Adversarial Dual Distinct Classifiers for Unsupervised Domain Adaptation

Taotao Jing[1], Zhengming Ding[2]

[1]Department of ECE, Indiana University-Purdue University Indianapolis, USA
[2]Department of CIT, Indiana University-Purdue University Indianapolis, USA
jingt, zd2@iu.edu

Abstract

Unsupervised Domain adaptation (UDA) attempts to recognize the unlabeled target samples by building a learning model from a differently-distributed labeled source domain. Conventional UDA concentrates on extracting domain-invariant features through deep adversarial networks. However, most of them seek to match the different domain feature distributions, without considering the task-specific decision boundaries across various classes. In this paper, we propose a novel Adversarial Dual Distinct Classifiers Network (AD^2CN) to align the source and target domain data distribution simultaneously with matching task-specific category boundaries. To be specific, a domain-invariant feature generator is exploited to embed the source and target data into a latent common space with the guidance of discriminative cross-domain alignment. Moreover, we naturally design two different structure classifiers to identify the unlabeled target samples over the supervision of the labeled source domain data. Such dual distinct classifiers with various architectures can capture diverse knowledge of the target data structure from different perspectives. Extensive experimental results on several cross-domain visual benchmarks prove the model's effectiveness by comparing it with other state-of-the-art UDA.

1. Introduction

Deep neural networks (DNNs) have made significant progress with the help of numerous well-labeled training data and achieved remarkable performance improvement on various tasks [20, 35]. However, massive amounts of annotated training data are not always available due to the dramatically expensive data collecting and annotating costs. Domain adaptation (DA) has attracted extremely increasing attention because it focuses on a frequent and real-world issue when we have no access to massive labeled target domain training data [27, 23, 30, 40, 18]. Domain adaptation explores the common factors between the source and target domain to find a shared latent feature space where both

marginal and conditional distribution mismatch across domains are minimized. Based on this, various domain adaptation efforts have been developed, including feature alignment and classifier adaptation [37, 14, 42].

Recent research efforts on domain adaptation have already shown promising performance via seeking an effective domain-invariant feature extractor across two domains so that the source knowledge could be adapted to facilitate the recognition task in the target domain [13, 39, 32, 16, 15]. The idea is to deploy cross-domain matching losses to guide the domain-invariant feature learning. First of all, the discrepancy loss (e.g., maximum mean discrepancy (MMD)) is one of the most widely-used strategies to measure the distribution difference across the source and target domains [1, 11, 17]. Along this, many DA approaches explore to design a class-wise MMD by incorporating the pseudo labels of target data [14, 30]. Secondly, the adversarial loss has been successfully applied to eliminate the domain shifts on the feature or pixel level [7, 34, 22, 45], where one domain discriminator or more are trained with a feature generator in an adversarial manner. Moreover, various reconstruction penalties are proposed on target samples to obtain the target specific structures, e.g., iCAN [44]. However, most existing domain adaptation methods suffer from only considering the domain-wise adaptation but leaving out the task-specific category boundaries alignment during the explicit cross-domain distribution alignment.

To address this issue, some recent DA works aim to consider the task-specific category-level alignment jointly [34, 22, 23]. Along this line, Saito *et al.* present Maximum Classifier Discrepancy (MCD) with two task-specific classifiers to detect category boundaries and jointly align features distribution and category boundaries across domains [34]. Following this, Lee *et al.* propose Sliced Wasserstein Discrepancy (SWD) as a new probability distribution discrepancy measurement to capture the natural notion of dissimilarity between the outputs of task-specific classifiers [22]. Later on, [45] promotes Domain-Symmetric Networks (SymNets) as well as a two-level (feature-level and category-level) domain confusion scheme to drive the
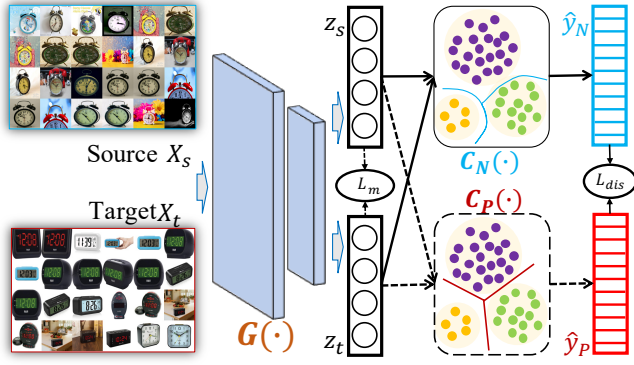
Figure 1. Framework overview of our proposed model, where $G(\cdot)$ is the domain-invariant embedding features generator, $C_N(\cdot)$ denotes the fully-connected neural networks classifier (solid line) and $C_P(\cdot)$ means the prototypical classifier (dash line). $\mathcal{L}_m$ and $\mathcal{L}_{dis}$ are explored to align the feature and prediction distribution differences across two domains and dual classifiers, respectively.

learning of intermediate features to be invariant at the corresponding categories of the two domains. These method benefit from various strategies to maximize the disparity of the dual classifiers prediction results, however, considering the utterly same architecture classifiers not only limits the features distribution knowledge obtained from different perspectives, but also suffers from the risk that the two task-specific classifiers may result in the similar class-wise boundaries, especially when the imbalanced data distribution across various categories.

In this paper, we propose a novel Adversarial Dual Distinct Classifiers Network (AD^{2}CN) with two different-architecture classifiers, e.g., Neural Networks Classifier and Prototypical Classifier, to facilitate the alignment of both domain distributions and category decision boundaries (Fig. 1). To our best knowledge, it is a pioneering work to explore dual different structure classifiers in domain adaptation. The general idea is to explore adversarial training over two different architecture classifiers on the output of one domain-invariant feature generator. To sum up, we highlight the three-fold contributions of this paper as follows:

- We exploit dual different architecture task-specific classifiers over source supervision to exploit the task-specific decision boundaries on the target domain. With different properties of dual classifiers in prediction, we have a better chance of capturing ground-truth classifier decision boundaries for the target domain.

- We propose a novel discriminative cross-domain alignment loss and *Importance Guided Optimization* strategy to mitigate the cross-domain mismatching. This will facilitate the process of aligning the domain-invariant embedding features distribution across domains, and eliminate the distraction of misestimated target samples at the beginning of optimizing.

- We adopt a discrepancy loss to maximally improve the prediction performance of dual classifiers in coupling the cross-domain label distributions, which is trained in an adversarial way with domain-invariant feature generator and dual classifiers. Thus, they can benefit from each other to boost the target learning task.

2. Related Work

Domain adaptation (DA) has been extensively studied recently, which casts a light when there are no or limited labels in target domain and shows very promising performance in different vision applications [45, 23, 28, 41, 22].

With the renaissance of deep neural networks, deep DA methods successfully embed DA into deep learning pipelines by either minimizing an appropriate distribution distance metric [26] or leveraging adversarial technologies to generate domain-invariant representations [34, 3]. The cross-domain distribution discrepancy enlarged by traditional deep learning models can be explicitly alleviated by incorporating various domain alignment strategies at the top layers. To name a few, Domain Adaptation Network (DAN) applies multiple kernel MMD distances on the last three task-specific layers to minimize the distribution difference [25]. Long *et al*. proposed a joint adaptation network (JAN) and joint MMD criterion to solve the problem [29]. Another strategy is to leverage generative adversarial networks (GAN) [10] to couple the cross-domain discrepancy in an adversarial manner [7, 34, 44, 45]. Such techniques aim to train a domain discriminator to differentiate source and target samples, while the feature generator will deceive the domain discriminator, such that the domain-invariant features will be produced. Ganin *et al*. proposed DANN to generate task-specific discriminative while domain-wise indiscriminative features [8]. Tzeng *et al*. presented ADDA for adversarial adaptation [38].

Both discrepancy and adversarial loss based methods attempt to match the whole source and target domain distribution completely, neither of them considers the target domain data structure and task-specific decision boundaries. To address this, Saito *et al*. adopted the task-specific category decision boundaries and proposes a model with two classifiers as a discriminator to detect the relationship between the source and target domain data (MCD) [34]. By maximizing the prediction results of the two classifiers, the framework is able to screen out target samples that are near the category decision boundaries and far from the source domain support. Following this, Lee *et al*. extended MCD and proposed a novel Wasserstein metric to capture the natural notion of dissimilarity between the outputs of two task-specific classifiers [22]. Most recently, Li *et al*. claimed that label distribution alignment is still not enough and proposed Joint Adversarial Domain Adaptation (JADA) to explore unified adversarial learning mechanism to align the cross-

domain domain-wise and class-wise distribution simultaneously [23]. Unfortunately, existing works seek to maximize the prediction difference between two same architecture classifiers to explore different task-specific knowledge, limiting the divergence of category decision boundaries captured across domains.

Differently, we propose a novel framework with two different structure classifiers, which assist the model to learn more diverse data distribution patterns and less similar category decision boundaries from different perspectives. Integrating task-specific category boundaries and feature-level cross-domain adaptation, our proposed model is able to narrow the data mismatch of source and target domain in the shared domain invariant embedding space. Moreover, we explore a cross-domain discriminative distribution alignment under the sample Importance Guided Optimization strategy, which has been experimentally proven to eliminate the source and target domain shift.

3. The Proposed Method

3.1. Preliminaries and Motivation

Given a labeled source domain $\mathcal{D}_s = \{\mathbf{X}_s, \mathbf{Y}_s\} = \{(\mathbf{x}_s^i, \mathbf{y}_s^i)\}_{i=1}^{n_s}$ which contains n_s labeled samples, as well as an unlabeled target domain $\mathcal{D}_t = \{\mathbf{X}_t\} = \{\mathbf{x}_t^j\}_{j=1}^{n_t}$ of n_t unlabeled samples. $\mathbf{P}_s(\mathbf{x}_s)$ and $\mathbf{P}_t(\mathbf{x}_t)$ denote the source and target domain different data distributions respectively $(\mathbf{P}_s(\mathbf{x}_s) \neq \mathbf{P}_t(\mathbf{x}_t))$. $\mathcal{C}_s$ and $\mathcal{C}_t$ mean source and target domain identical label spaces. $\mathbf{Y}_s \in \mathbb{R}^{n_s \times C}$ is the source domain ground truth label set which is accessible for training, where $C = C_{s/t} = |\mathcal{C}_{s/t}|$ is the number of total categories. The goal of domain adaption is to seek a model to predict the unlabeled target data over the supervision from the source domain.

Recent domain adaptation works apply adversarial networks to generate domain invariant features of the source and target domain samples, which will make the classifiers trained only on the source domain data available on the target domain[7, 9, 44]. Most of them aim to match the distribution of source and target domain completely, without considering the task-specific decision boundaries between different categories. Most recently, the idea of dual adversarial classifiers [34, 22, 45, 23] has been explored to replace the original adversarial domain adaptation with a binary domain discriminator. However, they obtain two same-type classifiers from scratch over labeled source data. This would limit the discriminative ability in target prediction since the same-type classifiers would tend to have similar properties. Traditional neural networks classifier aims to fit the training data by achieving optimal objective value, thus the learned classifier boundaries would capture the global structure of the data to maximally separate different classes. Such a decision boundary over source supervision cannot

be well adapted to target samples in different distribution. Therefore, two same-architecture neural network classifiers over source supervision are challenging to diversify the decision boundaries.

This motivates us to explore two different architecture classifiers, and thus we propose a novel adversarial dual classifiers network with two different structure classifiers, Neural Networks Classifier $C_N(\cdot)$ and Prototypical Classifier $C_P(\cdot)$ [36], which can capture various data distribution pattern and more diverse task-specific category boundaries from different perspectives, and also promote the out of source support target samples detection process. Interestingly, the prototypical classifier explores the local structure of the data since prototypes are used to assign labels based on the similarity between samples and each prototype. The competition between two different structure classifiers is more likely to diversify the decision boundaries to benefit from adversarial training with domain-invariant generator.

3.2. Adversarial Dual Distinct Classifiers Network

We first present the overall framework of our proposed adversarial dual classifier network in Fig. 1. Given the labeled source and unlabeled target domain data, the domain invariant embedding features are generated and aligned by the discriminative cross-domain alignment, then the dual classifiers, which consist of two classifiers with different architectures, will promote the task-specific decision boundaries further. $G(\cdot)$ is the generator used to map source and target domain data to a shared embedding feature space, in which the target samples are close to the support of the source domain data. The following two different structure classifiers, fully-connected neural network classifier $C_N(\cdot)$ and prototypical classifier $C_P(\cdot)$, will capture diverse and various task-specific categories knowledge on target domain from different perspectives.

3.2.1 Dual Classifiers Over Source Supervision

Since $\mathbf{X}_s$ and $\mathbf{X}_t$ have different distributions, a domain-invariant feature generator $G(\cdot)$ is deployed to capture more enriched information across source and target through hierarchical structures, followed by our dual classifiers, $C_N(\cdot)$ (fully-connected neural network classifier) and $C_P(\cdot)$ (prototypical classifier). With the extracted feature $\mathbf{z}_{s/t}^i = G(\mathbf{x}_{s/t}^i)$ from $G(\cdot)$ as input, we can calculate the corresponding probability prediction with two classifiers $C_N(\cdot)$ and $C_P(\cdot)$ as $\hat{\mathbf{y}}_{N/P,s/t}^i = C_{N/P}(\mathbf{z}_{s/t}^i)$.

Specifically, $C_N(\cdot)$ is the traditional multi-layer non-linear classifier, while $C_P(\cdot)$ is defined as the similarity between target sample feature $\mathbf{z}_t^i$ to each category prototype $\boldsymbol{\mu}_c$ (i.e., class center), that is, $\hat{\mathbf{y}}_{P,t}^{i(c)} = \boldsymbol{\Phi}(\mathbf{z}_t^i, \boldsymbol{\mu}_c)$. For each class, the prototype $\boldsymbol{\mu}_c = \frac{1}{n_t^c} \sum_{i=1}^{n_t^c} \mathbf{z}_t^{i(c)}$, where n_t^c and $\mathbf{z}_t^{i(c)}$

denote the number of target samples and extracted domain invariant feature belonging to class c. We apply the $C_P(\cdot)$ prediction $\hat{\mathbf{y}}^i_{P,t}$ as the predicted pseudo label to target sample $\mathbf{x}^i_t$ to get the category prototypes $\boldsymbol{\mu}_c$.

In order to obtain task-specific discriminative features from generator $G(\cdot)$, while keeping classification performance on source domain, we add the supervision from source to learn the parameters of $C_N(\cdot)$ and $G(\cdot)$. Since $C_P(\cdot)$ does not contain any trainable parameters, the supervision over $C_P(\cdot)$ prediction on the source domain tends to optimize the generator $G(\cdot)$ only. To this end, we aim to minimize the cross-entropy loss over $\mathbf{Y}_s$ and predicted labels from $C_N(\cdot)$ and $C_P(\cdot)$, defined as follows:

$$\mathcal{L}_s = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathcal{L}(\hat{\mathbf{y}}^i_{N,s}, \mathbf{y}^i_s) + \frac{1}{n_s} \sum_{i=1}^{n_s} \mathcal{L}(\hat{\mathbf{y}}^i_{P,s}, \mathbf{y}^i_s), \quad (1)$$

where $\mathcal{L}$ is the cross-entropy loss. $\hat{\mathbf{y}}^i_{N,s}$ and $\hat{\mathbf{y}}^i_{P,s}$ are the probability outputs of classifier $C_N(\cdot)$ and $C_P(\cdot)$, while $\mathbf{y}^i_s$ is the ground-truth label of source sample $\mathbf{x}^i_s$, respectively.

3.2.2 Adversarial Dual Classifiers

The dual classifiers are capable of recognizing target domain samples close to the support of the source domain. For those target domain samples which are far from the source domain support, the two classifiers would tend to obtain different probability outputs. To detect target samples outside of the support from source supervision, we propose to measure the disagreement of the classifiers prediction results with distribution discrepancy measurement [22, 34].

Existing works exploit varying the dual classifiers by maximizing the divergence between the predictions. However, the same classifier structure with slightly different random initializations [34, 22] will weaken the ability to capture diverse task-specific knowledge and decision boundaries from different perspectives. In our model, we build two different architecture classifiers, which are more likely to capture the inconsistent information from various perspective. Thus, adversarial training would further enhance the target prediction performance, and the classifier discrepancy is defined as:

$$\mathcal{L}_{dis} = \mathcal{F}(\hat{\mathbf{y}}^i_{N,t}, \hat{\mathbf{y}}^i_{P,t}), \quad (2)$$

where $\hat{\mathbf{y}}^i_{N/P,t}$ represent the probability prediction obtained from the two classifiers for the sample $\mathbf{x}^i_t$ respectively. $\mathcal{F}(\cdot, \cdot)$ denotes the discrepancy measurement function, which is able to capture distribution geometric information to calculate the discrepancy between the probability prediction distributions, and solve gradient vanishing problems occurred in adversarial learning methods.

3.2.3 Discriminative Cross-Domain Alignment

So far, our model only aligns cross-domain distributions in terms of label space, we further exploit feature distribution alignment to boost the domain-invariant feature learning. Maximum Mean Discrepancy (MMD) has been sufficiently explored as a promising strategy to reduce the domain-wise distance between the mean of source and target domain features, or class-wise distances between each class source and target features with the pseudo labels for target samples [27]. The domain-wise MMD to measure marginal distribution across domains is defined as $\mathcal{H}(\mathbb{E}_{\mathbf{x}^i_s \sim \mathcal{D}_s}[\mathbf{z}^i_s] - \mathbb{E}_{\mathbf{x}^j_t \sim \mathcal{D}_t}[\mathbf{z}^j_t])$ [27], where $\mathcal{H}(\cdot)$ is the function used to evaluate the distribution difference. Furthermore, existing works [5] also seek to explore the class-wise MMD to align conditional distribution disparity across domain:

$$\mathcal{L}_c = \frac{1}{C} \sum_{c=1}^{C} \mathcal{H}\Big(\mathbb{E}_{\mathbf{x}^i_s \sim \mathcal{D}^c_s}[\mathbf{z}^i_s] - \mathbb{E}_{\mathbf{x}^j_t \sim \mathcal{D}^c_t}[\mathbf{z}^j_t]\Big), \quad (3)$$

where C denotes the total number of categories, $\mathbf{z}^{i/j}_{s/t}$ denote the generated embedding representations of source sample $\mathbf{x}^i_s$ and target sample $\mathbf{x}^j_t$ belonging to class c.

However, conventional DA algorithms only seek to minimize the distribution difference between source and target domains when samples are from the same class. We further propose to explicitly take the information of different categories into account and measure the *diff-class* divergence across domains defined as:

$$\mathcal{L}_d = \frac{1}{C} \frac{1}{C-1} \sum_{c=1}^{C} \sum_{\substack{c'=1, \\ c' \neq c}}^{C} \mathcal{H}\Big(\mathbb{E}_{\mathbf{x}^i_s \sim \mathcal{D}^c_s}[\mathbf{z}^i_s] - \mathbb{E}_{\mathbf{x}^j_t \sim \mathcal{D}^{c'}_t}[\mathbf{z}^j_t]\Big),$$

$$(4)$$

where the *diff-class* divergence $\mathcal{L}_d$ calculates the average distances of all different class center pairs across domains. To sum up, our discriminative cross-domain alignment is defined as $\mathcal{L}_m = \mathcal{L}_c - \mathcal{L}_d$.

Due to the lack of target domain labels, we explicitly assign $\hat{\mathbf{y}}^i_{P,t}$, the prediction of $C_P(\cdot)$, as pseudo labels to the target samples $\mathbf{x}^i_t$. To exploit more effective knowledge transfer iteratively, we propose an *Importance Guided Optimization* strategy to only consider those target samples with high prediction confidences during the cross-domain alignment since lower-confident samples would mislead the optimization. That is, only samples with $\{(\mathbf{x}^i_t, \hat{\mathbf{y}}^{i(c)}_{P,t}) \mid \hat{y}^{i(c)}_{P,t} > \sigma_1, \mathbf{x}^i_t \in \mathcal{D}_t\}$ are accepted to construct the cross-domain alignment $\mathcal{L}_m$, where $\hat{\mathbf{y}}^{i(c)}_{P,t}$ is the $C_P(\cdot)$ probability prediction of $\mathbf{x}^i_t$ belonging to class c, and $\sigma \in [0,1]$ is a constant threshold. It is noteworthy that we do not impose always covering the whole label space during training, since only considering those classes with high-confident samples is prone to result in effective cross-domain alignment by

avoiding too many mis-classified target samples, especially in the early training stage.

3.3. Overall Objective and Optimization

To eliminate the side effect of uncertainty on unlabeled target prediction, we also explore the entropy minimization regularization [45, 26, 28]:

$$\mathcal{L}_{em} = -\frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{c=1}^{C} (\hat{\mathbf{y}}_{N,c}^i \log \hat{\mathbf{y}}_{N,c}^i + \hat{\mathbf{y}}_{P,c}^i \log \hat{\mathbf{y}}_{P,c}^i), \quad (5)$$

where $\hat{\mathbf{y}}_{N,c}^i$ and $\hat{\mathbf{y}}_{P,c}^i$ denote the prediction of $\mathbf{x}_t^i$ belonging to class c obtained by $C_N(\cdot)$ and $C_P(\cdot)$, respectively.

To sum up, we integrate adversarial dual classifiers training and cross-domain discriminative alignment together, and propose our overall objective function as:

$$\begin{aligned} \min_G \mathcal{L}_s + \mathcal{L}_{em} + \lambda_1 \mathcal{L}_{dis} + \lambda_2 \mathcal{L}_m, \\ \min_{C_N} \mathcal{L}_s - \lambda_1 \mathcal{L}_{dis}, \end{aligned} \quad (6)$$

where λ_1 and λ_2 are hyper-parameters to balance the contribution of loss terms $\mathcal{L}_{dis}$, $\mathcal{L}_m$, respectively.

Similar to existing adversarial networks training strategy, we freeze the generator $G(\cdot)$ to train classifiers, then freeze the parameters of the classifiers to update the generator $G(\cdot)$. It is noteworthy that only $C_N(\cdot)$ contains trainable parameters because $C_P(\cdot)$ only relies on the embedding features produced by the generator $G(\cdot)$. Meanwhile, inspired by [34], in order to keep the performance of the networks on the source domain and detect target samples far from source domain support, we train our framework by three steps:

Step A. We train the feature generator $G(\cdot)$ and classifier $C_N(\cdot)$ only on source domain $\mathcal{D}_s$ which is the same as supervised learning tasks. Due to $C_P(\cdot)$ does not have any trainable parameters, only parameters in $G(\cdot)$ and $C_N(\cdot)$ would be updated. Our model aims to detect target samples which are outside of source support from those which are close to support of source domain, keeping good ability and performance on classifying the source domain samples correctly is crucial and necessary. The optimization objective is defined as $\min_{G,C_N} \mathcal{L}_s$.

Step B. We need to assign unlabeled target domain samples pseudo labels by classifiers we already have. In our experiments, we explore the prediction results of $C_P(\cdot)$ to obtain pseudo labels of the target samples, which are experimentally proven to achieve better performance, and we will discuss it in the ablation analysis section. We fix the feature generator $G(\cdot)$ and update the classifier $C_N(\cdot)$ to maximize the distribution discrepancy between the classification results of $C_N(\cdot)$ and $C_P(\cdot)$ on the target domain, which can detect the target samples excluded by the source

domain data support, and we obtain the training objective function as $\min_{C_N} \mathcal{L}_s - \lambda_1 \mathcal{L}_{dis}$.

Step C. We freeze the parameters of the classifier $C_N(\cdot)$ and update generator $G(\cdot)$ to minimize the distribution discrepancy between the predictions of $C_N(\cdot)$ and $C_P(\cdot)$ on the target domain, through which both $C_N(\cdot)$ and $C_P(\cdot)$ classifiers will have more similar and correct prediction on target domain samples. Furthermore, together with the discriminative cross-domain alignment, the generator $G(\cdot)$ tends to couple the source and target domain closer but discriminative in the embedding feature space. The optimization objective is $\min_G \mathcal{L}_s + \mathcal{L}_{em} + \lambda_1 \mathcal{L}_{dis} + \lambda_2 \mathcal{L}_m$.

These three steps repeat once in each iteration in our experiments. The generator $G(\cdot)$ and classifier $C_N(\cdot)$ are initialized and pre-trained on source domain data.

4. Experimental Results

4.1. Datasets & Experimental Setup

Office-Home [39] consists of 15,500 images from 65 categories in 4 different domains: Artistic images (Ar), Clip Art (Cl), Product (Pr), and Real-World images (Rw). In total, by choosing any two domain as one task, we can build 12 cross-domain tasks to evaluate our proposed model.

Office-31 contains 4,110 images of 3 domains: Amazon (A), Webcam (W), and DSLR (D) and each domain consists of 31 categories. We evaluate our method on 6 cross-domain tasks to testify the validation of our model.

Comparisons. We compare our proposed method with several state-of-the-art unsupervised domain adaptation models: Deep Adaptation Networks (DAN) [25], Reverse Gradient (RevGrad) [7], Joint Adaptation Networks (JAN) [29], Self-Ensembling (SE) [6], Multi-adversarial Domain Adaptation (MADA) [31], Conditional Adversarial Domain Adaptation Networks (CDAN) [26], Disentangled Semantic Representation (DSR) [2], Domain-specific Whitening Transform & Min-Entropy Consensus (DWT-MEC) [33], Minimum Centroid Shift (MCS) [24], Adaptive Feature Norm Approach (AFN) [41], Domain Symmetric Networks (SymNets) [45], Bi-Directional Generation (BDG) [43]. All our experiments follow standard unsupervised domain adaptation protocols: all labeled source domain data and labels, as well as unlabeled target domain data are used for training. All comparisons are back-boned with ResNet-50 or using ResNet-50 features [12].

Implementation Details. We implement our model with PyTorch and adopt ResNet-50[12] as the backbone. Specifically, a ResNet-50 network is pre-trained on ImageNet [4] and fine-tuned on the source domain, then applied to both source and target domain data to obtain the feature representation with dimension 2,048 without the last fully connected layer. $G(\cdot)$ is a two-layer fully-connected neural network,

Table 1. Comparisons of Recognition Rates (%) of Unsupervised Domain Adaptation on Office+Home Dataset (ResNet-50).

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
Res-50 [12]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	51.2	59.9	46.1
DAN [25]	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
RevGrad [7]	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 [29]	45.9	61.2	68.9	50.4	59.7	60.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
SE [6]	48.8	61.8	72.8	54.1	63.2	65.1	50.6	49.2	72.3	66.1	55.9	78.7	61.5
DSR [2]	53.4	71.6	77.4	57.1	66.8	69.3	56.7	49.2	75.7	68.0	54.0	79.5	64.9
DWT-MEC [33]	50.3	72.1	77.0	59.6	69.3	70.2	58.3	48.1	77.3	69.3	53.6	82.0	65.6
CDAN+E [26]	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
MCS [24]	55.9	73.8	79.0	57.5	69.9	71.3	58.4	50.3	78.2	65.9	53.2	82.2	66.3
AFN [41]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
SymNets [45]	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	81.6	67.6
BDG [43]	51.5	73.4	78.7	**65.3**	71.5	73.7	**65.1**	49.7	**81.1**	74.6	55.1	**84.8**	68.7
Ours	**57.4**	**77.3**	**80.0**	63.4	**76.4**	**76.4**	64.2	**52.4**	80.7	69.6	**57.2**	83.9	**69.9**

Table 2. Comparisons of Recognition Rates (%) of Unsupervised Domain Adaptation on Office-31 Dataset (ResNet-50).

Method	Res-50 [12]	DAN [25]	RevGrad [7]	JAN [29]	MADA [31]	CDAN+E [26]	AFN [41]	SymNets [45]	BDG [43]	Ours
A→W	68.4±0.2	80.5±0.4	82.0±0.4	86.0±0.4	90.0±0.1	**94.1±0.1**	90.1±0.1	90.8±0.1	93.6±0.4	93.6±0.3
D→W	96.7±0.1	97.1±0.2	96.9±0.2	96.7±0.3	97.4±0.1	98.6±0.1	98.6±0.2	98.8±0.3	**99.0±0.1**	98.9±0.2
W→D	99.3±0.1	99.6±0.1	99.1±0.1	99.7±0.1	99.6±0.1	**100.0±0.0**	99.8±0.0	**100.0±0.0**	**100.0±0.0**	99.8±0.0
A→D	68.9±0.2	78.6±0.2	79.7±0.4	85.1±0.4	87.8±0.2	92.9±0.2	90.7±0.5	93.9±0.5	93.6±0.3	**95.4±0.3**
D→A	62.5±0.3	63.6±0.3	68.2±0.4	69.2±0.3	70.3±0.3	71.0±0.3	73.0±0.2	74.6±0.6	73.2±0.2	**74.9±0.3**
W→A	60.7±0.3	62.8±0.2	67.4±0.5	70.70.5	66.4±0.3	69.3±0.3	70.2±0.3	72.5±0.5	72.0±0.1	**75.0±0.5**
Avg.	76.1	80.4	82.2	84.6	85.2	87.7	87.1	88.4	88.5	**89.6**

with hidden layer output as 1,024 followed by ReLU activation function, and the dropout probability retaining is 0.5. The output embedding features $\mathbf{z}_{s/t}$ dimension is 512. $C_N(\cdot)$ is a two-layer fully-connected neural network with 512 as the input and hidden layer dimension, the output dimension is the same as the number of categories in the whole label space C. Cosine similarity is accepted as the measurement metric $\Phi(\cdot, \cdot)$ in $C_P(\cdot)$. All parameters are updated with Adam optimizer [19] and the learning rate is set as 0.001 on Office-Home and Office-31 dataset. $G(\cdot)$ and $C_N(\cdot)$ are pre-trained and initialized on source domain data only with the learning rate as 0.1 for 2,000 iterations. We deploy SWD distance [22] as the discrepancy measurement function $\mathcal{F}(\cdot, \cdot)$, and accept L-2 norm as $\mathcal{H}(\cdot)$ to evaluate the distribution divergence. λ_1 and λ_2 are fixed as 0.1 for all tasks. σ is set to be 0.03. For the prototypical classifier $C_P(\cdot)$, we initialize the class prototypes with the source domain features class centers $\boldsymbol{\mu}_c^s = \frac{1}{n_s^c} \sum_{i=1}^{n_s^c} \mathbf{z}_s^i$, then update the prototypes with target domain category centroids representation $\boldsymbol{\mu}_c^t = \frac{1}{n_t^c} \sum_{j=1}^{n_t^c} \mathbf{z}_t^j$ after obtaining the target domain samples pseudo labels $\hat{\mathbf{y}}_{P,t}$ iteratively till reaching convergence or the max step (which is set as 3), and return the last step $C_P(\cdot)$ prediction. All results reported in Tables 1 and 2 are the average of three random experimental results obtained by classifier $C_P(\cdot)$, and we will discuss the performances of $C_N(\cdot)$ and $C_P(\cdot)$ in the ablation study section.

4.2. Comparison Results

Table 1 and Table 2 report the classification results on target domain data of our proposed model and other comparative methods on Office-Home and Office-31 datasets respectively. All comparison results are from their original paper or quoted from [21, 45, 43], as we adopt exactly the same settings. It is noteworthy that our proposed model outperforms state-of-the-art methods on all benchmark datasets in terms of average accuracy, and obtains the best or comparable performances to the state-of-the-art domain adaptation methods in most cases. Although the Office-Home dataset is more challenging than Office-31 due to more categories and samples, as well as significant distribution dissimilarity, our proposed model still improves the performance on most tasks, which demonstrates the efficiency and effectiveness of our proposed framework.

DAN and JAN are both MMD-based methods, which seek to eliminate the cross-domain distribution disparity and match the whole source and target domain to a shared domain-invariant feature space. DAN attempts to align feature representations from multiple layers through a multi-kernel variant of MMD. JAN aims to transfer joint distributions of multi-layers' activation of the networks across domains. With the help of additional domain adaptation terms (e.g., MMD), DAN and JAN lead to a significant performance boost over the source-only-trained model (i.e.,
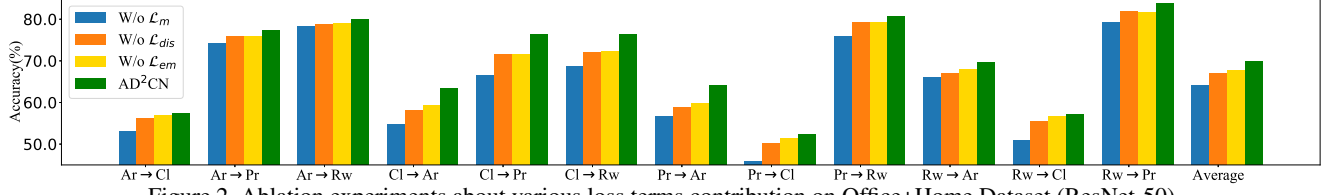
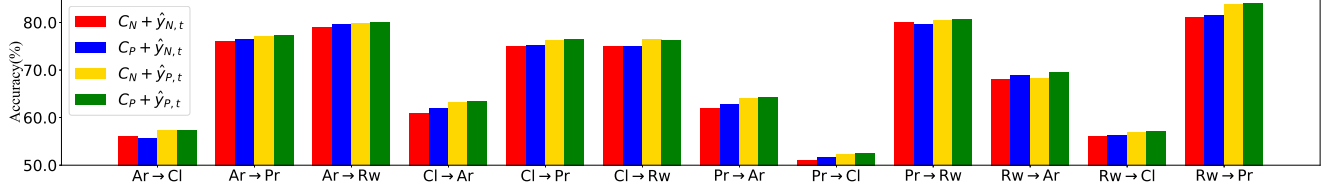Figure 2. Ablation experiments about various loss terms contribution on Office+Home Dataset (ResNet-50).

Figure 3. Accuracies of C_N and C_P on Office+Home. red and blue results are obtained with $\hat{\mathbf{y}}_{N,t}$ as target pseudo labels for $\mathcal{L}_m$, the others are based on $\hat{\mathbf{y}}_{P,t}$ as pseudo labels.

Table 3. C_N v.s. C_P accuracies (%) on Office+Home Ar $\to$ Cl

	Balanced			Imbalanced				
Y	Clock	Helmet	Knives	Bed	Couch	Folder	Marker	Pen
n_s	74	79	72	39	40	20	20	20
n_t	60	69	53	98	64	99	71	99
C_N	**75.0**	**71.0**	**52.8**	53.1	67.2	25.3	18.3	51.5
C_P	73.3	69.6	49.1	**55.1**	**68.8**	**28.3**	**21.1**	**53.5**

Table 4. Comparisons of Dual Classifiers Structure Influence to Recognition Rates (%) of Unsupervised Domain Adaptation on Office-31 Dataset (ResNet-50).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg.
MCD [34]	88.6	98.5	100.0	92.2	69.5	69.7	86.5
SWD [22]	90.4	98.7	100.0	94.7	70.3	70.5	87.4
Ours (same)	93.3	98.8	**100**	94.7	72.4	73.6	88.8
Ours	**93.6**	**98.9**	99.8	**95.4**	**74.9**	**75.0**	**89.6**

ResNet-50) on most adaptation tasks.

RevGrad implements adversarial networks and applies gradient reversal layer to train a domain discriminator. CDAN and MADA both exploit the multiplicative combination of feature embeddings and task-specific predictions as high-order representations to promote adversarial optimization. SE studies the self-ensembling to boost the visual domain adaptation performance. DSR assumes that the data generation process is controlled by the semantic latent variables and domain latent variables independently, so employs a variational auto-encoder in order to reconstruct them. MCS designs a unified framework without accessing the source data and iteratively assigns pseudo labels to the target samples by an alternating minimization scheme.

DWT-MEC proposes domain alignment layers with feature whitening to match source and target domain distributions and applies the Min-Entropy Consensus loss to unlabeled target data. AFN proposes a novel Adaptive Feature Norm approach to adapting the source and target domain feature norms to a specific range of values progressively.

SymNets exploits a novel adversarial classifiers networks and a two-level domain confusion scheme driving the learning of categories invariant intermediate features across domains. BDG bridges source and target domain through consistent classifiers interpolating two intermediate domains.

4.3. Ablation Analysis

In this section, we analyze the contribution and influence of several important terms and hyper-parameters sensitivity in our proposed model.

First, we discuss the influence of each component in our framework. By removing one of $\mathcal{L}_{dis}$, $\mathcal{L}_m$, and $\mathcal{L}_{em}$, while keeping other terms same as original AD^{2}CN, we obtain three variants AD^{2}CN w/o $\mathcal{L}_{dis}$, AD^{2}CN w/o $\mathcal{L}_m$, and AD^{2}CN w/o $\mathcal{L}_{em}$. From Fig. 2, we notice that all three components contribute to improving the domain adaptation performance, while our proposed discriminative cross-domain alignment $\mathcal{L}_m$ plays a more crucial role than others, i.e., discrepancy and entropy minimization loss.

Secondly, we compare the performances of $C_N(\cdot)$ and $C_P(\cdot)$ while accepting $\hat{\mathbf{y}}_{N,t}$ or $\hat{\mathbf{y}}_{P,t}$ as target domain pseudo labels for $\mathcal{L}_m$. From the results in Fig. 3, we observe that results with $\hat{\mathbf{y}}_{P,t}$ as pseudo labels are better than the results with $\hat{\mathbf{y}}_{N,t}$ in most cases. Compared to $C_N(\cdot)$, which is trained on the source domain, $C_P(\cdot)$ is based on the target prototypes and keeps better performance even on the early training stage. Fig. 4 shows several test samples that $C_P(\cdot)$ classifies correctly while $C_N(\cdot)$ cannot handle, which emphasizes the superiority of $C_P(\cdot)$.

Thirdly, we discuss the necessity and effectiveness of two different types of classifiers in our framework. Table 3 shows the selective target domain class-wise recognition accuracy on OfficeHome Ar $\to$ Cl case produced by the two classifiers C_N and C_P in our proposed model, as well as the number of samples in each class from the source and target domains. From the results we notice that for the categories having sufficient well labeled source sam-
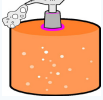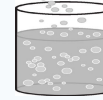
Y	Bottle	Candles	Chair	Desk Lamp	Drill	Knives	Notebook	Speaker	Soda	Webcam
C_N	Toys	Refrigerator	Telephone	LampShade	Hammer	Scissors	Table	Radio	Bottle	Speaker
C_P	Bottle	Candles	Chair	Desk Lamp	Drill	Knives	Notebook	Speaker	Soda	Webcam

Figure 4. Ten Samples from Office-Home Ar→Cl. Y row denotes the ground-truth labels, C_N row shows the mis-classified labels, while C_P means the correctly prediction.

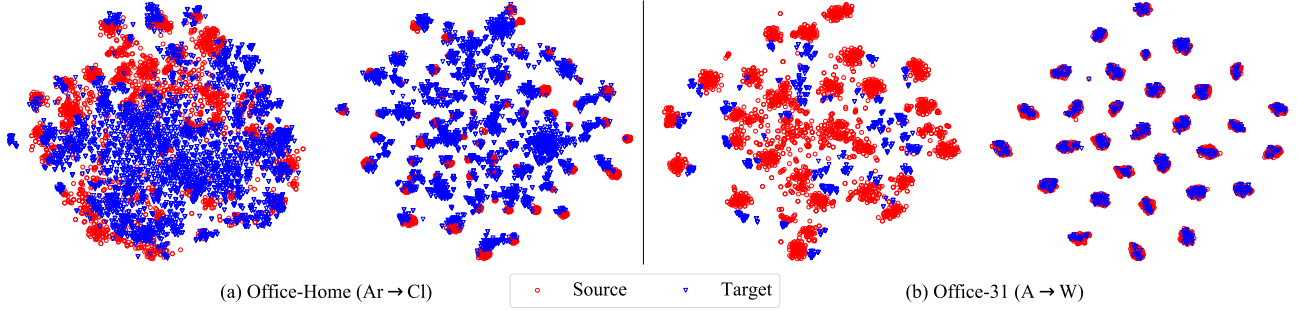(a) Office-Home (Ar → Cl)　　○ Source　▽ Target　　(b) Office-31 (A → W)

Figure 5. t-SNE visualization of source and target samples features before (left column) and after (right column) domain adaptation through our proposed model. (a) shows the task of Ar→Cl from Office-Home and (b) reports the task of A→W from Office-31.
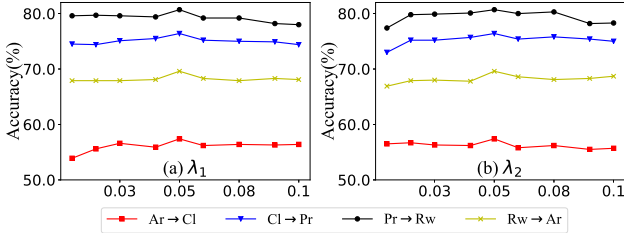

Figure 6. Parameters sensitivity analysis on 4 different tasks from Office-Home dataset of (a) λ_1 and (b) λ_2

ples as well as balanced target domain samples for training, C_N have better performance than C_P, while for other categories with imbalanced distribution across domains and insufficient labeled source samples for training, C_P always performs better than C_N. The observation proves that for imbalanced dataset, C_N and C_P have different speciality for different categories with various cross-domain distributions. More over, we show the comparison results of MCD [34], SWD[22], and our proposed model on Office-31 dataset in Table 4. MCD and SWD are two dual classifier adversarial frameworks for domain adaptation, but using two completely same structure neural networks classifiers. We also replace the C_N and C_P in our proposed model with two same structure neural networks classifiers and report the results as Ours(same). It is noteworthy that our proposed model achieves the best performance on most cases as well as the average accuracy compared to other same classifier structure methods, which proves the effectiveness and necessity of applying two distinct architecture classifiers.

Fourthly, we visualize the t-SNE embeddings (Fig. 5) of feature representations generated by $G(\cdot)$ before and af-

ter the domain adaptation through our proposed model, in which each category is represented as a cluster and different colors denote the different domains. Before adaptation, the source and target domains are totally mismatched, while our method shows the promising ability to make inter-class separated and intra-class clustered tightly.

Finally, we analyze the sensitivity of λ_1 (Fig. 6 (a))and λ_2 (Fig. 6 (b)) by listing four tasks from Office-Home dataset (Ar → Cl, Cl → Pr, Pr → Rw, Rw → Ar). Specifically, we set the ranges of λ_1 and λ_2 from 0.001 to 0.2, and evaluate one by fixing the other one as 0.1. From the results, we notice the accuracy curves are almost flat and stable, which indicates our proposed model is not sensitive to the values of λ_1 nor λ_2.

5. Conclusion

We presented a novel Adversarial Dual Distinct Classifier Networks (AD^{2}CN) for unsupervised domain adaptation to align source and target domain distribution discrepancy as well as task-specific category boundaries. Specifically, we designed two different architecture classifiers to detect target samples excluded by the source domain support by aligning the task-specific decision boundaries obtained by the two classifiers. Meanwhile, a domain-invariant feature generator was proposed to embed source and target domain data to a shared feature space under the guidance of discriminative cross-domain alignment. We evaluated our proposed model on two cross-domain visual benchmarks and obtained better performance over state-of-the-art methods, proving the effectiveness of our method.

References

[1] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[2] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI*, pages 2060–2066, 7 2019.

[3] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, pages 1081–1090, 2019.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[5] Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu. Graph adaptive knowledge transfer for unsupervised domain adaptation. In *ECCV*, pages 37–52, 2018.

[6] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

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

[9] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, pages 597–613, 2016.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[11] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *NIPS*, pages 513–520, 2007.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Learning an invariant hilbert space for domain adaptation. In *CVPR*, pages 3956–3965, 2017.

[14] Cheng-An Hou, Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Unsupervised domain adaptation with label and structural consistency. *TIP*, 25(12):5552–5562, 2016.

[15] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *WACV*, pages 749–757, 2020.

[16] Javed Iqbal and Mohsen Ali. Mlsl: Multi-level self-supervised learning for domain adaptation with spatially independent and semantically consistent labeling. In *WACV*, pages 1864–1873, 2020.

[17] Taotao Jing, Ming Shao, and Zhengming Ding. Discriminative cross-domain feature learning for partial domain adaptation. *arXiv preprint arXiv:2008.11360*, 2020.

[18] Taotao Jing, Haifeng Xia, and Zhengming Ding. Adaptively-accumulated knowledge transfer for partial domain adaptation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1606–1614, 2020.

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 12 2014.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[21] Vinod Kumar Kurmi, Shanu Kumar, and Vinay P Namboodiri. Attending to discriminative certainty for domain adaptation. In *CVPR*, pages 491–500, 2019.

[22] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, pages 10285–10295, 2019.

[23] Shuang Li, Chi Harold Liu, Binhui Xie, Limin Su, Zhengming Ding, and Gao Huang. Joint adversarial domain adaptation. In *ACM MM*, pages 729–737, 2019.

[24] Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *CVPR*, 2019.

[25] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[26] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NIPS*, pages 1640–1650, 2018.

[27] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, pages 2200–2207, 2013.

[28] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, pages 136–144, 2016.

[29] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017.

[30] Pietro Morerio, Riccardo Volpi, Ruggero Ragonesi, and Vittorio Murino. Generative pseudo-label refinement for unsupervised domain adaptation. In *WACV*, pages 3130–3139, 2020.

[31] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, 2018.

[32] Fabio Pizzati, Raoul de Charette, Michela Zaccaria, and Pietro Cerri. Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In *WACV*, pages 2990–2998, 2020.

[33] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *CVPR*, pages 9471–9480, 2019.

[34] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017.

[37] Yao-Hung Hubert Tsai, Cheng-An Hou, Wei-Yu Chen, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Domain-constraint transfer coding for imbalanced unsupervised domain adaptation. In *AAAI*, pages 3597–3603, 2016.

[38] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 7167–7176, 2017.

[39] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017.

[40] L Wang, B Sun, J Robinson, T Jing, and Y Fu. Ev-action: Electromyography-vision multi-modal action dataset. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*, pages 129–136.

[41] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, pages 1426–1435, 2019.

[42] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *CVPR*, pages 2272–2281, 2017.

[43] Guanglei Yang, Haifeng Xia, Mingli Ding, and Zhengming Ding. Bi-directional generation for unsupervised domain adaptation. In *AAAI*, pages 6615–6622, 2020.

[44] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, pages 3801–3809, 2018.

[45] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, pages 5031–5040, 2019.