Deep Subdomain Adaptation Network for Image Classification

Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, *Fellow, IEEE*, and Qing He

***Abstract*— For a target task where the labeled data are unavailable, domain adaptation can transfer a learner from a different source domain. Previous deep domain adaptation methods mainly learn a global domain shift, i.e., align the global source and target distributions without considering the relationships between two subdomains within the same category of different domains, leading to unsatisfying transfer learning performance without capturing the fine-grained information. Recently, more and more researchers pay attention to subdomain adaptation that focuses on accurately aligning the distributions of the relevant subdomains. However, most of them are adversarial methods that contain several loss functions and converge slowly. Based on this, we present a deep subdomain adaptation network (DSAN) that learns a transfer network by aligning the relevant subdomain distributions of domain-specific layer activations across different domains based on a local maximum mean discrepancy (LMMD). Our DSAN is very simple but effective, which does not need adversarial training and converges fast. The adaptation can be achieved easily with most feedforward network models by extending them with LMMD loss, which can be trained efficiently via backpropagation. Experiments demonstrate that DSAN can achieve remarkable results on both object recognition tasks and digit classification tasks. Our code will be available at https://github.com/easezyc/deep-transfer-learning.**

***Index Terms*— Domain adaptation, fine grained, subdomain.**

I. Introduction

IN RECENT years, deep learning methods have achieved impressive success in computer vision [1], which, however, usually needs large amounts of labeled data to train a good deep network. In the real world, it is often expensive and laborsome to collect enough labeled data. For a target task with the shortage of labeled data, there is a strong motivation to build effective learners that can leverage rich labeled data from a related source domain. However, this learning paradigm suffers from the shift of data distributions across different domains, which will undermine the generalization ability of machine learning models [2], [3].

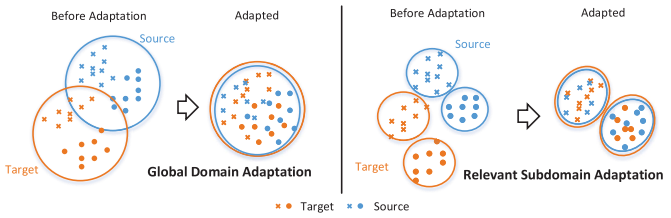


Fig. 1. Left: global domain adaptation might lose some fine-grained information. Right: relevant subdomain adaptation can exploit the local affinity to capture the fine-grained information for each category.

Learning a discriminative model in the presence of the shift between the training and test data distributions is known as domain adaptation or transfer learning [2], [4], [5]. Previous shallow domain adaptation methods bridge the source and target domains by learning invariant feature representations [6]–[8] or estimate instance importance without using target labels [9]. Recent studies have shown that deep networks can learn more transferable features for domain adaptation [10], [11], by disentangling explanatory factors of variations behind domains. The latest advantages have been achieved by embedding domain adaptation modules in the pipeline of deep feature learning to extract domain-invariant representations [12]–[16].

The previous deep domain adaptation methods [13], [16], [17] mainly learn a global domain shift, i.e., aligning the global source and target distributions without considering the relationships between two subdomains in both domains (a subdomain contains the samples within the same class.). As a result, not only all the data from the source and target domains will be confused, but also the discriminative structures can be mixed up. This might loss the fine-grained information for each category. An intuitive example is shown in Fig. 1 (left). After global domain adaptation, the distributions of the two domains are approximately the same, but the data in different subdomains are too close to be classified accurately. This is a common problem in previous global domain

Manuscript received August 13, 2019; revised December 11, 2019; accepted April 13, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1004300, in part by the National Natural Science Foundation of China under Grant U1836206, Grant U1811461, and Grant 61773361, and in part by the Project of Youth Innovation Promotion Association CAS under Grant 2017146. *(Corresponding author: Fuzhen Zhuang.)*

Yongchun Zhu, Fuzhen Zhuang, and Qing He are with the Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhuyongchun18s@ict.ac.cn; zhuangfuzhen@ict.ac.cn; heqing@ict.ac.cn).

Jindong Wang, Guolin Ke, and Jiang Bian are with Microsoft Research, Beijing, China (e-mail: jindong.wang@microsoft.com; guolin.ke@microsoft.com; jiang.bian@microsoft.com).

Jingwu Chen is with ByteDance (e-mail: chenjingwu@bytedance.com).

Hui Xiong is with Rutgers, The State University of New Jersey, New Brunswick, NJ USA (e-mail: hxiong@rutgers.edu).

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

Digital Object Identifier 10.1109/TNNLS.2020.2988928

adaptation methods. Hence, matching the global source and target domains may not work well for diverse scenarios.

With regard to the challenge of global domain shift, recently, more and more researchers [14], [15], [18]–[21] pay attention to subdomain adaptation (also called semantic alignment or matching conditional distribution) which is centered on learning a local domain shift, i.e., accurately aligning the distribution of the relevant subdomains within the same category in the source and target domains. An intuitive example is shown in Fig. 1 (right). After subdomain adaptation, with the local distribution that is approximately the same, the global distribution is also approximately the same. However, all of them are adversarial methods that contain several loss functions and converge slowly. We list the comparison of the subdomain adaptation methods in Experiment IV.

Based on the subdomain adaptation, we propose a deep subdomain adaptation network (DSAN) to align the relevant subdomain distributions of activations in multiple domain-specific layers across domains for unsupervised domain adaptation. DSAN extends the feature representation ability of deep adaptation networks (DANs) by aligning relevant subdomain distributions as mentioned earlier. A key improvement over previous domain adaptation methods is the capability of subdomain adaptation to capture the fine-grained information for each category, which can be trained in an end-to-end framework. To enable proper alignment, we design a local maximum mean discrepancy (LMMD), which measures the Hilbert–Schmidt norm between kernel mean embedding of empirical distributions of the relevant subdomains in source and target domains with considering the weight of different samples. The LMMD method can be achieved with most feedforward network models and can be trained efficiently using standard backpropagation. In addition, our DSAN is very simple and easy to implement. Note that the most remarkable results are achieved by adversarial methods recently. Experiments show that DSAN, which is a nonadversarial method, can obtain the remarkable results for standard domain adaptation on both object recognition tasks and digit classification tasks.

The contributions of this article are summarized as follows.

1) We propose a novel deep neural network architecture for subdomain adaptation, which can extend the ability of DANs by capturing the fine-grained information for each category.
2) We show that DSAN, which is a nonadversarial method, can achieve the remarkable results. In addition, our DSAN is very simple and easy to implement.
3) We propose LMMD to measure the discrepancy between kernel mean embedding relevant subdomains in source and target domains and successfully apply it to DSAN.
4) A new local distribution discrepancy measure $d_{\mathcal{A}_L}$ is proposed to estimate the discrepancy between two subdomain distributions.

II. Related Work

In this section, we will introduce the related work in three aspects: domain adaptation, maximum mean discrepancy (MMD), and subdomain adaptation methods.

1) Domain Adaptation: Recent years have witnessed many approaches to solve the visual domain adaptation problem, which is also commonly framed as the visual data set bias problem [2], [3]. Previous shallow methods for domain adaptation include reweighting the training data so that they can more closely reflect those in the test distribution [22], and finding a transformation in a lower dimensional manifold that draws the source and target subspaces closer [6]–[8], [23], [24].

Recent studies have shown that deep networks can learn more transferable features for domain adaptation [10], [11], by disentangling explanatory factors of variations behind domains. The latest advances have been achieved by embedding domain adaptation modules in the pipeline of deep feature learning to extract domain-invariant representations [12]–[16], [25]. Two main approaches are identified among the literature. The first is statistic moment matching-based approach, i.e., MMD [13], [26], [27], central moment discrepancy (CMD) [28], and second-order statistics matching [16]. The second commonly used approach is based on an adversarial loss, which encourages samples from different domains to be nondiscriminative with respect to domain labels, i.e., domain adversarial net-based adaptation methods [17], [29], [30] borrowing the idea of GAN. Generally, the adversarial approaches can achieve better performance than the statistic moment matching-based approaches. In addition, most state-of-the-art approaches [14], [29], [31] are domain adversarial net-based adaptation methods. Our DSAN is an MMD-based method. We show that DSAN without adversarial loss can achieve remarkable results.

2) Maximum Mean Discrepancy: MMD has been adopted in many approaches [8], [13], [27] for domain adaptation. In addition, there are some extensions of MMD [7], [26]. Conditional MMD [7] and joint MMD [26] measure the Hilbert–Schmidt norm between kernel mean embedding of empirical conditional and joint distributions of the source and target data, respectively. Weighted MMD [32] alleviates the class weight bias by assigning class-specific weights to source data. However, our local MMD measures the discrepancy between kernel mean embedding relevant subdomains in source and target domains with considering the weight of different samples. CMMD [7], [23], [33] is a special case of our LMMD.

3) Subdomain Adaptation: Recently, we have witnessed considerable interest and research [14], [15], [18], [20] for subdomain adaptation that focuses on accurately aligning the distributions of the relevant subdomains. Multiadversarial domain adaptation (MADA) [15] captures the multimode structures to enable fine-grained alignment of different data distributions based on multiple-domain discriminators. Moving semantic transfer network (MSTN) [20] learns the semantic representations for unlabeled target samples by aligning labeled source centroid and pseudolabeled target centroid. CDAN [14] conditions the adversarial adaptation models on discriminative information conveyed in the classifier predictions. Co-DA [18] constructs multiple diverse feature spaces and aligns source and target distributions in each of them individually while encouraging that alignments agree with each other with regard

to the class predictions on the unlabeled target examples. The adversarial loss is adopted by all of them. However, compared DSAN with them, our DSAN that is more simple and easy to implement can achieve better performance without adversarial loss.

III. Deep Subdomain Adaptation Network

In unsupervised domain adaptation, we are given a source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ of n_s labeled samples ($\mathbf{y}_i^s \in \mathbb{R}^C$ is an one-hot vector indicating the label of $\mathbf{x}_i^s$, i.e., $y_{ij}^s = 1$ means $\mathbf{x}_i^s$ belonging to the jth class, where C is the number of classes.) and a target domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ of n_t unlabeled samples. $\mathcal{D}_s$ and $\mathcal{D}_t$ are sampled from different data distributions p and q, respectively, and $p \neq q$. The goal of deep domain adaptation is to design a deep neural network $\mathbf{y} = f(\mathbf{x})$ that formally reduces the shifts in the distributions of the relevant subdomains in different domains and learns transferable representations simultaneously such that the target risk $R_t(f) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\ q}[f(\mathbf{x}) \neq \mathbf{y}]$ can be bounded by leveraging the source domain supervised data.

Recent studies reveal that deep networks [34] can learn more transferable representations than traditional handcrafted features [11], [35]. The favorable transferability of deep features leads to several popular deep transfer learning methods [12], [13], [26], [36], which mainly use adaptation layers with a global domain adaptation loss to jointly learn a representation. The formal representation can be

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda \hat{d}(p, q) \tag{1}$$

where $J(\cdot, \cdot)$ is the cross-entropy loss function (classification loss) and $\hat{d}(\cdot, \cdot)$ is domain adaptation loss. $\lambda > 0$ is the tradeoff parameter of the domain adaptation loss and the classification loss.

The common problem with these methods is that they mainly focus on aligning the global source and target distributions without considering the relationships between subdomains within the same category of different domains. These methods derive a global domain shift for the source and target domains, and the global distribution of the two domains is approximately the same after adaptation. However, the global alignment may lead to some irrelevant data too close to be classified accurately. Actually, while by exploiting the relationships between the subdomains in different domains, just aligning the relevant subdomain distributions can not only match the global distributions but also the local distributions mentioned earlier. Therefore, subdomain adaptation that exploits the relationships between two subdomains to overcome the limitation of aligning global distributions is necessary.

To divide the source and target domains into multiple subdomains that contain the samples within the same class, the relationships between the samples should be exploited. It is well known that the samples within the same category are more relevant. However, data in the target domain is unlabeled. Hence, we would use the output of the networks as the pseudolabels of target domain data, which will be detailed later. According to the category, we divide $\mathcal{D}_s$ and $\mathcal{D}_t$ into C subdomains $\mathcal{D}_s^{(c)}$ and $\mathcal{D}_t^{(c)}$ where $c \in \{1, 2, \ldots, C\}$ denotes the class label, and the distributions of $\mathcal{D}_s^{(c)}$ and $\mathcal{D}_t^{(c)}$ are $p^{(c)}$ and $q^{(c)}$, respectively. The aim of subdomain adaptation is to align the distributions of relevant subdomains that have samples with the same label. Combining the classification loss and subdomain adaptation loss, the loss of subdomain adaptation method is formulated as

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(\mathbf{x}_i^s), \mathbf{y}_i^s) + \lambda \mathbf{E}_c[\hat{d}(p^{(c)}, q^{(c)})] \tag{2}$$

where $\mathbf{E}_c[\cdot]$ is the mathematical expectation of the class. To compute the discrepancy in 2 between the relevant subdomain distributions based on MMD [37] that is a nonparametric measure, we propose LMMD to estimate the distribution discrepancy between subdomains.

A. Maximum Mean Discrepancy

MMD [37] is a kernel two-sample test, which rejects or accepts the null hypothesis $p = q$ based on the observed samples. The basic idea behind MMD is that if the generating distributions are identical, all the statistics are the same. Formally, MMD defines the following difference measure:

$$d_{\mathcal{H}}(p, q) \triangleq \|\mathbf{E}_p[\phi(\mathbf{x}^s)] - \mathbf{E}_q[\phi(\mathbf{x}^t)]\|_{\mathcal{H}}^2 \tag{3}$$

where $\mathcal{H}$ is the reproducing kernel Hillbert space (RKHS) endowed with a characteristic kernel k. Here, $\phi(\cdot)$ denotes some feature map to map the original samples to RKHS and the kernel k means $k(\mathbf{x}^s, \mathbf{x}^t) = \langle \phi(\mathbf{x}^s), \phi(\mathbf{x}^t) \rangle$, where $\langle \cdot, \cdot \rangle$ represents inner product of vectors. The main theoretical result is that $p = q$ if and only if $D_{\mathcal{H}}(p, q) = 0$ [37]. In practice, an estimate of the MMD compares the square distance between the empirical kernel mean embeddings as

$$\begin{aligned} \hat{d}_{\mathcal{H}}(p, q) &= \left\| \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \phi(\mathbf{x}_i) - \frac{1}{n_t} \sum_{\mathbf{x}_j \in \mathcal{D}_t} \phi(\mathbf{x}_j) \right\|_{\mathcal{H}}^2 \\ &= \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k\left(\mathbf{x}_i^s, \mathbf{x}_j^s\right) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k\left(\mathbf{x}_i^t, \mathbf{x}_j^t\right) \\ &\quad - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k\left(\mathbf{x}_i^s, \mathbf{x}_j^t\right) \end{aligned} \tag{4}$$

where $\hat{d}_{\mathcal{H}}(p, q)$ is an unbiased estimator of $d_{\mathcal{H}}(p, q)$.

B. Local Maximum Mean Discrepancy

As a nonparametric distance estimate between two distributions, MMD has been widely applied to measure the discrepancy between the source and target distributions. Previous deep MMD-based methods [13], [26], [38] mainly focus on the alignment of the global distributions, ignoring the relationships between two subdomains within the same category. Taking the relationships of the relevant subdomains into consideration, it is important to align the distributions of the relevant subdomains within the same category in source

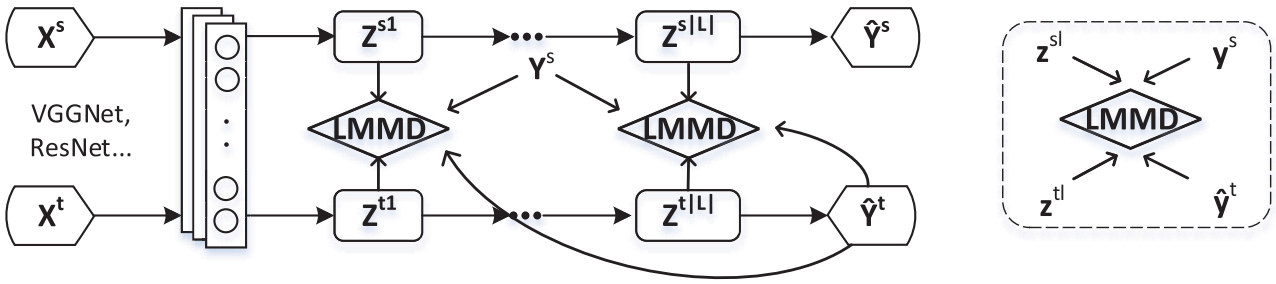


Fig. 2. Left: architecture of DSAN. DSAN will formally reduce the discrepancy between the relevant subdomain distributions of the activations in layers L by using LMMD minimization. Right: LMMD module needs four inputs: the activations $\mathbf{z}^{sl}$ and $\mathbf{z}^{tl}$ where $l \in L$, the ground-truth label $\mathbf{y}^s$, and the predicted label $\hat{\mathbf{y}}^t$.

and target domains. With the desire to align distributions of the relevant subdomains, we propose the LMMD as

$$d_{\mathcal{H}}(p,q) \triangleq \mathbf{E}_c \|\mathbf{E}_{p^{(c)}}[\phi(\mathbf{x}^s)] - \mathbf{E}_{q^{(c)}}[\phi(\mathbf{x}^t)]\|^2_{\mathcal{H}} \tag{5}$$

where $\mathbf{x}^s$ and $\mathbf{x}^t$ are the instances in $\mathcal{D}_s$ and $\mathcal{D}_t$, and $p^{(c)}$ and $q^{(c)}$ are the distributions of $\mathcal{D}_s^{(c)}$ and $\mathcal{D}_t^{(c)}$, respectively. Different from MMD that focuses on the discrepancy of global distributions, 5 can measure the discrepancy of local distributions. By minimizing 5 in deep networks, the distributions of relevant subdomains within the same category are drawn close. Therefore, the fine-grained information is exploited for domain adaptation.

We assume that each sample belongs to each class according to weight w^c. Then, we formulate an unbiased estimator of 5 as

$$\hat{d}_{\mathcal{H}}(p,q) = \frac{1}{C}\sum_{c=1}^{C}\left\| \sum_{\mathbf{x}_i^s \in \mathcal{D}_s} w_i^{sc}\phi(\mathbf{x}_i^s) - \sum_{\mathbf{x}_j^t \in \mathcal{D}_t} w_j^{tc}\phi(\mathbf{x}_j^t) \right\|^2_{\mathcal{H}} \tag{6}$$

where w_i^{sc} and w_j^{tc} denote the weight of $\mathbf{x}_i^s$ and $\mathbf{x}_j^t$ belonging to class c, respectively. Note that $\sum_{i=1}^{n_s} w_i^{sc}$ and $\sum_{j=1}^{n_t} w_j^{tc}$ are both equal to one, and $\sum_{\mathbf{x}_i \in \mathcal{D}} w_i^c \phi(\mathbf{x}_i)$ is a weighted sum on category c. We compute w_i^c for the sample $\mathbf{x}_i$ as

$$w_i^c = \frac{y_{ic}}{\sum_{(\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}} y_{jc}} \tag{7}$$

where y_{ic} is the cth entry of vector $\mathbf{y}_i$. For samples in the source domain, we use the true label $\mathbf{y}_i^s$ as a one-hot vector to compute w_i^{sc} for each sample. However, in unsupervised adaptation where the target domain has no labeled data, we can not calculate 6 directly with the $\mathbf{y}_j^t$ unavailable. We find that the output of the deep neural network $\hat{\mathbf{y}}_i = f(\mathbf{x}_i)$ is a probability distribution that well characterizes the probability of assigning $\mathbf{x}_i$ to each of the C classes. Thus, for target domain $\mathcal{D}_t$ without labels, it is a natural idea to use $\hat{\mathbf{y}}_i^t$ as the probability of assigning $\mathbf{x}_i^t$ to each of the C classes. Then, we can calculate w_j^{tc} for each target sample. Finally, we can calculate 6.

It is easy to access the labels of the source domain, while for the target domain, the label predicted (hard prediction) by the model might be wrong, and using this wrong label might degrade the performance. Hence, using the probability prediction (soft prediction) might alleviate the negative impact. Note that CMMD, which assumes that each sample has the same weight, is a special case of LMMD, whereas LMMD takes the uncertainty of target samples into consideration.

To adapt feature layers, we need the activations in the layers. Given source domain $\mathcal{D}_s$ with n_s labeled instances and target domain $\mathcal{D}_t$ with n_t unlabeled points drawn independent identically distributed (i.i.d.) from p and q, respectively, the deep networks will generate activations in layers l as $\{\mathbf{z}_i^{sl}\}_{i=1}^{n_s}$ and $\{\mathbf{z}_j^{tl}\}_{j=1}^{n_t}$. In addition, we cannot compute the $\phi(\cdot)$ directly. Then, we reformulate 6 as

$$\begin{aligned}\hat{d}_l(p,q) = \frac{1}{C}\sum_{c=1}^{C}\Bigg[&\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} w_i^{sc} w_j^{sc} k\left(\mathbf{z}_i^{sl}, \mathbf{z}_j^{sl}\right) \\ &+ \sum_{i=1}^{n_t}\sum_{j=1}^{n_t} w_i^{tc} w_j^{tc} k\left(\mathbf{z}_i^{tl}, \mathbf{z}_j^{tl}\right) \\ &- 2\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} w_i^{sc} w_j^{tc} k\left(\mathbf{z}_i^{sl}, \mathbf{z}_j^{tl}\right)\Bigg]\end{aligned} \tag{8}$$

where $\mathbf{z}^l$ is the lth ($l \in L = \{1, 2, \ldots, |L|\}$) layer activation. Equation 8 can be used as the adaptation loss in 2 directly, and the LMMD can be achieved with most feedforward network models.

C. Deep Subdomain Adaptation Network

Based on LMMD, we propose DSAN as shown in Fig. 2. Different from previous global adaptation methods, DSAN not only aligns the global source and target distributions but also aligns the distributions of the relevant subdomains by integrating deep feature learning and feature adaptation in an end-to-end deep learning model. We try to reduce the discrepancy between the relevant subdomain distributions of the activations in layers L. We use the LMMD in (8) over the domain-specific layers L as the subdomain adaptation loss in the following equation:

$$\min_f \frac{1}{n_s}\sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda \sum_{l \in L} \hat{d}_l(p,q). \tag{9}$$

Since training deep CNNs requires a large amount of labeled data that are prohibitive for many domain adaptation applications, we start with the CNN models pretrained on the ImageNet 2012 data and fine-tune it as [26]. The training of DSAN mainly follows standard minibatch stochastic gradient

descent (SGD) algorithm. It is worth noting that, with DSAN iteration, the labeling for target samples usually becomes more accurate. This EM-like pseudolabel refinement procedure is empirically effective, as shown in the experiments.

Remark: The theory of domain adaptation [39], [40] suggests $\mathcal{A}$-distance as a measure of distribution discrepancy, which, together with the source risk, will bound the target risk. The proxy $\mathcal{A}$-distance is defined as $d_{\mathcal{A}} = 2(1 - 2\epsilon)$, where ϵ is the generalization error of a classifier (e.g., kernel SVM) trained on the binary problem of discriminating the source and target. The $\mathcal{A}$-distance just focuses on the global distribution discrepancy; hence, we propose the $\mathcal{A}_L$-distance to estimate the subdomain distribution discrepancy. First, we define $d_{\mathcal{A}}$ of class c as $d_{\mathcal{A}_c} = 2(1 - 2\epsilon^c)$, where ϵ^c is the generalization error of a classifier trained on the same class in different domains. Then, we define $d_{\mathcal{A}_L} = \mathbf{E}[d_{\mathcal{A}_c}] = 2\mathbf{E}[1 - 2\epsilon^c] = 2\sum_{c=1}^{C} p(c)(1 - 2\epsilon^c)$, where $\mathbf{E}[\cdot]$ denotes the mathematical expectation and $p(c)$ denotes the probability of class c in the target domain.

D. Theoretical Analysis

In this section, we give an analysis of the effectiveness of using the classifier predictions on the target samples, making use of the theory of domain adaptation [39], [41].

Theorem 1: Let $\mathcal{H}$ be the hypothesis space. Given two domains $\mathcal{S}$ and $\mathcal{T}$, we have

$$\forall h \in \mathcal{H}, R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + C \tag{10}$$

where $R_{\mathcal{S}}(h)$ and $R_{\mathcal{T}}(h)$ are the expected error on the source samples and target samples, respectively. $R_{\mathcal{S}}(h)$ can be minimized easily with source label information. Besides, $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$ is the domain divergence measure by a discrepancy distance between two distributions $\mathcal{S}$ and $\mathcal{T}$. Actually, there are many approaches to minimize $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$, such as adversarial learning [12], MMD [13], and Coral [16]. C is the shared expected loss and is expected to be negligibly small, thus usually disregarded by previous methods [12], [13]. However, it is possible that C tends to be large when the cross-domain category alignment is not explicitly enforced. Hence, C needs to be bounded as well. Unfortunately, we cannot directly measure C without target true labels. Therefore, we utilize the pseudolabels to give the approximate evaluation and minimization.

Definition 1: C is defined as

$$C = \min_{h \in \mathcal{H}} R_{\mathcal{S}}(h, f_{\mathcal{S}}) + R_{\mathcal{T}}(h, f_{\mathcal{T}}) \tag{11}$$

where $f_{\mathcal{S}}$ and $f_{\mathcal{T}}$ are true labeling functions for source and target domain, respectively.

We show our DSAN is trying to optimize the upper bound for C. From [39], for any labeling functions f_1, f_2, and f_3, we have

$$R(f_1, f_2) \leq R(f_1, f_3) + R(f_2, f_3). \tag{12}$$

Then, we have

$$\begin{aligned} C &= \min_{h \in \mathcal{H}} R_{\mathcal{S}}(h, f_{\mathcal{S}}) + R_{\mathcal{T}}(h, f_{\mathcal{T}}) \\ &\leq \min_{h \in \mathcal{H}} R_{\mathcal{S}}(h, f_{\mathcal{S}}) + R_{\mathcal{T}}(h, f_{\mathcal{S}}) + R_{\mathcal{T}}(f_{\mathcal{S}}, f_{\mathcal{T}}) \\ &\leq \min_{h \in \mathcal{H}} R_{\mathcal{S}}(h, f_{\mathcal{S}}) + R_{\mathcal{T}}(h, f_{\mathcal{S}}) + R_{\mathcal{T}}(f_{\mathcal{S}}, f_{\hat{\mathcal{T}}}) \\ &\quad + R_{\mathcal{T}}(f_{\mathcal{T}}, f_{\hat{\mathcal{T}}}) \end{aligned} \tag{13}$$

where $f_{\hat{\mathcal{T}}}$ is pseudolabeling function for target domain. The first term $R_{\mathcal{S}}(h, f_{\mathcal{S}})$ and the second term $R_{\mathcal{T}}(h, f_{\mathcal{S}})$ denotes the disagreement between h and the source labeling function $f_{\mathcal{S}}$ on source and target samples, respectively. Since h is learned with the labeled source samples, the gap between them can be very small. The last term $R_{\mathcal{T}}(f_{\mathcal{T}}, f_{\hat{\mathcal{T}}})$ denotes the discrepancy between the ideal target labeling function $f_{\mathcal{T}}$ and the pseudolabeling function $f_{\hat{\mathcal{T}}}$, which would be minimized as learning proceeds. Then, we should focus on the third term $R_{\mathcal{T}}(f_{\mathcal{S}}, f_{\hat{\mathcal{T}}}) = \mathbf{E}_{x \sim \mathcal{T}}[l(f_{\mathcal{S}}(x), f_{\hat{\mathcal{T}}}(x))]$, where $l(\cdot, \cdot)$ is typically 0–1 loss function. The source samples of class k would be predicted with label k by the source labeling function $f_{\mathcal{S}}$. If the feature of target samples in class k is similar with the source feature in class k, the target samples can be predicted the same as the pseudotarget labeling function. Therefore, if the distributions of subdomains in different domain are matching, $R_{\mathcal{T}}(f_{\mathcal{S}}, f_{\hat{\mathcal{T}}})$ is expected to be small.

In summary, by aligning relevant subdomain distributions, our DSAN could further minimize the shared expected loss C. Hence, utilizing the prediction of the target samples is effective for unsupervised domain adaptation.

IV. Experiment

We evaluate DSAN against competitive transfer learning baselines on object recognition and digit classification. The four data sets, including ImageCLEF-DA, Office-31, Office-Home, and VisDA-2017, are used for object recognition task, while for digit classification, we construct the transfer tasks from MNIST, USPS, and SVHN. We denote all transfer tasks as source domain $\rightarrow$ target domain.

A. Setup

ImageCLEF-DA[1] is a benchmark data set for ImageCLEF 2014 domain adaptation challenge, which is organized by selecting 12 common categories shared by the following three public data sets, each is considered as a domain: Caltech-256 (**C**), ImageNet ILSVRC 2012 (**I**), and Pascal VOC 2012 (**P**). There are 50 images in each category and 600 images in each domain. We use all domain combinations and build six transfer tasks: **I** $\rightarrow$ **P**, **P** $\rightarrow$ **I**, **I** $\rightarrow$ **C**, **C** $\rightarrow$ **I**, **C** $\rightarrow$ **P**, **P** $\rightarrow$ **C**.

Office-31 [43] is a benchmark data set for domain adaptation, comprising 4110 images in 31 classes collected from three distinct domains: Amazon (**A**), which contains images downloaded from amazon.com, and Webcam (**W**) and DSLR (**D**), which contain images taken by Web camera and digital SLR camera with different photographical settings,

[1] http://imageclef.org/2014/adaptation

respectively. To enable unbiased evaluation, we evaluate all methods on all six transfer tasks **A** → **W**, **D** → **W**, **W** → **D**, **A** → **D**, **D** → **A**, **W** → **A** as in [12], [26], and [38].

Office-Home [44] is a new data set, which consists of 15 588 images and is much larger than Office-31 and ImageCLEF-DA. It consists of images from four different domains: artistic images (**A**), clip art (**C**), product images (**P**), and real-world images (**R**). For each domain, the data set contains the images of 65 object categories collected in office and home settings. Similarly, we use all domain combinations and construct 12 transfer tasks.

VisDA-2017 [45] is a challenging simulation-to-real data set, with two very distinct domains: synthetic, renderings of 3-D models from different angles and with different lightning conditions, and real, natural images. It contains over 280k images across 12 classes in the training, validation, and test domains.

MNIST-USPS-SVHN: We explore three digit data sets: MNIST [46], USPS, and SVHN [47] for transfer digit classification. Different from Office-31, MNIST contains gray digit images of size 28×28, USPS contains 16×16 gray digits, and SVHN contains color 32×32 digits images that might contain more than one digit in each image. We conduct experiments on three transfer tasks MNIST → USPS, USPS → MNIST, and SVHN → MNIST.

Baseline Methods: For ImageCLEF-DA and Office-31, we compare our model DSAN with several standard deep learning methods and deep transfer learning methods: deep convolutional neural network (ResNet) [1], deep domain confusion (DDC) [38], DAN [13], Deep CORAL (D-CORAL) [16], domain adversarial neural networks (DANNs) [17], residual transfer network (RTN) [26], adversarial discriminative domain adaptation (ADDA) [30], joint adaptation networks (JANs) [26], MADA [15], collaborative and adversarial network (CAN and iCAN) [31], generate to adapt (GTA) [42], and conditional adversarial domain adaptation (CDAN and CDAN+E) [14]. For Office-Home, we compare DSAN with ResNet, DAN, DANN, JAN, and CDAN, and the results of all baselines are extracted from [14] and [15]. For VisDA-2017, we compare DSAN with ResNet, DANN, DAN, JAN, and MCD [48], and the results of all baselines are extracted from [49].

For MNIST-USPS-SVHN, we compare DSAN with DANNs [17], deep reconstruction classification networks (DRCNs) [50], coupled generative adversarial networks (CoGANs) [51], ADDA [30], unsupervised image-to-image translation networks (UNIT) [], asymmetric tritraining domain adaptation (ATDA) [53], GTA [42], and MSTN [20]. The results of SourceOnly, DANN, DRCN, CoGAN, ADDA, and GTA are extracted from [42]. For the rest, we refer to the results in their original articles.

Implementation Details For object recognition tasks, we employed the ResNet [1]. Following CDAN [14], a bottleneck layer fcb with 256 units is added after the last average pooling layer for safe transfer representation learning. We use the output of fcb as inputs to the LMMD. Note that, it is easy to add LMMD in multiple layers and we only add LMMD to one layer. Also, image random flipping and cropping are adopted following JAN [26]. For a fair comparison, all baselines use the same architecture (for VisDA-2017, we use ResNet101 [1], whereas ResNet50 for others). We fine-tune all convolutional and pooling layers from ImageNet pretrained models and train the classifier layer via back-propagation. Since the classifier is trained from scratch, we set its learning rate to be ten times that of the other layers. For digit classification tasks, we follow the protocols in ADDA [30] and use the same architecture with ADDA.

For all tasks, we use minibatch SGD with a momentum of 0.9 and the learning rate annealing strategy in Revgrad [12]; the learning rate is not selected by a grid search due to high computational cost, it is adjusted during SGD using the following formula: $\eta_\theta = \eta_0/(1+\alpha\theta)^\beta$, where θ is the training progress linearly changing from 0 to 1, $\eta_0 = 0.01$, $\alpha = 10$, and $\beta = 0.75$. To suppress noisy activations at the early stages of training, instead of fixing the adaptation factor λ, we gradually change it from 0 to 1 by a progressive schedule: $\lambda_\theta = 2/\exp(-\gamma\theta) - 1$, and $\gamma = 10$ is fixed throughout the experiments [12].

We implement DSAN in PyTorch and report the average classification accuracy and standard error of three random trials. For all MMD-based methods [13], [26], [38] including DSAN, we adopt Gaussian kernel with bandwidth set to median pairwise squared distances on the training data [37].

B. Results

1) Object Recognition: The classification results of ImageCLEF-DA, Office-31, Office-Home, and VisDA-2017 are, respectively, shown in Tables I–IV. DSAN outperforms all compared methods on most transfer tasks. In particular, DSAN substantially improves the average accuracy by large margins (more than 3%) on Image-CLEF, Office-Home, and VisDA-2017. The encouraging results indicate the importance of subdomain adaptation and show that DSAN is able to learn more transferable representations.

The experimental results further reveal several insightful observations.

1) In standard domain adaptation, subdomain adaptation methods (MADA [15], CDAN [14], and our DSAN) outperform previous global domain adaptation methods. The improvement from previous global domain adaptation methods to subdomain adaptation methods is crucial for domain adaptation; previous methods align global distribution without considering the relationship between subdomains, whereas DSAN accurately aligns the relevant subdomain distributions, which can capture more fine-grained information for each category.
2) In particular, comparing DSAN with the most recent subdomain adaptation methods [14], [15], DSAN achieves better performance. This verifies the effectiveness of our model.
3) Comparing DSAN with the nonadversarial methods [13], [16], [26], [38], DSAN also largely improves the average performance on 24 object recognition tasks (6.65% higher than JAN [26]).

TABLE I

ACCURACY (%) ON IMAGECLEF-DA FOR UNSUPERVISED DOMAIN ADAPTATION (RESNET50)

Method	I → P	P → I	I → C	C → I	C → P	P → C	Avg
ResNet [1]	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DDC [38]	74.6±0.3	85.7±0.8	91.1±0.3	82.3±0.7	68.3±0.4	88.8±0.2	81.8
DAN [13]	75.0±0.4	86.2±0.2	93.3±0.2	84.1±0.4	69.8±0.4	91.3±0.4	83.3
DANN [17]	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
D-CORAL [16]	76.9±0.2	88.5±0.3	93.6±0.3	86.8±0.6	74.0±0.3	91.6±0.3	85.2
JAN [26]	76.8±0.4	88.0±0.2	94.7±0.2	89.5±0.3	74.2±0.3	91.7±0.3	85.8
MADA [15]	75.0±0.3	87.9±0.2	96.0±0.3	88.8±0.3	75.2±0.2	92.2±0.3	85.8
CAN [31]	78.2	87.5	94.2	89.5	75.8	89.2	85.7
iCAN [31]	79.5	89.7	94.7	89.9	78.5	92.0	87.4
CDAN [14]	76.7±0.3	90.6±0.3	97.0±0.4	90.5±0.4	74.5±0.3	93.5±0.4	87.1
CDAN+E [14]	77.7±0.3	90.7±0.2	**97.7**±0.3	91.3±0.3	74.2±0.2	94.3±0.3	87.7
DSAN	**80.2**±0.2	**93.3**±0.4	97.2±0.2	**93.8**±0.2	**80.8**±0.4	**95.9**±0.4	**90.2**

TABLE II

ACCURACY (%) ON OFFICE-31 FOR UNSUPERVISED DOMAIN ADAPTATION (RESNET50)

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet [1]	68.4±0.5	96.7±0.5	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DDC [38]	75.8±0.2	95.0±0.2	98.2±0.1	77.5±0.3	67.4±0.4	64.0±0.5	79.7
DAN [13]	83.8±0.4	96.8±0.2	99.5±0.1	78.4±0.2	66.7±0.3	62.7±0.2	81.3
D-CORAL [16]	77.7±0.3	97.6±0.2	99.7±0.1	81.1±0.4	64.6±0.3	64.0±0.4	80.8
DANN [17]	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
ADDA [30]	86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN [26]	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
MADA [15]	90.0±0.1	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
GTA [42]	89.5±0.5	97.9±0.3	99.8±0.4	87.7±0.5	72.8±0.3	71.4±0.4	86.6
CAN [31]	81.5	98.2	99.7	85.5	65.9	63.4	82.4
iCAN [31]	92.5	98.8	**100.0**	90.1	72.1	69.9	87.2
CDAN [14]	93.1±0.2	98.2±0.2	**100.0**±.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
CDAN+E [14]	**94.1**±0.1	**98.6**±0.1	**100.0**±.0	**92.9**±0.2	71.0±0.3	69.3±0.3	87.7
DSAN	93.6±0.2	98.3±0.1	**100.0**±0.0	90.2±0.7	**73.5**±0.5	**74.8**±0.4	**88.4**

TABLE III

ACCURACY (%) ON OFFICE-HOME FOR UNSUPERVISED DOMAIN ADAPTATION (RESNET50)

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet [1]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [13]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [17]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [26]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [14]	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN+E [14]	50.7	70.6	**76.0**	57.6	**70.0**	**70.0**	57.4	50.9	77.3	70.9	56.7	81.6	65.8
DSAN	**54.4**	**70.8**	75.4	**60.4**	67.8	68.0	**62.6**	**55.9**	**78.5**	**73.8**	**60.6**	**83.1**	**67.6**

TABLE IV

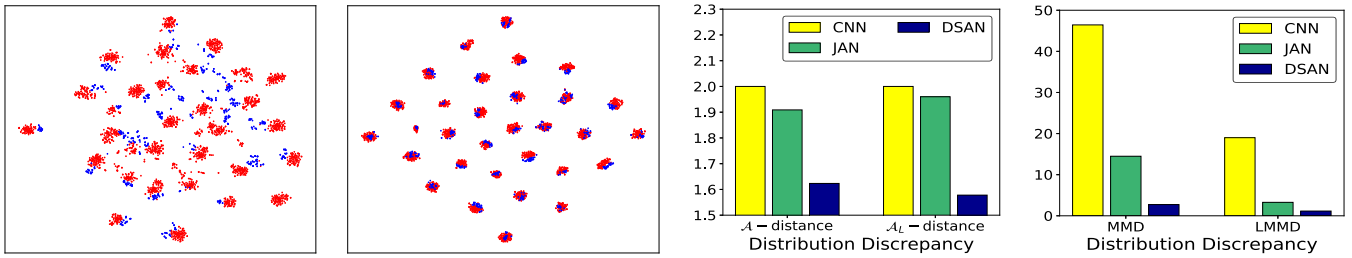
ACCURACY (%) ON VISDA-2017 FOR UNSUPERVISED DOMAIN ADAPTATION (RESNET101)

Method	airplane	bicycle	bus	car	horse	knife	motorcycle	person	plant	skateboard	train	truck	Avg
ResNet [1]	72.3	6.1	63.4	**91.7**	52.7	7.9	80.1	5.6	90.1	18.5	78.1	25.9	49.4
DANN [17]	81.9	**77.7**	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [13]	68.1	15.4	76.5	87.0	71.1	48.9	82.3	51.5	88.7	33.2	88.9	42.2	62.8
JAN [26]	75.7	18.7	82.3	86.3	70.2	56.9	80.5	53.8	92.5	32.2	84.5	**54.5**	65.7
MCD [48]	87.0	60.9	**83.7**	64.0	**88.9**	**79.6**	84.7	**76.9**	88.6	40.3	83.0	25.8	71.9
DSAN	**90.9**	66.9	75.7	62.4	**88.9**	77.0	**93.7**	75.1	**92.8**	**67.6**	**89.1**	39.4	**75.1**

4) Comparing DSAN with LMMD, DAN with MMD, and JAN with JMMD, DSAN achieves the best performance, which implies that LMMD is more suitable for aligning distributions than MMD and JMMD.

2) Digit Classification: The classification results of three tasks of MNIST–USPS–SVHN are shown in Table V. Except for DRCN [50], all other baselines are adversarial ones. DSAN largely outperforms all baselines except SVHN → MNIST task. Comparing DSAN with MSTN [20] which is also a subdomain adaptation method, DSAN achieves better average accuracy and more stable results with lower standard error.

Overall, all the abovementioned results demonstrate the effectiveness of the proposed model.

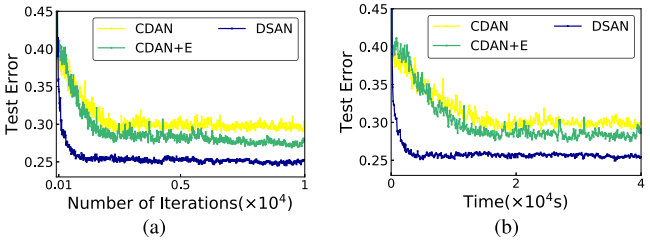


(a) (b) (c) (d)

Fig. 3. (a) and (b) Visualizations of the learned representations using t-SNE for JAN and DSAN on task A→W, respectively. Red points are source samples and blue are target samples. (c) $\mathcal{A}$-distance and $\mathcal{A}_L$-distance on task A→W. (d) MMD and (e) LMMD on task A→W.

TABLE V
ACCURACY (%) ON DIGIT RECOGNITION TASKS FOR UNSUPERVISED DOMAIN ADAPTATION. ("−" MEANS THAT WE DID NOT FIND THE RESULT ON THE TASK)

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST
SourceOnly	75.2±0.16	57.1±0.17	60.1±0.11
DANN [17]	77.1±1.8	73.0±2.0	73.91±0.07
DRCN [50]	91.8±0.09	73.7±0.04	82.0±0.16
CoGAN [51]	91.2±0.8	89.1±0.8	-
ADDA [30]	89.4±0.2	90.1±0.8	76.0±1.8
UNIT [52]	95.97	93.58	90.53
ATDA [53]	93.17	84.14	85.8
GTA [42]	92.8±0.9	90.8±1.3	**92.4**±0.9
MSTN [20]	92.9±1.1	-	91.7±1.5
DSAN	**96.9**±0.2	**95.3**±0.1	90.1±0.4



(a) (b)

Fig. 4. On task D → A (Office31), we further analyze the convergence. (a) Convergence (iteration). (b) Convergence (time).

C. Analysis

1) Feature Visualization: We visualize in Fig. 3(a) and (b) the network activations of task **A** → **W** learned by JAN and DSAN (both use Gaussian kernel) using t-SNE embeddings [10]. Red points are source samples and blue are target samples. Fig. 3(a) shows the result for JAN [26], which is a typical statistic moment matching-based approach using JMMD. We can find that the source and target domains are not aligned very well and some points are hard to classify. In contrast, Fig. 3(b) shows the representations learned by our DSAN using LMMD. It is observed that the source and target domains are aligned very well. We not only can see that the subdomains in different domains with the same class are very close but also the subdomains with different classes are dispersed. This result suggests that our model DSAN is able to capture more fine-grained information for each category than JAN, and LMMD is more effective than JMMD to align the distributions.

2) Distribution Discrepancy: We use $\mathcal{A}$-distance and $\mathcal{A}_L$-distance mentioned in Section III-C to measure global distribution discrepancy and the subdomain distribution discrepancy. Fig. 3(c) shows $d_{\mathcal{A}}$ and $d_{\mathcal{A}_L}$ on task **A** → **W** with representations of CNN, JAN, and DSAN. We observe that $d_{\mathcal{A}}$ and $d_{\mathcal{A}_L}$ using DSAN are much smaller than the ones using CNN and JAN, which shows that DSAN can not only close the cross-domain gap but also one of relevant subdomains more effectively.

MMD is a method to measure the discrepancy of global distributions, whereas LMMD is a method to measure the discrepancy of local subdomain distributions. We compute MMD and LMMD across domains on task **A** → **W** using CNN, JAN, and DSAN based on the features in pool layer and ground-truth labels. Fig. 3(d) shows that both MMD and LMMD using DSAN activations are much smaller than using CNN and JAN activations, which again validates that DSAN successfully reduces the discrepancy of global and local distributions. In addition, LMMD is smaller than MMD for the reason that LMMD can estimate the distribution discrepancy by eliminating the irrelevant data.

3) Convergence: We testify the convergence of CDAN, CDAN+E, and DSAN, with the test errors on task D → A (Office31) shown in Fig. 4. From Fig. 4(a), with the same number of iterations, DSAN achieves faster convergence than CDAN and CDAN+E. From Fig. 4(b), with the same period, DSAN also converges faster. Besides, the results further reveal that for each iteration, DSAN runs faster than CDAN and CDAN+E.

4) Discussion on the Advantage of DSAN: To give an overview of the results, we further compare our DSAN with several adversarial subdomain adaptation methods [14], [15], [18], [20] in Table VI and find some insightful observations. First, the adversarial subdomain adaptation methods usually have several loss functions, while DSAN only needs one classification loss and one LMMD loss. In addition, DSAN only has one hyperparameter, whereas MSTN [20] and Co-DA [18] have several hyperparameters. DSAN has fewer loss terms and hyperparameter, which also indicates the easy implementation. Second, comparing DSAN with MADA [15] and CDAN [14], DSAN also takes less time to converge. Third, DSAN achieves the best performance. Especially, DSAN achieves 3% accuracy higher than CDAN [14] which is one of the most recent subdomain adaptation methods. Overall, all the results again validate the advantage of our model DSAN.

TABLE VI

COMPARISON OF THE SUBDOMAIN ADAPTATION METHODS. K IN MADA MEANS THE NUMBER OF CLASSES. PARAMETER MEANS THE NUMBER OF HYPERPARAMETERS IN THE METHODS. TIME MEANS THE AVERAGE CONVERGENCE TIME ON THE IMAGECLEF-DA DATA SET (SECONDS). TIME IS MEASURED ON A GEFORCE GTX 1080 TI GPU BY OURSELVES. ACCURACY MEANS THE AVERAGE ACCURACY ON THE IMAGECLEF-DA DATA SET. "−" MEANS THAT WE DOSE NOT FIND THE RESULTS FROM THE ORIGINAL ARTICLE

Method	MADA	MSTN	CDAN	Co-DA	**DSAN**
Adversarial	Yes	Yes	Yes	Yes	**No**
Loss terms	$1+K$	3	3	12	**2**
Parameter	1	3	1	6	**1**
Time(s)	4318	-	1944	-	**702**
Accuracy	85.8	-	87.1	-	**90.2**

V. CONCLUSION

Unlike the previous methods that align the global source and target distributions, subdomain adaptation can accurately align the distributions of relevant subdomains within the same category of the source and target domains. However, most recent subdomain adaptation methods are adversarial approaches that contain several loss functions and converge slowly. Based on this, we proposed a new method DSAN, which is a nonadversarial method and very simple and easy to implement. Furthermore, to measure the discrepancy between relevant subdomains within the same category of different domains, we proposed a new local distribution discrepancy measure LMMD. Extensive experiments conducted on both object recognition and digit classification tasks demonstrate the effectiveness of the proposed model.

REFERENCES

[1] 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.

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

[3] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[4] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proc. IJCAI*, 2015, pp. 4119–4125.

[5] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," 2019, *arXiv:1911.02685*. [Online]. Available: http://arxiv.org/abs/1911.02685

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

[7] 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.

[8] 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.

[9] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Proc. NIPS*, 2007, pp. 601–608.

[10] J. Donahue *et al.*, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. ICML*, 2014, pp. 647–655.

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

[12] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

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

[14] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. NIPS*, 2018, pp. 1647–1657.

[15] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. AAAI*, 2018, pp. 3934–3941.

[16] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. ECCV*, 2016, pp. 443–450.

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

[18] A. Kumar *et al.*, "Co-regularized alignment for unsupervised domain adaptation," in *Proc. NIPS*, 2018, pp. 9367–9378.

[19] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu, "Stratified transfer learning for cross-domain activity recognition," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–10.

[20] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. ICML*, 2018, pp. 5419–5428.

[21] 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.

[22] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. ACL*, 2007, pp. 264–271.

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

[24] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, and Q. Yang, "Transfer learning with dynamic distribution adaptation," 2019, *arXiv:1909.08531*. [Online]. Available: http://arxiv.org/abs/1909.08531

[25] Y. Zhu *et al.*, "Multi-representation adaptation network for cross-domain image classification," *Neural Netw.*, vol. 119, pp. 214–221, Nov. 2019.

[26] M. Long, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. ICML*, 2017, pp. 2208–2217.

[27] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 5989–5996.

[28] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," in *Proc. ICLR*, 2017, pp. 1–13.

[29] J. Hoffman *et al.*, "Cycada: Cycle-consistent adversarial domain adaptation," in *Proc. ICML*, 2018, pp. 1–15.

[30] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, p. 4.

[31] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3801–3809.

[32] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2272–2281.

[33] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 1129–1134.

[34] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[35] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724.

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

[37] 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.

[38] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: http://arxiv.org/abs/1412.3474

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

[40] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Proc. NIPS*, 2009, pp. 1041–1048.

[41] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. NIPS*, 2007, pp. 137–144.
[42] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8503–8512.
[43] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*, 2010, pp. 213–226.
[44] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," 2017, *arXiv:1706.07522*. [Online]. Available: http://arxiv.org/abs/1706.07522
[45] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "VisDA: The visual domain adaptation challenge," 2017, *arXiv:1710.06924*. [Online]. Available: http://arxiv.org/abs/1710.06924
[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," *NIPS workshop DLUFL*, 2011.
[48] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.
[49] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4893–4902.
[50] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. ECCV*, 2016, pp. 597–613.
[51] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 469–477.
[52] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. NIPS*, 2017, pp. 700–708.
[53] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proc. ICML*, 2017, pp. 2988–2997.

Yongchun Zhu received the B.S. degree from Beijing Normal University, Beijing, China, in 2018. He is currently pursuing the M.S. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing.

He has published some papers in journals and conference proceedings, including *Neural Networks*, AAAI, WWW, and PAKDD. His main research interests include transfer learning, metalearning, and recommendation systems.

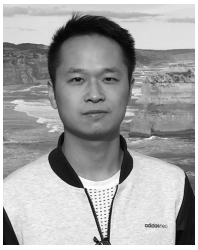
Fuzhen Zhuang is currently an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He has published more than 100 papers in the prestigious refereed journals and conference proceedings, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, ACM TIST, SIGKDD, IJCAI, AAAI, WWW, and ICDE. His research interests include transfer learning, machine learning, data mining, multitask learning, and recommendation systems.

Jindong Wang received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

He is currently a Researcher with Microsoft Research, Beijing. His research interests mainly include transfer learning, machine learning, data mining, and artificial intelligence.

Dr. Wang is also a member of ACM, AAAI, and CCF. He serves as a Reviewer for many journals and conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, CSUR, ACM CHI, and *Neurocomputing*.

Guolin Ke is currently a Senior Researcher with the Machine Learning Group, Microsoft Research Asia, Beijing, China. He created one of the most popular decision tree learning tool LightGBM. His research interest mainly includes machine learning algorithms.



Jingwu Chen received the master's degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2019.

He is currently a member of the Recommendation Team, ByteDance. His research interests include machine learning and its applications, such as recommendation systems and computational advertising.

Jiang Bian is currently a Researcher and Engineer with rich experience in information retrieval, data mining, and machine learning. He is also a Principal Researcher and a Research Manager with Microsoft Research, Beijing, China, with research interests in AI for finance, AI for logistics, deep learning, multiagent reinforcement learning, computational advertising, and a variety of machine learning applications. He was a Senior Scientist with Yidian Inc., China, a start-up company, where he worked on recommendation and search problems.

Hui Xiong (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Minnesota–Twin Cities, Minneapolis, MN, USA, in 2005.

He is currently a Full Professor with Rutgers, The State University of New Jersey, New Brunswick, NJ USA.

Dr. Xiong received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, the RBS Deans Research Professorship in 2016, the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence in 2009, the IEEE ICDM Best Research Paper Award in 2011, and the IEEE ICDM Outstanding Service Award in 2017. He is a Co-Editor-in-Chief of *Encyclopedia of GIS*, an Associate Editor of the IEEE TRANSACTIONS ON BIG DATA, ACM TKDD, and ACM TMIS. He has served regularly on the organization committees of numerous conferences, such as the Program Co-Chair for ACM KDD 2018 (research track), ACM KDD 2012 (industry track), and IEEE ICDM 2013, and the General Co-Chair for IEEE ICDM 2015. He is a Distinguished Scientist of ACM.

Qing He received the B.S. degree in mathematics from Hebei Normal University, Shijiazhang, China, in 1985, the M.S. degree in mathematics from Zhengzhou University, Zhengzhou, China, in 1987, and the Ph.D. degree in fuzzy mathematics and artificial intelligence from Beijing Normal University, Beijing, China, in 2000.

From 1987 to 1997, he was with the Hebei University of Science and Technology, Shijiazhuang. He is currently a Doctoral Tutor with the Institute of Computing and Technology, CAS. He is currently a Professor with the Institute of Computing Technology, Chinese Academy of Science (CAS), Beijing, and the Graduate University of Chinese Academy of Sciences (GUCAS). His interests include data mining, machine learning, classification, fuzzy clustering.