Transfer Learning with Dynamic Adversarial Adaptation Network

Chaohui Yu*†, Jindong Wang‡, Yiqiang Chen*†(✉), Meiyu Huang§
*Beijing Key Lab. of Mobile Computing and Pervasive Device, Inst. of Comp. Tech., Chinese Academy of Sciences, Beijing, China
†University of Chinese Academy of Sciences, Beijing, China
‡Microsoft Research Asia, Beijing, China
§Qian Xuesen Lab. of Space Technology, China Academy of Space Technology, Beijing, China
Email:{yuchaohui17s,yqchen}@ict.ac.cn, jindong.wang@microsoft.com

***Abstract*— The recent advances in deep transfer learning reveal that adversarial learning can be embedded into deep networks to learn more transferable features to reduce the distribution discrepancy between two domains. Existing adversarial domain adaptation methods either learn a single domain discriminator to align the global source and target distributions, or pay attention to align subdomains based on multiple discriminators. However, in real applications, the marginal (global) and conditional (local) distributions between domains are often contributing differently to the adaptation. There is currently no method to dynamically and quantitatively evaluate the relative importance of these two distributions for adversarial learning. In this paper, we propose a novel Dynamic Adversarial Adaptation Network (DAAN) to dynamically learn domain-invariant representations while quantitatively evaluate the relative importance of global and local domain distributions. To the best of our knowledge, DAAN is the first attempt to perform dynamic adversarial distribution adaptation for deep adversarial learning. DAAN is extremely easy to implement and train in real applications. We theoretically analyze the effectiveness of DAAN, and it can also be explained in an attention stragegy. Extensive experiments demonstrate that DAAN achieves better classification accuracy compared to state-of-the-art deep and adversarial methods. Results also imply the necessity and effectiveness of the dynamic distribution adaptation in adversarial transfer learning.**

***Index Terms*—domain adaptation, dynamic, global and local, adversarial learning**

I. Introduction

Deep neural networks have significantly improved the performance of diverse data mining and computer vision applications [1], [2]. In order to avoid overfitting and achieve better performance, a large amount of labeled data is needed to train a deep learning model. Unfortunately, it is often expensive and time-consuming to acquire sufficient labeled data. Thus, a natural idea is to leverage the abundant labeled samples in an existing domain (i.e. *source* domain) to facilitate learning in the domain to be learned (i.e. *target* domain).

A promising approach to solve such cross-domain learning problems is called *transfer learning*, or *domain adaptation* [3]. The key to successful adaptation is to learn a discriminative model to reduce the distribution discrepancy between the two domains. Traditional methods perform adaptation by either reweighting samples from the source domain [4], [5], [6], or seeking an explicit feature transformation that transforms the source and target samples into the same feature space [7], [8], [9], [10], [11]. Recent studies have indicated that deep networks can learn more transferable features for domain adaptation [12], [13]. The latest advances have been achieved by embedding domain adaptation modules in the pipeline of deep feature learning to extract domain-invariant representations [14], [15], [16], [17], [18], [19].

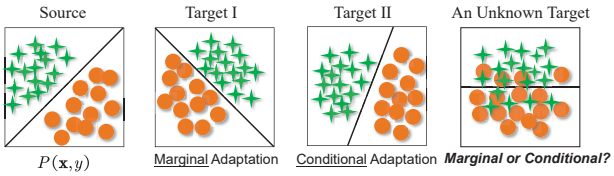


Fig. 1. The different effects of marginal and conditional distributions in transfer learning applications

TABLE I
Comparison between the recent methods on Office-Home [20] dataset (C denotes number of classes)

Method	DANN [18]	JAN [21]	MEDA [7]	DAAN
Dynamic adaptation	No	No	Yes	**Yes**
Hyperparameter	λ	λ	λ, p, η, ρ	λ
Extra classifier	No	No	$C+1$	**No**
Accuracy (%)	57.6	58.3	60.2	**61.8**

Recently, adversarial learning [22] has been successfully embedded into deep networks to reduce distribution discrepancy between the source and target domains. Prior advanced adversarial adaptation methods [18], [23], [24], [25] have shown promising results in several domain adaptation tasks. Most of them either learn a single domain discriminator to align the global source and target distributions, or pay attention to align subdomains based on multiple discriminators. For instance, Domain-adversarial Neural Network (DANN) [18], [26] focuses on the global adversarial learning, while Multi-adversarial Domain Adaptation (MADA) [25] pays attention to the subdomain adaptation by training several domain classifiers. However, in real applications, the marginal (global) and conditional (local) distributions between domains are often contributing differently to the adaptation. For example, when two domains are very dissimilar (source → target I in Fig. 1), the global distribution is more important. When the global

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00088

distributions are close (source → target II in Fig. 1), the local distribution should be given more attention. Two more recent work called Balanced Distribution Adaptation (BDA) [8] and Manifold Embedded Distribution Alignment (MEDA) [7] proposed to adaptively align these two distributions, while it is based on kernel method with high computational cost. In addition, MEDA is incapable of handling large-scale data. To date, there is no effort that could dynamically evaluate the relative importance of the marginal and conditional distributions for adversarial domain adaptation.

In this paper, we propose a novel **Dynamic Adversarial Adaptation Network (DAAN)** for unsupervised domain adaptation. DAAN is able to learn domain-invariant features through end-to-end adversarial training. The key component in DAAN is the *Dynamic Adversarial Factor*, which is capable of easily, dynamically, and quantitatively evaluating the relative importance of the marginal and conditional distributions. The adaptation can be achieved by Stochastic Gradient Descent (SGD) with the gradients computed by backpropagation in linear-time. To the best of our knowledge, DAAN is the *first* adversarial domain adaptation method that is able to dynamically learn the relationship between the marginal and conditional distributions. Extensive experiments demonstrate that DAAN outperforms state-of-the-art methods on standard domain adaptation benchmarks. More importantly, it is shown that there does exist the relative importance of two distributions, of which DAAN could make accurate evaluation.

The contributions of this paper are four-fold:

(1) We propose a novel dynamic adversarial adaptation network to learn domain-invariant features. DAAN is accurate and robust, and can be easily implemented by most deep learning libraries.

(2) We propose the dynamic adversarial factor to easily, dynamically, and quantitatively evaluate the relative importance of the marginal and conditional distributions in adversarial transfer learning.

(3) We theoretically analyze the effectiveness of DAAN, and it can also be explained in an attention stragegy.

(4) Extensive experiments on public datasets demonstrate the significant superiority of our DAAN in both classification accuracy and the estimation of the dynamic adversarial factor.

II. Related Work

A. Unsupervised Domain Adaptation

Unsupervised domain adaptation is a specific area of transfer learning [3], which is to learn a discriminative model in the presence of the domain shift between domains. There are mainly two categories: traditional (shallow) learning and deep learning.

Traditional (shallow) learning methods can mainly be divided into two categories: (1) Subspace learning. Subspace Alignment (SA) [27] aligns the base vectors of both domains and Subspace Distribution Alignment (SDA) [28] extends SA by adding the subspace variance adaptation. CORAL [29] aligns subspaces in second-order statistics. (2) Distribution alignment. Pan *et al.* proposed the Transfer Component Analysis (TCA) [11] to align the marginal distributions between domains. Based on TCA, Joint Distribution Adaptation (JDA) [30] is proposed to match both marginal and conditional distributions. But these works treat the two distributions equally and fail to leverage the different importance of distributions. Recently, Wang *et al.* proposed Balanaced Distribution Adaptation (BDA) [8] and Manifold Embedded Distribution Alignment (MEDA) [7] approaches to dynamically evaluate the different effect of marginal and conditional distributions and achieved the state-of-the-art results on domain adaptation. However, MEDA is based on kernel method and requires to train several linear classifiers in each iteration. DAAN is significantly different from MEDA in two folds as shown in Table I. Firstly, MEDA uses shallow features to learn the adaptive factor, while DAAN uses deep adversarial representations for end-to-end learning. Secondly, MEDA uses extra linear classifiers to learn the adaptive factor, while DAAN directly uses the adversarial features, which is more efficient.

In recent years, deep networks can learn more transferable features for domain adaptation [12], [13], by disentangling explanatory factors of variations behind domains compared to traditional methods. Most work on deep domain adaptation is based on discrepancy measurement. For instance, Correlation Alignment (CORAL) [17], Kullback-Leibler (KL) divergence [31], Maximum Mean Discrepancy (MMD) [32], [21], [33], [34], [15], and Central Moment Discrepancy (CMD) [35] are used to reduce the distribution divergence between domains. However, there is no effective deep learning method that can dynamically align the marginal and conditional distributions.

B. Domain-adversarial Learning

As a special case of deep domain adaptation, domain-adversarial learning has been popular in recent years. In this case, a domain discriminator that classifies whether a data point is drawn from the source or target domain is used to encourage domain confusion through an adversarial objective to minimize the distance between the source and target distributions [18]. Adversarial learning has been explored in Generative Adversarial Networks (GANs) [22]. And Generative Multi-Adversarial Network (GMAN) [36] extends GANs to multiple discriminators including formidable adversary and forgiving teacher, which significantly eases model training.

Recently, we have witnessed considerable research [18], [25], [37], [38] for adversarial domain adaptation. DANN [18] aligns the whole source and target distributions with a global domain discriminator. MADA [25] captures multi-mode structures to enable fine-grained alignment of different data distributions based on multiple domain discriminators. Co-DA [37] constructs multiple diverse feature spaces and aligns source and target distributions in each of them individually. The proposed DAAN is also based on adversarial learning, while it significantly outperforms existing methods by dynamically evaluating the relative importance of the marginal and conditional distributions.

III. Dynamic Adversarial Adaptation Network

In this section, we introduce the proposed Dynamic Adversarial Adaptation Network (DAAN).

A. Problem Definition

In unsupervised domain adaptation, we are given a source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ of n_s labeled examples and a target domain $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ of n_t unlabeled examples. $\mathcal{D}_s$ and $\mathcal{D}_t$ have the same label space, i.e. $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ where d is the dimensionality. The marginal distributions between two domains are different, i.e. $P_s(\mathbf{x}_s) \neq P_t(\mathbf{x}_t)$. The goal of deep UDA is to design a deep neural network that enables learning of transfer classifiers $y = f(\mathbf{x})$ that formally reduces the shifts in the distributions of two domains such that the target risk $\epsilon_t(f) = \mathbb{E}_{(x,y)\sim q}[f(x) \neq y]$ can be bounded by using the source domain while achieving better performance on the target domain.

B. Adversarial Learning for Domain Adaptation

Domain adversarial adaptation methods borrow the idea of GAN [22] to help learn transferable features. The adversarial learning procedure is a two-player game, where the first player is the domain discriminator G_d trained to distinguish the source domain from the target domain, and the second player is the feature extractor G_f that tries to confuse the domain discriminator by extracting domain-invariant features. The two players are trained adversarially: the parameters θ_f of feature extractor G_f are learned by maximizing the loss of domain discriminator G_d, while the parameters θ_d of G_d are trained by minimizing the loss of the domain discriminator. In addition, the loss of the label classifier G_y is also minimized. The loss function can be formalized as:

$$\begin{aligned} L\left(\theta_f, \theta_y, \theta_d\right) =& \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y\left(G_y\left(G_f\left(\mathbf{x}_i\right)\right), y_i\right) \\ & - \frac{\lambda}{n_s + n_t} \sum_{\mathbf{x}_i \in (\mathcal{D}_s \cup \mathcal{D}_t)} L_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right), \end{aligned} \tag{1}$$

where λ is a trade-off parameter and L_y and L_d denote the label classifier loss and domain discriminator loss. Since there are no labels for the target domain, d_i means the domain label of the input samples (d_i for source domain is 0, d_i for target domain is 1). After the training converges, the parameters $\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d$ will deliever a saddle point of Eq. (1):

$$\begin{aligned} \left(\hat{\theta}_f, \hat{\theta}_y\right) &= \arg\min_{\theta_f, \theta_y} L\left(\theta_f, \theta_y, \theta_d\right) \\ \left(\hat{\theta}_d\right) &= \arg\max_{\theta_d} L\left(\theta_f, \theta_y, \theta_d\right) \end{aligned} \tag{2}$$

C. Dynamic Adversarial Adaptation Network

Most state-of-the-art domain adaptation approaches [18], [25], [37], [38] are adversarial learning methods. Competitive results are achieved by either aligning the marginal distributions [32], [26] (Source → Target I in Fig. 1), or aligning the conditional distributions [25] (Source → Target II in Fig. 1). It has been shown that aligning these two distributions together would lead to better performance [21] since both distributions are helpful in learning domain-invariant features. However, in real applications, these two different distributions may have totally different contributions to the domain discrepancy. In real applications, it is extremely challenging to account for the relative importance of these two distributions. Therefore, we need to *dynamically* and *quantitatively* evaluate their importance in domain adaptation.

Recently, a Manifold Embedded Distribution Alignment (MEDA) [7] approach has been proposed to compute the weights of marginal and conditional distributions. MEDA learns a domain-invariant classifier in the Reproducing Kernel Hilbert Space (RKHS) while evaluating the weights of the two distributions using the proxy $\mathcal{A}$-distance [39]. However, MEDA has to train $1+C$ extra linear classifiers in each iteration, which is computationally expensive and time-consuming. Furthermore, MEDA can only be applied to small-scale data since it calculates the pseudo-inverse of all the samples each time thus it cannot be deployed online. To sum up, it is extremely challenging to easily, dynamically, and quantitatively evaluate the relative importance of both distributions, while the system still remains scalable to large-scale data.

In this paper, we make key technical improvements by proposing the *Dynamic Adversarial Adaptation Network (DAAN)* to address the above challenge. As shown in Fig. 2, DAAN is based on the well-established generative adversarial networks (GAN) [22] that aims at learning domain-invariant features via adversarial training. In DAAN, high-level features f are extracted by a feature extractor (G_f, the blue part). Then, the adaptation of marginal and conditional distributions are achieved by the *Global* domain discriminator (G_d, the purple part) and *Local* domain discriminator (G_d^c, the green part), respectively. Most importantly, DAAN proposes a novel *Dynamic Adversarial Factor* (ω, the yellow part) to perform easy, dynamic, and quantitative evaluation of these two distributions. Along with the label classifier (G_y, the orange part), the parameters of DAAN can be trained efficiently with the Gradient Reversal Layer (GRL) [26].

In the next sections, we will first introduce the label classifier, global domain discriminator, and local domain discriminator. Then, the dynamic adversarial factor is presented in Section III-D. Finally, we show the loss function of DAAN and how to train DAAN.

1) Label Classifier: The label classifier (G_y, the orange part in Fig. 2) is trained to discriminate the label of the input samples from the source domain. Thus, the supervised information on $\mathcal{D}_s$ can be utilized. Its training objective is a cross-entropy loss, which can be formulated as:

$$L_y = -\frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \sum_{c=1}^{C} P_{\mathbf{x}_i \to c} \log G_y(G_f(\mathbf{x}_i)), \tag{3}$$

where C is the number of classes, $P_{\mathbf{x}_i \to c}$ is the probability of $\mathbf{x}_i$ belonging to class c, G_y is the label classifier and G_f is the feature extractor.

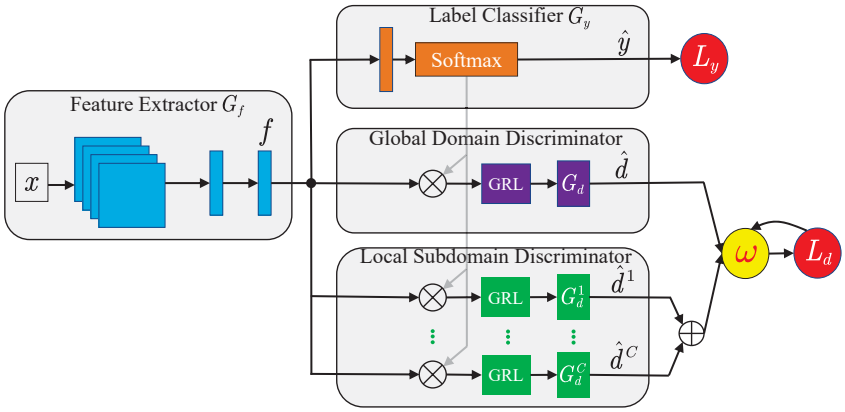


Fig. 2. (Best viewed in color) The architecture of the proposed Dynamic Adversarial Adaptation Network (DAAN). DAAN consists of a deep feature extractor G_f (blue), a label classifier G_y (orange), a global domain discriminator (G_d, purple), C local subdomain discriminators (G_d^c for $c \in \{1, \cdots, C\}$, green), and a dynamic adversarial factor module (ω, yellow). $\oplus$ denotes the plus operator while $\otimes$ is the product operator. f is the extracted deep features, $\hat{y}$ is the predicted label, L_y and L_d are the classification loss and domain loss. $\hat{d}$ and $\hat{d}^c$ are the predicted domain label. GRL stands for Gradient Reversal Layer.

2) Global Domain Discriminator: The global domain discriminator (G_d, the purple part in Fig. 2) is designed to align the marginal (global) distributions between the source and target domains. The general idea of global domain discriminator follows the Domain-adversarial Neural Network (DANN) [26], which has been described in the previous section. In DAAN, we calculate the loss of the global domain discriminator as:

$$L_g = \frac{1}{n_s + n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_d(G_d(G_f(\mathbf{x}_i)), d_i), \tag{4}$$

where L_d is the domain discriminator loss (cross-entropy), G_f is the feature extractor, and d_i is the domain label of the input sample $\mathbf{x}_i$.

3) Local Domain Discriminator: The local domain discriminator (G_d^c, the green part in Fig. 2) is designed to align the conditional (local) distributions between the source and target domains. Compared to the global domain discriminator, local domain discriminator is able to align the multi-mode structure in two distributions, thus it can perform more fine-grained domain adaptation.

To be concrete, the domain discriminator G_d can be split into C *class-wise* domain discriminators G_d^c, each is responsible for matching the source and target domain data associated with class c. The output of the label predictor $G_y(\mathbf{x}_i)$ to each data point $\mathbf{x}_i$ can be used to indicate how much each data points $\mathbf{x}_i$ should be attended to the C domain discriminators $G_d^c, c = 1, ..., C$. The loss function of the local domain discriminator can be calculated as:

$$L_l = \frac{1}{n_s + n_t} \sum_{c=1}^{C} \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_d^c(G_d^c(\hat{y}_i^c G_f(\mathbf{x}_i)), d_i), \tag{5}$$

where G_d^c and L_d^c are the domain discriminator and its cross-entropy loss associated with class c, respectively. $\hat{y}_i^c$ is the predicted probability distribution over the class c of the input sample $\mathbf{x}_i$, and d_i is the domain label of the input sample $\mathbf{x}_i$.

D. Dynamic Adversarial Factor ω

In this section, we introduce how to dynamically evaluate the global and local distributions. It is extremely challenging to design such a dynamic scheme for adversarial learning. Intuitively, there are two natural ideas to acquire ω: *Random guessing* and *Average search*. Random guessing randomly picks a value of ω in $[0, 1]$, then performs DAAN using the corresponding value to get the result. This process can be repeated t times and the final result can be obtained by averaging all the results. Average search picks the value of $\omega = 0, 0.1, \cdots, 1.0$ to perform DAAN 11 times and uses the average results as the final result. However, both of these two ideas are computationally expensive for adversarial domain adaptation.

In this paper, we propose the *dynamic adversarial factor* (ω, the yellow part in Fig. 2) to easily, dynamically, and quantitatively evaluate the relative importance of the marginal and conditional distributions. Compared to MEDA [7] that needs to build $1 + C$ binary classifiers for the calculation of its adaptive factor, DAAN is able to update the value of the dynamic adversarial factor within the network. Firstly, instead of using the shallow features, we use the deep adversarial representations to learn and update ω, which makes DAAN more robust and accurate. Secondly, DAAN directly uses the loss of the domain discriminators to automatically fine-tune the dynamic adversarial factor, which is easier and more efficient.

To be more specific, the global domain distributions and the local domain distributions can be seen as the marginal and conditional distributions, respectively. Therefore, in DAAN, we denote the global $\mathcal{A}$-distance of the global domain discriminator as:

$$d_{\mathcal{A},g}(\mathcal{D}_s, \mathcal{D}_t) = 2(1 - 2(L_g)). \tag{6}$$

And we calculate the local $\mathcal{A}$-distance as:

$$d_{\mathcal{A},l}(\mathcal{D}_s^c, \mathcal{D}_t^c) = 2(1 - 2(L_l^c)), \tag{7}$$

where D_s^c and D_t^c denote samples from class c and L_l^c is the local subdomain discriminator loss over class c. Eventually, the dynamic adversarial factor ω can be estimated as:

$$\hat{\omega} = \frac{d_{\mathcal{A},g}(\mathcal{D}_s, \mathcal{D}_t)}{d_{\mathcal{A},g}(\mathcal{D}_s, \mathcal{D}_t) + \frac{1}{C}\sum_{c=1}^{C} d_{\mathcal{A},l}(\mathcal{D}_s^c, \mathcal{D}_t^c)}. \quad (8)$$

Note that there is no need to explicitly build extra classifiers in order to compute the local distances such as MEDA [7]. In DAAN, they can be easily implemented by taking advantages of the global and local domain discriminators. More specifically, ω is initialized as 1 in the first epoch. After each epoch, the pseudo labels of the target domain can be obtained. Then, the local distance for class c can be easily computed as:

$$L_l^c = CrossEntropy(\hat{\boldsymbol{d}}^c, \boldsymbol{d}^c), \quad (9)$$

where $\hat{\boldsymbol{d}}^c = [\hat{\boldsymbol{d}}_s^c; \hat{\boldsymbol{d}}_t^c]$ is the concatenation of the predictions output by the c-th domain discriminator d_c, and $\boldsymbol{d}^c = [\mathbf{0}; \mathbf{1}]$ with $\mathbf{0} \in \mathbb{R}^{|\hat{\boldsymbol{d}}_s^c| \times 1}$ and $\mathbf{1} \in \mathbb{R}^{|\hat{\boldsymbol{d}}_t^c| \times 1}$ is the concatenation of the true domain labels (suppose the source domain has label 0 and target domain has label 1). Similarly, the global distances can be obtained. The calculation of the dynamic adversarial factor can be performed after each epoch of iteration. Eventually, DAAN will learn a rather robust dynamic adversarial factor as the training converges.

E. Learning Procedure

DAAN mainly consists of three components: Label Classifier (Eq. (3)), Global Domain Discriminator (Eq. (4)), and Local Subdomain Discriminator (Eq. (5)). Integrating all components, the learning objective of DAAN can finally be formulated as:

$$L(\theta_f, \theta_y, \theta_d, \theta_d^c|_{c=1}^C) = L_y - \lambda((1-\omega)L_g + \omega L_l), \quad (10)$$

where λ is a trade-off parameter.

It is worth noting that although DAAN involves two hyper-parameters (λ and ω), the value of ω can be *self-calculated* by the network. Therefore, DAAN remains the same sample and efficient as other popular adversarial methods [26], [21].

When $\omega \to 0$, it means that the global distribution alignment is more important (Target I in Fig. 1), and DAAN will degenerate to DANN [26]. When $\omega \to 1$, it means that global distributions between two domains are relatively small, so the local subdomain distributions of each class is dominant (Target II in Fig. 1). In this case, DAAN will degenerate to MADA [25]. Note that in real applications, the marginal and conditional distributions are not determined. Therefore, by learning the dynamic adversarial factor ω, DAAN can be applied to diverse domain adaptation scenarios.

Denoting $\boldsymbol{\Theta} = \{\theta_f, \theta_y, \theta_d, \theta_d^c|_{c=1}^C\}$ as all the parameters to be learned, the gradient of Eq. (10) can be computed as:

$$\Delta_{\boldsymbol{\Theta}} = \frac{\Delta L_y}{\Delta \boldsymbol{\Theta}} - \lambda \frac{\Delta((1-\omega)L_g + \omega L_l)}{\Delta \boldsymbol{\Theta}} \quad (11)$$

DAAN can be trained efficiently by the Stochastic Gradient Descent (SGD) algorithm. There are two alternatives for the optimization of DAAN. We can either optimize Eq. (11) directly according to [26], or we can optimize the two objectives in Eq. (2) iteratively.

F. Discussions

In theory, the risk of DAAN can be bounded by the following theorem since it is designed by directly minimizing the target risk according to [39]:

Theorem 1 *Let $h \in \mathcal{H}$ be a hypothesis, $\epsilon_s(h)$ and $\epsilon_t(h)$ be the expected risks on the source and target domain, respectively, then*

$$\epsilon_t(h) \leqslant \epsilon_s(h) + d_{\mathcal{H}}(p, q) + C_0, \quad (12)$$

where C_0 is a constant for the complexity of hypothesis and plus the risk of an ideal hypothesis for both domains. More importantly, according to [39], $d_{\mathcal{H}}(p, q)$ is $\mathcal{H}$-divergence between domains, which can be approximately measured by the $\mathcal{A}$-distances in Eq. (4) and Eq. (5). In fact, we can regard the dynamic distribution adaptation of DAAN as the dynamic version of $\mathcal{H}$-divergence, which could learn both global and local divergences between domains. Therefore, the risk of DAAN can be theoretically bounded.

DAAN can also be explained using the attention mechanism [40]. Attention plays a critical role in human visual experience. In a computer vision task, attention tries to learn the important factors of the images. In a machine translation task, attentions helps to learn the important hidden states of the encoders. In transfer learning, we can regard that DAAN is learning the dynamic importance of the marginal and conditional distributions using neural networks. Therefore, it perceives the accurate information about distributions using the dynamic adversarial factor.

DAAN is significantly different from existing adversarial adaptation methods. Specifically, compared with global domain adaptation methods [18], [26] and the local subdomain adaptation methods [25], DAAN is able to perform dynamic adversarial distribution alignment by quantitatively calculating the relative importance of global and local distributions with a novel dynamic adversarial factor ω. Compared with MEDA [7], DAAN uses deep adversarial representations to fine-tune ω without training extra classifiers, which makes our estimation of ω significantly more accurate, easy, and efficient.

IV. Experiments

In this section, we evaluate the proposed DAAN against several state-of-the-art transfer learning methods on unsupervised domain adaptation problems. DAAN is validated on two popular datasets: ImageCLEF-DA [21] and Office-Home [20]. The code of DAAN is released at http://transferlearning.xyz.

A. Datasets

Examples of the two datasets are shown in Figure 3.

ImageCLEF-DA is a benchmark dataset for ImageCLEF 2014 domain adaptation challenge, and it is collected by selecting the 12 common categories shared by the following public datasets and each of them is considered as a

TABLE II
Accuracy(%) on Office-Home for unsuperevised domain adaptation.

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 [32]	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 [18]	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 [21]	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
MEDA [7]	46.6	**68.9**	68.8	49.0	**66.4**	**66.1**	51.8	45.0	72.9	61.2	50.3	76.0	60.2
DAAN	**50.5**	65.0	**73.7**	**53.7**	62.7	64.6	**53.5**	**45.2**	**74.0**	**66.3**	**54.0**	**78.8**	**61.8**

Fig. 3. Datasets. Up: ImageCLEF-DA; Down: Office-Home

TABLE III
Accuracy(%) on ImageCLEF-DA for unsuperevised domain adaptation.

Method	I→P	P→I	I→C	C→I	C→P	P→C	AVG
ResNet [1]	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DDC [33]	74.6	85.7	91.1	82.3	68.3	88.8	81.8
DAN [32]	75.0	86.2	93.3	84.1	69.8	91.3	83.3
RTN [41]	75.6	86.8	95.3	86.9	72.7	92.2	84.9
DANN [18]	75.0	86.0	**96.2**	87.0	74.3	91.5	85.0
D-CORAL [17]	76.9	88.5	93.6	86.8	74.0	91.6	85.2
JAN [21]	76.8	88.0	94.7	**89.5**	74.2	91.7	85.8
MADA [25]	75.0	87.9	96.0	88.8	75.2	92.2	85.8
CAN [14]	78.2	87.5	94.2	**89.5**	**75.8**	89.2	85.7
MEDA [7]	78.1	90.4	93.1	86.4	73.2	91.7	85.5
DAAN	**78.3**	**91.3**	94.4	88.0	73.5	**94.3**	**86.6**

domain: $Caltech-256$ (**C**), $ImageNet\ ILSVRC\ 2012$ (**I**), $Pascal\ VOC\ 2012$ (**P**). There are 50 images in each category and 600 images in each domain. We use all domain combinations and build 6 transfer tasks: **I→P**, **P→I**, **I→C**, **C→I**, **P→C** and **C→P**.

Office-Home is a new dataset which consists 15,588 images, which is much larger than ImageCLEF-DA. It consists of images from 4 different domains: $Artistic\ images$ (**A**), $Clip\ Art$ (**C**), $Product\ images$ (**P**) and $Real-World\ images$ (**R**). For each domain, the dataset contains images of 65 object categories collected in office and home settings. Similarly, we use all domain combinations and construct 12 transfer tasks.

B. Baselines

We compare our proposed Dynamic Adversarial Adaptation Network (DAAN) with several state-of-the-art deep unsupervised domain adaptation methods:

- Deep residual learning [1]
- Deep Domain Confusion (**DDC**) [33]
- Deep Adaptation Network (**DAN**) [32]
- Residual Transfer Network (**RTN**) [41]
- Domain Adversarial Neural Networks (**DANN**) [18]
- Deep CORAL (**D-CORAL**) [17]
- Joint Adaptation Networks (**JAN**) [21]
- Multi-Adversarial Domain Adaptation (**MADA**) [25]
- Collaborative and Adversarial Network (**CAN**) [14]
- Manifold Embedded Distribution Alignment (**MEDA**) [7]

C. Implementation Details

We implement all deep methods based on the PyTorch [42] framework, and fine-tune from ResNet-50 [1] models pre-trained on the ImageNet dataset [43]. We obtain the results of MEDA by running it on the features pre-trained by ResNet. For all the unsupervised domain adaptation tasks, we fine-tune all convolutional and pooling layers and train the classifier layer via backpropagation. Since the classifier is trained from scratch, we set its learning rate to be 10 times that of the other layers. The mini-batch Stochastic Gradient Descent (SGD) with momentum of 0.9 is taken as optimization scheme, and the learning rate changing strategy follows existing work [18]: the learning rate is not selected by a grid search due to high computational cost, it is adjusted during SGD using these formulas [26]: $\eta_k = \frac{\eta_0}{(1+\alpha k)^\beta}$, where k is the training progress linearly changing from 0 to 1, $\eta_0 = 0.01$, $\alpha = 10$ and $\beta = 0.75$. We fix $\lambda = 1, batchsize = 32$ in DAAN all the time. Other Hyperparameters are tuned via transfer cross validation [44]. Following [26], [41], classification accuracy is used as the evaluation metric. The labels for the target domain are only used for evaluation. The results are obtained by running the method 10 times to get the average accuracy.

D. Results

The classification accuracy on the ImageCLEF-DA dataset based on ResNet is shown in Table III. DAAN outperforms all comparison methods on most transfer tasks. It is noteworthy that DAAN outperforms DANN and MADA. Table II shows the results of DAAN and several baselines on the more challenging Office-Home dataset. DAAN also outperforms all comparison methods on most tasks.

TABLE IV
Error comparison between our evaluation of ω and Average search and MEDA (the results of grid search are 0)

Task	I → P	P→I	I→C	C→I	A→R	R→A	A→C	C→A	AVG
Avg search	1.50	2.30	1.70	2.51	1.57	1.20	1.32	2.00	1.76
MEDA	0.48	0.67	0.34	0.50	1.49	0.89	0.57	1.32	0.78
DAAN	**0.15**	**0.24**	**0.1**	**0.32**	**0.20**	**0.29**	**0.42**	**0.36**	**0.26**

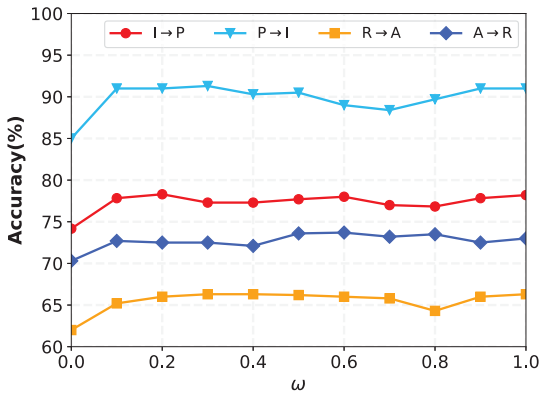


Fig. 4. Performance of several tasks when searching $\omega \in [0, 1]$.

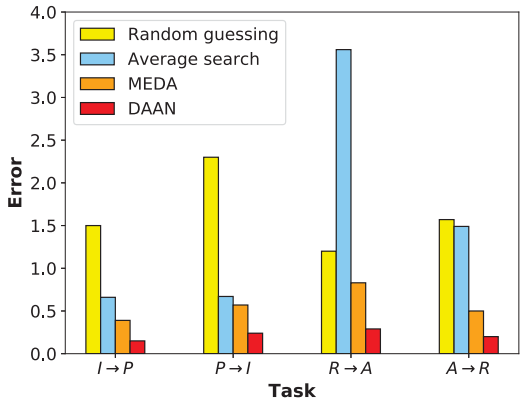


Fig. 5. (Best viewed in color) The performance w.r.t different calculation method of ω: Random guessing, Average search, MEDA, and our DAAN

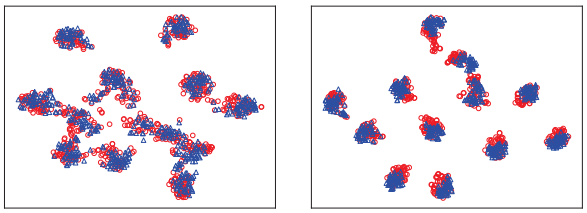
Combining these results, more insightful conclusions can be made. **(1)** The adversarial based methods (DANN [18], MADA [25], and our DAAN) usually perform better than the non-adversarial based methods (DDC [33], DAN [32], RTN [41]), which indicates that domain-adversarial learning is important for domain adaptation. **(2)** Compared with the recent domain adaptation methods MEDA and CAN, DAAN achieves better performance which proves our method is more effective. **(3)** In contrast to other latest adversarial methods, especially DANN [18] and MADA [25], our DAAN shows better performance. During training, we also noticed that our DAAN is able to converge quickly (within ¡ 30 epoches) compared to other methods. This implies its fast training performance with steady results (which will be shown in later experiments). This indicates that DAAN is capable of performing dynamic adversarial distribution alignment by quantitatively calculating the relative importance of global and local distributions.

E. Analysis of the Importance of the Dynamic Adversarial Factor ω

In this section, we evaluate the importance of dynamic adversarial factor ω in DAAN. To this end, there are two questions to be answered: 1) Is it necessary to consider the different effects of marginal and conditional distributions in adversarial domain adaptation? And 2) Is our evaluation method of ω effective?

To answer the first question, we randomly pick two tasks from Office-Home and ImageCLEF-DA and draw the results of DAAN under different ω in Fig. 4. It can be seen that the classification accuracy varies with different values of ω, which indicates the *necessity* to consider the different effects of the marginal (global) distributions and conditional (local) distributions not only in shallow domain adaptation (which can be verified in BDA [8] and MEDA [7]), but also in adversarial transfer learning. Moreover, We find that the value of optimal ω varies on different tasks and even for the same task, ω may have several optimal values. This may be because of different feature representations learned by calculating ω. Again, it implies the importance of ω in adversarial domain adaptation problems.

To answer the second question, we compare the accuracy of domain adaptation tasks contributed by different calculation method of ω: Random guessing ($t = 20$), Average search, MEDA [7], and our DAAN. For a fair study, the results of MEDA is obtained by replacing the dynamic adversarial factor in DAAN with the adaptive factor in MEDA. Note that there is *no* ground truth for ω. Instead, we run DAAN and record its accuracy by *grid search* $\omega \in \{0, 0.1, ..., 0.9, 1.0\}$ to find the optimal results as the ground truth. We use the labels of the target domain only for evaluation. The results are shown in Fig. 5. Additionally, we also list some results in Table IV to show the errors of different calculation methods. Combining the results, we can conclude that our evaluation of ω significantly and consistently outperforms other comparison methods. In addition, our evaluation is more efficient than the other three methods since it only requires to run the whole network *once* while other methods require to run DAAN several times to get stable results. Compared with MEDA, our evaluation is more efficient and accurate since it uses the adversarial representations and does not need to train extra linear classifiers. Furthermore, it is worth noting that our evaluation of ω is extremely *close* to the grid search results (which can never be reached in real applications). Therefore, the proposed dynamic adversarial factor is necessary and our



(a) JAN: P → I (b) DAAN: P → I

Fig. 6. (Best viewed in color) The t-SNE visualization of network activation. (a) and (b) are the learned representations on task P → I, respectively.

TABLE V
Ablation study of DAAN

Dataset	DANN ($\omega = 1$)	MADA ($\omega = 0$)	JAN ($\omega = 0.5$)	DAAN
ImageCLEF-DA	85.0	85.8	85.8	**86.4**
Office-Home	57.6	-	58.3	**61.6**

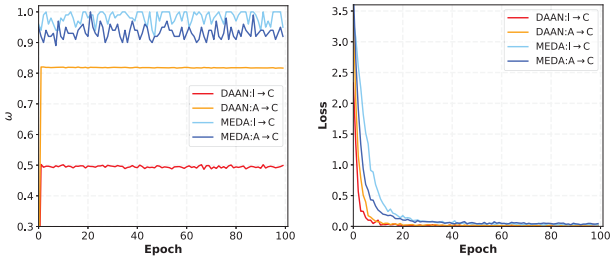
evaluation of ω is more effective and efficient.

F. Effectiveness Analysis

In this section, we analyze the effectiveness of DAAN from several aspects: ablation study, feature visualizations, and convergence analysis.

1) Ablation Study: We compare the performance of DAAN with DANN ($\omega = 0$), MADA ($\omega = 1$), and JAN ($\omega = 0.5$). All these methods can be seen as special cases of our DAAN. The average results on each dataset in Table V indicate that it is not enough to only align the marginal or conditional distributions, or aligning them with equal weights. Therefore, the proposed DAAN is able to perform dynamic distribution alignment between domain and achieve better performance. This property of DAAN is extremely important in real applications since given an unknown target domain, we can never know the contributions of either marginal or conditional distribution in domain divergence. DAAN makes it possible to easily, dynamically, and quantitatively evaluate their relative importance in adversarial learning.

2) Feature Visualization: To further evaluate the performance of DAAN, we visualize the network activations on task P → I (12 classes) learned by JAN and DAAN using t-SNE embeddings [12] in Fig. 6(a)-6(b). Red circles are the source samples and blue triangles are the target samples. The visualization results reveal some important observations. **(1)** As for the results of JAN, the distributions between the source and target domains are not aligned very well and different categories are not well discriminated clearly. **(2)** In contrast, for the representations learned with our DAAN, not only the distributions between the source and target domains are aligned very well, different categories can also be discriminated more clearly. This ensures that our DAAN can achieve better performance. The above observations suggest that DAAN is able to learn more representative and transferable features by quantitatively calculating the relative



(a) ω (b) Loss

Fig. 7. Value change of dynamic adversarial factor ω and loss w.r.t. iterations

importance of global domain distributions and local subdomain distributions.

3) Convergence Analysis: In this section, we evaluate the convergence of DAAN. On the same DAAN architecture, we compare the change of ω between DAAN and MEDA w.r.t. iterations in Fig. 7(a). Additionally, their loss can be seen in Fig. 7(b). From these results, we can observe: **(1)** DAAN can reach a quick and steady convergence after 20 epochs. **(2)** The dynamic adversarial factor ω can also reach a steady value after several iterations, while the adaptive factor of MEDA takes more iterations. These results demonstrate that the proposed DAAN can not only reach competitive performances, it can also be trained easily with steady results.

V. Conclusions and Future Work

In this paper, we propose a novel Dynamic Adversarial Adaptation Network (DAAN) for adversarial transfer learning. DAAN is able to learn a domain-invariant network while performing dynamic adversarial distribution alignment to quantitatively evaluate the relative importance of marginal (global) domain distributions and conditional (local) subdomain distributions. To the best of our knowledge, DAAN is the first attempt to perform easy, dynamic, and quantitative evaluation of these two distributions in adversarial neural networks. DAAN can be easily implemented and used in real domain adaptation tasks. Experimental results demonstrate that DAAN achieves superior performance compared to state-of-the-art deep methods.

DAAN is a general transfer learning and domain adaptation approach and it can be applied to a large amount of classification related applications such as object detection, image segmentation, and visual tracking. In the future, we plan to extend DAAN for the more challenging cross-domain data mining problems.

VI. Acknowledgment

The first two authors contributed equally. This work is supported in part by National Key R & D Plan of China (No.2017YFB1002802), National Natural Science Foundation of China (No.61572471), Beijing Municipal Science & Technology Commission (No.Z171100000117001) and Chinese Academy of Sciences Research Equipment Development Project under Grant (No.YZ201527).

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] S. J. Pan, Q. Yang *et al.*, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[4] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in neural information processing systems*, 2007, pp. 601–608.

[5] Y. Chen, J. Wang, M. Huang, and H. Yu, "Cross-position activity recognition with stratified transfer learning," *Pervasive and Mobile Computing*, vol. 57, pp. 1–13, 2019.

[6] J. Wang, Y. Chen, H. Yu, M. Huang, and Q. Yang, "Easy transfer learning by exploiting intra-domain structures," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2019.

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

[8] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 1129–1134.

[9] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2066–2073.

[10] J. Wang, Y. Chen, L. Hu, X. Peng, and S. Y. Philip, "Stratified transfer learning for cross-domain activity recognition," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2018, pp. 1–10.

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

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[14] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3801–3809.

[15] Y. Zhu, F. Zhuang, J. Wang, J. Chen, Z. Shi, W. Wu, and Q. He, "Multi-representation adaptation network for cross-domain image classification," *Neural Networks*, 2019.

[16] C. Yu, J. Wang, Y. Chen, and Z. Wu, "Accelerating deep unsupervised domain adaptation with transfer channel pruning," *arXiv preprint arXiv:1904.02654*, 2019.

[17] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European Conference on Computer Vision*. Springer, 2016, pp. 443–450.

[18] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning (ICML)*, 2015.

[19] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, "Deep transfer learning for cross-domain activity recognition," in *Proceedings of the 3rd International Conference on Crowd Science and Engineering*. ACM, 2018, p. 16.

[20] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5018–5027.

[21] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *ICML*, 2017.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[23] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[24] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2017, pp. 6670–6680.

[25] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[26] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[27] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2960–2967.

[28] B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation." in *BMVC*, 2015, pp. 24–1.

[29] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation." in *AAAI*, vol. 6, no. 7, 2016, p. 8.

[30] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2200–2207.

[31] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[32] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning (ICML)*, 2015.

[33] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[34] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2272–2281.

[35] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (cmd) for domain-invariant representation learning," in *International Conference on Learning Representations (ICLR)*, 2017.

[36] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," *arXiv preprint arXiv:1611.01673*, 2016.

[37] A. Kumar, P. Sattigeri, K. Wadhawan, L. Karlinsky, R. Feris, B. Freeman, and G. Wornell, "Co-regularized alignment for unsupervised domain adaptation," in *Advances in Neural Information Processing Systems*, 2018, pp. 9367–9378.

[38] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2018, pp. 5419–5428.

[39] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in neural information processing systems*, 2007, pp. 137–144.

[40] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *International Conference on Learning Representations (ICLR)*, 2016.

[41] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 136–144.

[42] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[44] E. Zhong, W. Fan, Q. Yang, O. Verscheure, and J. Ren, "Cross validation framework to choose amongst models and datasets for transfer learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 547–562.