal{C}^t$ is the target classifier, and $\mathcal{T}_k$ is the set of target samples with label $\mathbf{y}_c$.

E. Results

The results on three tasks of *Digits* are shown in Table III. Obviously, our method achieves the best performance (89.0% on OS and 90.0% on OS*) on three tasks. Moreover, compared with $U \rightarrow M$ and $M \rightarrow U$, $S \rightarrow M$ is more challenging. There is a bigger distributional discrepancy between S and M. On the most difficult task, our method still outperforms the best baseline STA by 6% and 7.2% on OS and OS*, respectively. It is worth noting that DAOD is a shallow method, which cannot extract feature by convolutional neural network. Therefore, there is no comparison on Digits. The results of ATI-λ are from [26].

The results on standard object datasets (*Office-31* and *Office-Home*) are recorded in Table IV. For *Office-31*, our method significantly outperforms baselines among 4 out of 6 transfer tasks. Especially on $A \rightarrow D$, our method surpasses the most competitive baseline SCA_SCM by 5.9% and 5.5% on OS and OS*, respectively. For *Office-Home*, our method also achieves better performance than baselines among 9 out of 12 transfer tasks.

The results on *PIE* are shown in Table V. Although *PIE* is a dataset with significant distributional discrepancy, our method still outperforms baselines among 17 out of 20 transfer tasks. Specifically, our method surpasses the best baseline SCA_SCM by 7.5% and 7.9% on OS and OS*, respectively.

To further verify the efficiency of our method, additional experiments on *ImageNet-Caltech* are conducted, which is usually used in partial DA [56], [57]. As shown in Table VI, even though the tasks on *ImageNet-Caltech* contain many unknown image, the proposed method can outperform baselines with higher OS and OS*, which proves the effectiveness of the proposed method.

Moreover, we observe that:

1) The performance of ATI-λ is lower than that of other methods. That is because ATI-λ cannot accurately separate unknown data, and it needs numerous unknown data in the source domain to train a classifier to recognize unknown data.
2) OSBP and SCA_SCM leverage an adversarial net to separate unknown data, which can find a rough boundary

TABLE IV
ACC(OS*) AND ACC(OS) (%) ON *Office-31* (VGG-19) AND *Office-Home* (RESNET-50)

Dataset	ATI-λ		OSBP		SCA_SCM		STA		DAOD		OURS	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
$A \rightarrow D$	79.8	86.8	89.2	90.2	90.1	92.0	88.6	92.8	89.2	91.1	**96.0**	**97.5**
$A \rightarrow W$	86.4	93.0	88.1	89.2	86.4	87.7	91.9	**94.3**	90.5	91.9	**92.5**	93.7
$D \rightarrow A$	75.0	81.5	82.5	84.3	81.6	88.4	73.4	74.3	75.4	73.6	**85.3**	**86.0**
$D \rightarrow W$	91.7	98.6	96.1	96.6	97.9	99.8	96.5	99.5	**98.6**	**100.0**	98.4	**100.0**
$W \rightarrow A$	75.8	82.0	81.3	81.5	80.3	82.6	71.3	71.3	75.6	74.7	**83.2**	**83.9**
$W \rightarrow D$	91.5	99.3	96.8	97.0	98.2	99.3	95.4	**100.0**	98.6	99.3	**98.6**	**100.0**
Average	83.4	90.2	89.0	89.8	89.1	91.6	86.2	88.7	88.0	88.4	**92.3**	**93.5**
$Ar \rightarrow Cl$	53.1	54.2	53.1	53.3	58.9	59.9	57.0	59.3	55.4	55.3	**61.6**	**62.8**
$Ar \rightarrow Pr$	68.6	70.4	68.4	69.2	73.4	74.4	67.2	69.5	71.8	72.6	**76.6**	**78.3**
$Ar \rightarrow Rw$	77.3	78.1	78.0	79.1	79.2	80.2	79.1	81.9	77.6	78.2	**83.2**	**85.0**
$Cl \rightarrow Ar$	57.8	59.1	57.9	58.2	60.6	61.5	59.1	61.3	59.2	59.1	**62.2**	**62.8**
$Cl \rightarrow Pr$	66.7	68.3	**71.6**	**72.4**	67.5	68.4	63.4	65.9	70.1	70.8	71.0	72.2
$Cl \rightarrow Rw$	74.3	75.3	71.4	72.3	74.8	75.8	72.7	75.5	77.0	77.8	**77.7**	**79.0**
$Pr \rightarrow Ar$	61.2	62.6	59.6	61.0	63.8	64.7	63.8	65.2	**65.8**	**66.7**	64.6	65.4
$Pr \rightarrow Cl$	53.9	54.1	55.7	56.9	58.1	59.0	56.5	58.6	59.1	60.0	**60.0**	**60.8**
$Pr \rightarrow Rw$	79.9	81.1	82.1	83.9	77.7	78.7	80.1	82.4	**82.2**	**84.1**	81.5	82.9
$Rw \rightarrow Ar$	70.0	70.8	66.5	68.2	67.3	68.2	69.3	71.3	70.5	71.3	**70.6**	**71.6**
$Rw \rightarrow Cl$	55.2	55.4	57.8	59.2	55.8	56.7	57.5	59.2	57.8	58.4	**58.8**	**59.6**
$Rw \rightarrow Pr$	78.3	79.4	78.6	80.8	77.7	78.6	79.4	82.2	80.6	81.8	**81.3**	**82.8**
Average	66.4	67.4	66.7	67.9	67.9	68.8	67.1	69.4	68.9	69.6	**70.8**	**71.9**

TABLE V
ACC(OS*) AND ACC(OS) (%) ON *PIE*

Dataset	ATI-λ		OSBP		SCA_SCM		STA		DAOD		OURS	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
$P1 \rightarrow P2$	41.9	44.0	64.2	66.6	60.7	60.9	54.2	55.0	56.5	57.3	**76.4**	**78.1**
$P1 \rightarrow P3$	53.6	56.3	66.4	69.1	65.7	66.0	67.7	68.8	52.2	53.1	**75.7**	**77.4**
$P1 \rightarrow P4$	64.6	67.9	76.2	80.0	79.5	80.3	81.6	83.6	82.4	85.2	**89.6**	**91.6**
$P1 \rightarrow P5$	43.3	45.4	49.1	50.2	45.7	45.3	42.4	41.7	46.1	47.3	**57.2**	**58.0**
$P2 \rightarrow P1$	56.7	59.5	52.9	54.2	63.6	65.2	51.0	51.6	68.1	69.7	**81.6**	**83.9**
$P2 \rightarrow P3$	53.6	56.3	61.5	63.5	66.9	68.5	58.3	59.0	69.9	71.7	**76.5**	**78.3**
$P2 \rightarrow P4$	73.5	77.1	90.4	92.9	91.2	93.6	78.6	80.6	88.2	91.2	**94.0**	**96.4**
$P2 \rightarrow P5$	34.9	36.7	45.1	45.9	45.3	46.0	39.6	39.6	49.4	49.8	**51.8**	**52.6**
$P3 \rightarrow P1$	66.9	68.4	61.3	61.0	75.2	77.3	69.2	70.7	66.6	68.3	**82.7**	**85.0**
$P3 \rightarrow P2$	52.4	55.0	64.1	64.6	68.9	70.7	59.5	61.0	68.5	70.4	**76.0**	**78.0**
$P3 \rightarrow P4$	70.5	74.0	74.7	76.9	**86.6**	**89.1**	77.6	79.8	83.9	87.1	84.9	87.2
$P3 \rightarrow P5$	44.8	47.1	46.3	46.7	59.7	61.0	46.3	46.7	52.3	53.3	**62.8**	**64.2**
$P4 \rightarrow P1$	63.7	66.8	67.2	68.7	85.7	86.9	84.4	86.6	84.4	87.1	**93.1**	**95.4**
$P4 \rightarrow P2$	74.4	78.1	82.2	85.0	90.0	91.3	89.7	92.5	82.4	84.8	**93.9**	**96.2**
$P4 \rightarrow P3$	58.7	61.7	66.9	67.6	**86.0**	**87.1**	81.6	84.4	77.6	80.0	85.1	86.9
$P4 \rightarrow P5$	46.2	48.5	61.7	63.8	63.2	63.6	68.8	71.0	59.9	61.3	**71.3**	**72.7**
$P5 \rightarrow P1$	30.2	23.5	**64.2**	**66.6**	54.3	55.7	61.2	62.6	59.2	60.6	62.8	64.3
$P5 \rightarrow P2$	34.9	36.7	35.4	35.8	48.8	49.7	49.8	50.0	35.0	34.8	**50.2**	**51.1**
$P5 \rightarrow P3$	39.9	41.9	45.1	46.3	58.7	60.0	46.5	46.3	44.6	44.4	**69.2**	**70.8**
$P5 \rightarrow P4$	55.8	58.6	52.2	53.5	71.1	73.0	70.2	71.7	68.6	70.3	**80.2**	**82.4**
Average	53.0	55.2	61.4	62.9	68.3	69.6	63.9	65.2	64.8	66.4	**75.8**	**77.5**

between known classes and unknown classes. However, the classifier is easily affected by hyperparameter t, which means that the classifier cannot recognize the target data well. For example, for OSBP, in *Digits*, the accuracy of classifying unknown data is significantly higher than known classes, but the opposite situation is apparent in *Office-31*, which proves that this method is not robust. For SCA_SCM, it cannot recognize unknown data well. Especially on the task $D \rightarrow A$ of *Office-31*, SCA_SCM fails to recognize unknown data. That is because OS* is greater than OS by 8.6%.

3) STA separates known data and unknown data by a multibinary classifier. It can achieve a good performance in known classes, but it cannot cope with unknown classes well, especially for the tasks with a large domain gap.

TABLE VI
EXPERIMENTS ON *ImageNet-Caltech*

Dataset	$I \rightarrow C$		$C \rightarrow I$		Avg	
	OS	OS*	OS	OS*	OS	OS*
OSBP	78.6	79.5	75.4	76.3	77.0	77.9
STA	77.9	78.5	74.2	75.0	76.1	76.8
OURS	**80.7**	**81.6**	**76.6**	**77.4**	**78.7**	**79.5**

Compared with baselines, the proposed risk estimator, ϵ-open set difference, can help effectively estimate the risk of the classifier on unknown data. As a result, a clear boundary between known classes and unknown classes can be found. Moreover, our method leverages a novel open set conditional adversarial strategy to capture the multimodal structure of distributions, which can be used to align distribution adequately. Better recognizing unknown data and better aligning

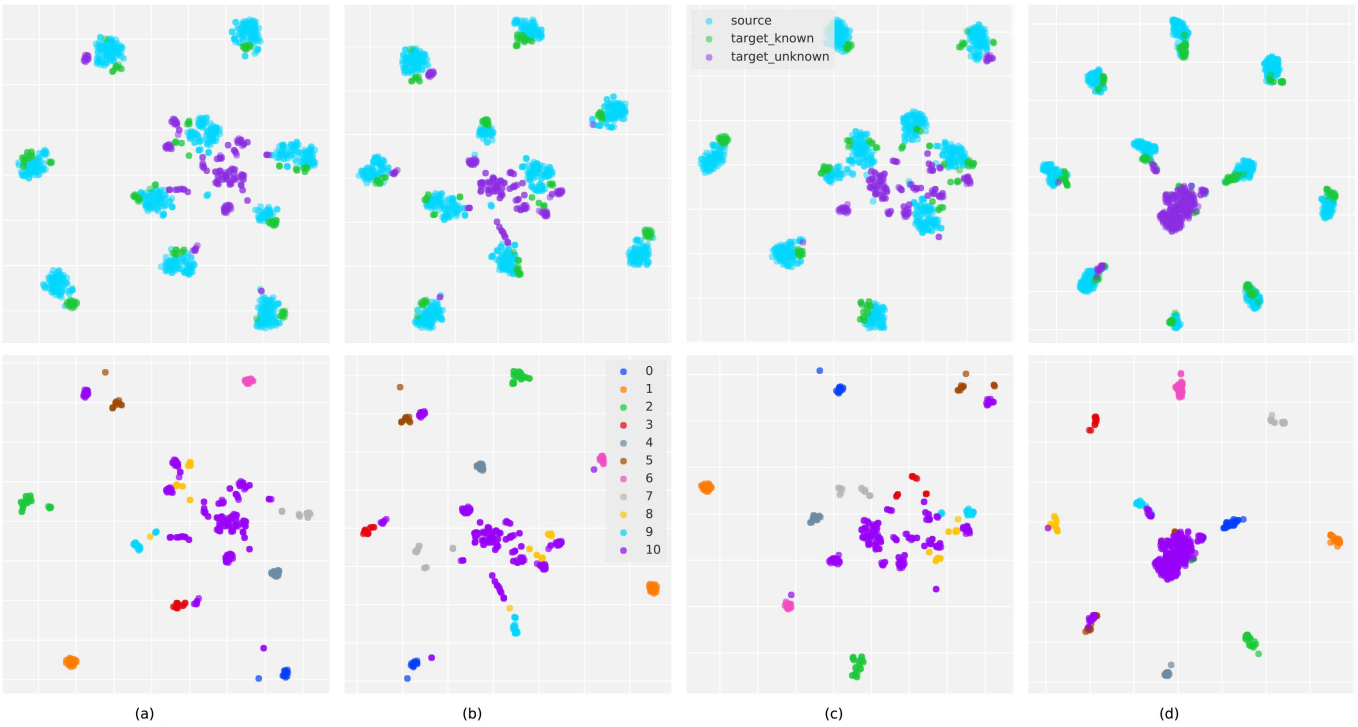

Fig. 4. Feature visualization on $A \rightarrow D$. **First row**: visualization of target and source features. *Blue points* indicate source samples. *Green points* indicate target known samples. *Purple points* indicate target unknown samples. **Second row**: visualization of target samples only. (a) OSBP. (b) SCA_SCM. (c) STA. (d) OURS.

TABLE VII
ABLATION STUDY ON *Office-31*

Dataset	$A \rightarrow D$		$A \rightarrow W$		$D \rightarrow A$		$D \rightarrow W$		$W \rightarrow A$		$W \rightarrow D$		Avg	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
BADA	89.2	90.2	88.1	89.2	82.5	84.3	96.1	96.6	81.3	81.5	96.8	97.0	89.0	89.8
BADA+Δ	92.7	93.3	89.8	90.6	81.6	81.7	98.0	99.5	78.9	83.6	98.5	**100.0**	89.9	91.4
BADA+c	92.2	94.1	87.6	89.0	81.5	84.1	97.7	**100.0**	80.3	83.4	97.3	**100.0**	89.5	91.8
BADA+Δ+c	94.1	94.6	89.2	89.7	83.2	83.4	**98.5**	**100.0**	79.5	83.3	**98.6**	**100.0**	90.5	91.8
BADA+Δ_ϵ	95.5	97.0	**92.6**	**94.0**	82.3	82.6	98.0	99.5	81.9	83.4	98.4	**100.0**	91.5	92.8
OURS	**96.0**	**97.5**	92.5	93.7	**85.3**	**86.0**	98.4	**100.0**	**83.2**	**83.9**	**98.6**	**100.0**	**92.3**	**93.5**

distribution make our method achieve an excellent performance, which is the reason why we can outperform all baselines on 4 benchmark datasets.

F. Analysis

1) Ablation Study: It is necessary to conduct the ablation experiments to demonstrate the effect of each part of our method. Since our method is based on BADA [19], we introduce open set difference (Δ), ϵ-open set difference ($\Delta\epsilon$), and CDAN (c) into BADA and construct ablation experiments as follows: 1) BADA; 2) BADA + Δ; 3) BADA + c; 4) BADA + Δ + c; 5) BADA + Δ_ϵ; and 6) OURS (i.e., BADA + Δ_ϵ + c). The results of ablation experiments are shown in Table VII.

From Table VII, the following facts can be verified: 1) by comparing BADA, BADA + c, and BADA + Δ_ϵ, the accuracy of BADA is the lowest, which proves that Δ_ϵ and CDAN are all useful for UOSDA; 2) the results of BADA + Δ_ϵ and OURS are higher than BADA + Δ and BADA + Δ + c, respectively, which adequately indicates that Δ_ϵ can overcome the issue caused by Δ. The method with Δ_ϵ can establish a boundary between known and unknown classes, preventing the negative transfer caused by unknown classes during the process of distribution alignment; and 3) the accuracies of BADA + Δ + c and OURS are higher than those of BADA + Δ and BADA + Δ_ϵ, respectively, which proves that the novel CDAN effectively elevates the performance of our method.

2) Visualization: To intuitively demonstrate the effect of our method, we visualize the 2-D features of the source and the target by t-distributed stochastic neighbor embedding (t-SNE) [58], which is an effective dimensionality reduction method. Fig. 4 shows the effect of DA of baselines and our method. Clearly, our method outperforms baselines in separating unknown data and aligning the distributions of two domains.

From the first row of Fig. 4, OSBP, SCA_SCM, and STA cannot adequately align distributions of the source and target domains, which is because they cannot distinguish unknown data from known data. As a result, the distribution of unknown classes gets closer to the distribution of known classes. However, our method, in Fig. 4(d), can effectively recognize unknown data and make the distribution of unknown classes far from the distribution of known classes. The second row shows the feature distribution of the target data only. Compared with baselines, it is clear that our method can effectively recognize unknown data and align distributions.

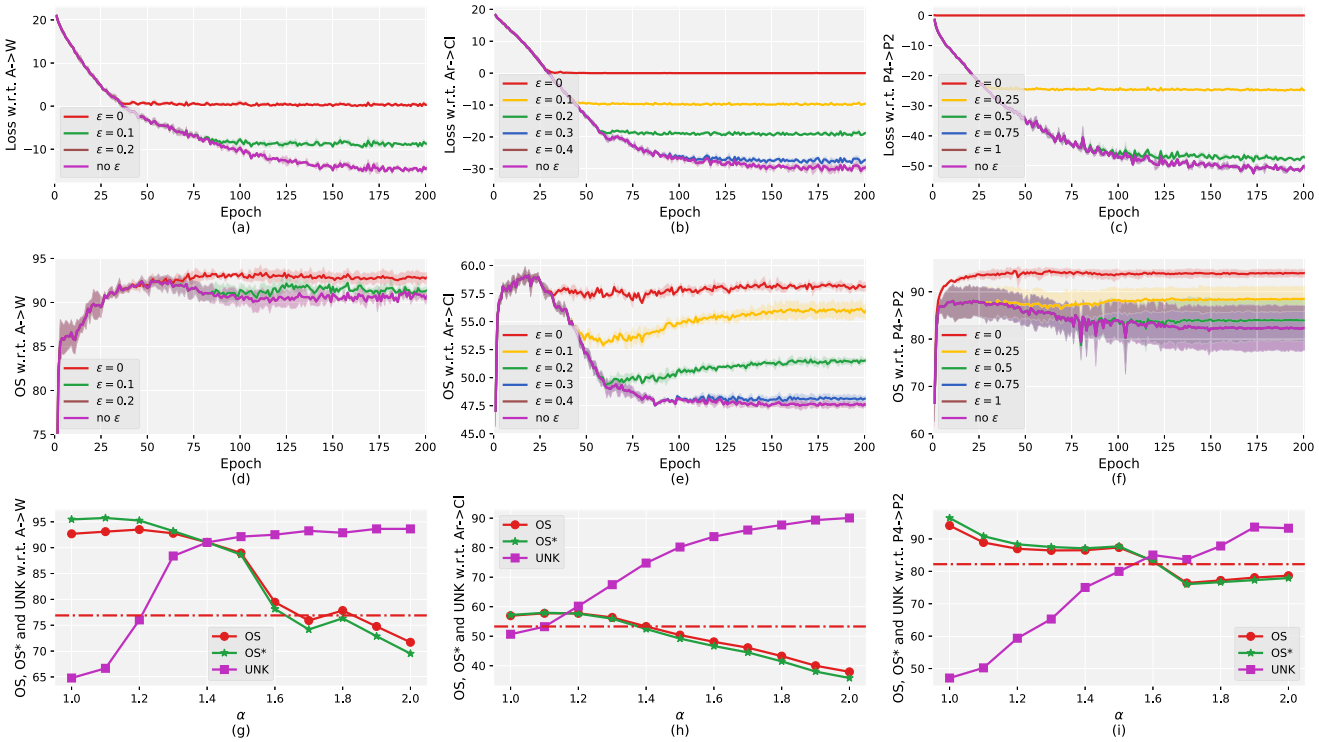

Fig. 5. Parameter analyses w.r.t. ϵ and α. Experiments are conducted on $A \rightarrow W$ of *Office-31* (first column), $Ar \rightarrow Cl$ of *Office-Home* (second column), and $P4 \rightarrow P2$ (third column). First row: The value of Δ_ϵ or Δ ("no ϵ" indicates Δ). Second row: The accuracy of OS w.r.t. Δ_ϵ and Δ when ϵ changes. Third row: (g)–(i) The accuracy of OS, OS*, and UNK w.r.t. α. The losses in (a)–(c) are the values of Δ or Δ_ϵ. It is worth noting that: The line of "ϵ=0.2" coincides with the line of "no ϵ" in (a) and (d). Line of "ϵ=0.4" coincides with the line of "no ϵ" in (b) and (e). Lines of "ϵ=0.75" and "ϵ=1" coincide with the line of "no ϵ" in (c) and (f).

3) Parameter Analysis: In this article, there are two critical parameters in ϵ-open set difference: ϵ and α. Theoretically, ϵ is a variable related to distributional discrepancy. According to (18), distributional discrepancy is greater than the negative open set difference. We hope that the distributional discrepancy is close to 0. Thus, an intuitive thought is to set ϵ as 0. Moreover, α is equal to $1 - \pi^t_{K+1}$, but π^t_{K+1} is an unknown value and is hard to estimate in a batch for a deep method. Therefore, we conduct related experiments to demonstrate the effect of these two parameters for our method. Fig. 5 shows the effect of parameters on $A \rightarrow W$ of *Office-31* (first column), $Ar \rightarrow Cl$ of *Office-Home* (second column), and $P4 \rightarrow P2$ (third column).

The influence of ϵ is shown in the first and second rows. On the task $A \rightarrow W$, the accuracy of OS decreases with the increase in ϵ, which is because ϵ is related to distribution alignment. The bigger ϵ means the smaller value of the lower bound of distributional discrepancy. However, it is worth noting that the value of OS does not change, which is because the domain gap between A and W is small. Therefore, the effect of Δ_ϵ is the same as Δ when ϵ greater than a constant. In Fig. 5(a) and (d), the line of ϵ equal to 0.2 coincides with the line of "no ϵ" (i.e., Δ).

On task $Ar \rightarrow Cl$, there is a large domain gap. In the same way as task $A \rightarrow W$, the bigger the ϵ, the smaller the OS. The line of "$\epsilon = 0.4$" coincides with the line of Δ, which also indicates that the distributional discrepancy of *Office-Home* is larger than *Office-31*. On task $P4 \rightarrow P2$, the domain gap is also large. These lines "$\epsilon = 0.75$" and "$\epsilon = 1.0$" coincide with the line of Δ. It is worth noting that the increasing tendency of the yellow line and the green line after the turning point in Fig. 5(e) owes to the effect of Δ_ϵ and it prevents the problem caused by open set difference.

The effect of parameter α is shown in the third row. The dashed line denotes the accuracy of the OSBP. From Fig. 5(c), (f), and (i), we can conclude that: When we choose a large α, the classifier tends to recognize data as unknown, which leads to the increase in accuracy on unknown classes (UNK[2]) and the decrease in accuracy on known classes (OS*). When we choose a small α, the classifier tends to distinguish data as known data. That is why the classifier achieves a good performance on OS* and a bad performance on UNK. From Fig. 5(g), it is easy to observe that our method can outperform the best baselines when $\alpha \in [1.0, 1.6]$. So we recommend to set α in the range of $[1.0, 1.6]$ on *Office-31*. Similarly, the recommendation parameter range is $[1.0, 1.4]$ and $[1.0, 1.6]$ on *Office-Home* and *PIE*, respectively. Thus, we recommend to set α in the range of $[1.0, 1.4]$.

VII. Conclusion and Future Work

In this article, we tackled a challenging problem called UOSDA. We proposed a practical theoretical bound for UOSDA, which contains an effective risk estimator (Δ_ϵ) to evaluate the risk on data with unknown classes. Furthermore, we proposed a DNN-based UOSDA method under the guidance of the proposed theoretical bound. The method can

[2]UNK is a metric to evaluate the accuracy on unknown target data [19].

accurately estimate the risk of the classifier on data with unknown classes via Δ_ϵ and adequately align the distributions of data with known classes via a novel open set conditional adversarial training strategy. Experiments on several datasets demonstrated that our method significantly outperforms the state-of-the-art UOSDA methods.

In the future, we aim to investigate a more challenging problem called universal DA [59], which contains unknown classes in both the source and target domains. This setting is a more general one and includes UCSDA, UOSDA, and partial DA [56].

REFERENCES

[1] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl.-Based Syst.*, vol. 80, pp. 14–23, May 2015.

[2] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5715–5725.

[3] H. Zuo, J. Lu, G. Zhang, and F. Liu, "Fuzzy transfer learning using an infinite Gaussian mixture model and active learning," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 2, pp. 291–303, Feb. 2019.

[4] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Granular fuzzy regression domain adaptation in Takagi–Sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 847–858, Apr. 2018.

[5] L. Pereira and R. da Silva Torres, "Semi-supervised transfer subspace for domain adaptation," *Pattern Recognit.*, vol. 75, pp. 235–249, Mar. 2018.

[6] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8050–8058.

[7] H. Zuo, J. Lu, Y. Z. Jiang, G. Q. Zhang, and W. Pedrycz, "Fuzzy rule-based domain adaptation in homogeneous and heterogeneous spaces," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 2, pp. 348–361, Feb. 2019.

[8] S. Zhao *et al.*, "Multi-source distilling domain adaptation," in *Proc. AAAI*, 2020, vol. 34, no. 7, pp. 12975–12983.

[9] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1406–1415.

[10] F. Liu, G. Zhang, and J. Lu, "Multi-source heterogeneous unsupervised domain adaptation via fuzzy-relation neural networks," *IEEE Trans. Fuzzy Syst.*, early access, Aug. 20, 2020, doi: 10.1109/TFUZZ.2020.3018191.

[11] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 999–1006.

[12] M. Kan, J. Wu, S. Shan, and X. Chen, "Domain adaptation for face recognition: Targetize source domain bridged by common subspace," *Int. J. Comput. Vis.*, vol. 109, no. 1, pp. 94–109, 2014.

[13] Z. Fang, J. Lu, F. Liu, and G. Zhang, "Unsupervised domain adaptation with sphere retracting transformation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[14] Q. Zhang *et al.*, "A cross-domain recommender system with consistent information transfer," *Decision Support Syst.*, vol. 104, pp. 49–63, Dec. 2017.

[15] W. Lu, Y. Yu, Y. Chang, Z. Wang, C. Li, and B. Yuan, "A dual input-aware factorization machine for CTR prediction," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1–7.

[16] F. Liu, G. Zhang, and J. Lu, "Heterogeneous domain adaptation: An unsupervised approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5588–5602, Dec. 2020.

[17] V. M. K. Peddinti and P. Chintalapoodi, "Domain adaptation in sentiment analysis of Twitter," in *Proc. AAAI*, 2011, pp. 44–49.

[18] P. P. Busto and J. Gall, "Open set domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 754–763.

[19] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, "Open set domain adaptation by backpropagation," in *Proc. ECCV*, 2018, pp. 153–168.

[20] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang, "Open set domain adaptation: Theoretical bound and algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4309–4322, Oct. 2021.

[21] F. Liu, G. Zhang, and J. Lu, "Heterogeneous unsupervised domain adaptation based on fuzzy feature fusion," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2017, pp. 1–6.

[22] Y. Zhang, F. Liu, Z. Fang, B. Yuan, G. Zhang, and J. Lu, "Clarinet: A one-step approach towards budget-friendly unsupervised domain adaptation," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2526–2532.

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, 2014, pp. 3320–3328.

[24] F. Liu, J. Lu, and G. Zhang, "Unsupervised heterogeneous domain adaptation via shared fuzzy equivalence relations," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3555–3568, Dec. 2018.

[25] Q. Feng, G. Kang, H. Fan, and Y. Yang, "Attract or distract: Exploit the margin of open set," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7990–7999.

[26] H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, "Separate to adapt: Open set domain adaptation via progressive separation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2927–2936.

[27] L. Song, J. Huang, A. Smola, and K. Fukumizu, "Hilbert space embeddings of conditional distributions with applications to dynamical systems," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 961–968.

[28] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. NIPS*, 2018, pp. 1640–1650.

[29] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. NIPS*, 2007, pp. 137–144.

[30] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[31] 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.

[32] F. Liu, W. Xu, J. Lu, G. Zhang, A. Gretton, and D. J. Sutherland, "Learning deep kernels for non-parametric two-sample tests," in *Proc. ICML*, 2020, pp. 6316–6326.

[33] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.

[34] J. Wang, Y. Chen, H. Yu, M. Huang, and Q. Yang, "Easy transfer learning by exploiting intra-domain structures," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1210–1215.

[35] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI*, 2016, pp. 2058–2065.

[36] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 402–410.

[37] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, 2015, pp. 97–105.

[38] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI*, 2018, pp. 4058–4063.

[39] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.

[40] Z. Fang, J. Lu, A. Liu, F. Liu, and G. Zhang, "Learning bounds for open-set learning," in *Proc. ICML*, vol. 139, 2021, pp. 3122–3132.

[41] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *Proc. ECCV*. Zürich, Switzerland: Springer, 2014, pp. 393–409.

[42] P. R. M. Júnior *et al.*, "Nearest neighbors distance ratio open-set classifier," *Mach. Learn.*, vol. 106, no. 3, pp. 359–386, 2017.

[43] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1563–1572.

[44] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. ICML*, 2019, pp. 7404–7413.

[45] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Proc. COLT*, 2009.

[46] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[47] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS*, 2011.

[48] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[49] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*. Heraklion, Greece: Springer, 2010, pp. 213–226.

[50] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5018–5027.
[51] A. Rasouli, I. Kotseruba, T. Kunic, and J. Tsotsos, "PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6262–6271.
[52] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[53] R. Shu, H. Bui, H. Narui, and S. Ermon, "A DIRT-t approach to unsupervised domain adaptation," in *Proc. ICLR*, 2018.
[54] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[56] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *Proc. ECCV*, 2018, pp. 135–150.
[57] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2985–2994.
[58] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[59] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2720–2729.

Li Zhong received the B.Sc. degree in automation from the School of Electronic Control and Engineering, Changan University, Xi'an, China, in 2018. He is currently pursuing the M.Sc. degree in control engineering with the Faculty of Information Science and Technology, Tsinghua University, Shenzhen, China.

He is a Member with the Intelligent Computing Laboratory, Tsinghua University. His research interests include transfer learning and domain adaptation.

Zhen Fang (Member, IEEE) received the B.Sc. degree in pure mathematics from Lanzhou University, Lanzhou, China, in 2014, the M.Sc. degree in pure mathematics from Xiamen University, Xiamen, China, in 2017, and the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia, in 2021.

He is a Post-Doctoral Research Associate with the Faculty of Engineering and Information Technology, Australian Artificial Intelligence Institute, University of Technology Sydney. He is a Member with the Decision Systems and e-Service Intelligence (DeSI) Research Laboratory, CAI, University of Technology Sydney. He has published several articles related to domain adaptation and learning theory in International Joint Conference on Neural Networks (IJCNN), Association for the Advancement of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), International Conference on Machine Learning (ICML), and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS). His research interests include transfer learning, domain adaptation, and learning theory.

Feng Liu (Member, IEEE) received the B.Sc. degree in mathematics and the M.Sc. degree in probability and statistics from the School of Mathematics and Statistics, Lanzhou University, Lanzhou, China, in 2013 and 2015, respectively, and the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia, in 2020.

He is a Lecturer with the Faculty of Engineering and Information Technology, Australian Artificial Intelligence Institute, University of Technology Sydney. His research interests include hypothesis testing and trustworthy machine learning.

Dr. Liu has served as a Senior Program Committee Member for ECAI and a Program Committee Member for Conference on Neural Information Processing Systems (NeurIPS), International Conference on Machine Learning (ICML), International Conference on Artificial Intelligence and Statistics (AISTATS), International Conference on Learning Representations (ICLR), Association for the Advancement of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), and IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). He was a recipient of the UTS Research Publication Award (2018), the UTS-FEIT HDR Research Excellence Award (2019), the Best Student Paper Award of FUZZ-IEEE (2019), and the Outstanding Reviewer Award of ICLR (2021). He also served as a Reviewer for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), the IEEE TRANSACTIONS ON FUZZY SYSTEMS (TFS), and the IEEE TRANSACTIONS ON CYBERNETICS (TCYB).

Bo Yuan (Senior Member, IEEE) received the B.E. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 1998, and the M.Sc. and Ph.D. degrees in computer science from The University of Queensland (UQ), Saint Lucia, QLD, Australia, in 2002 and 2006, respectively.

From 2006 to 2007, he was a Research Officer on a project funded by the Australian Research Council, UQ. He is currently an Associate Professor with the Division of Informatics, Shenzhen International Graduate School, Tsinghua University, Shenzhen, China, and a Member with the Intelligent Computing Laboratory, Tsinghua University. He has authored or coauthored more than 100 refereed research articles in data mining, evolutionary computation, and GPU computing.

Guangquan Zhang received the Ph.D. degree in applied mathematics from the Curtin University of Technology, Bentley, WA, Australia, in 2001.

He is a Professor and the Director with the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. He has authored four monographs, five textbooks, and 350 articles including 160 refereed international journal articles. His research interests include fuzzy machine learning, fuzzy optimization, and machine learning and data analytics.

Dr. Zhang has won seven Australian Research Council (ARC) Discovery Project grants and many other research grants. He was a recipient of the ARC QEII Fellowship in 2005. He has served as a member of the editorial boards of several international journals, as a Guest Editor of eight special issues for the IEEE TRANSACTIONS and other international journals, and has co-chaired several international conferences and workshops in the area of fuzzy decision-making and knowledge engineering.

Jie Lu (Fellow, IEEE) received the Ph.D. degree from the Curtin University of Technology, Bentley, WA, Australia, in 2000.

She is a Distinguished Professor and the Director with the Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW, Australia. She has published six research books and over 450 articles in refereed journals and conference proceedings and has won more than 20 ARC Laureate, ARC Discovery projects, and government and industry projects. She has delivered more than 25 keynote speeches at international conferences and chaired 15 international conferences. Her main research interests are in the areas of fuzzy transfer learning, concept drift, decision support systems, and recommender systems.

Dr. Lu is an IFSA fellow and Australian Laureate fellow. She was a recipient of the UTS Medal for Research and Teaching Integration (2010), the Computer Journal Wilkes Award (2018), the UTS Medal for Research Excellence (2019), the IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award (2019), and the Australian Most Innovative Engineer Award (2019). She serves as an Editor-In-Chief for *Knowledge-Based Systems* (Elsevier) and *International Journal of Computational Intelligence Systems*.