Adversarial Partial Domain Adaptation by Cycle Inconsistency

Kun-Yu Lin[1◇], Jiaming Zhou[1◇], Yukun Qiu[1], and Wei-Shi Zheng[1,2,3,4‡]

[1] School of Computer Science and Engineering, Sun Yat-sen University, China
[2] Peng Cheng Laboratory, Shenzhen, China
[3] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
[4] Guangdong Province Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou
{linky5,zhoujm55,qiuyk}@mail2.sysu.edu.cn, wszheng@ieee.org

Abstract. Unsupervised partial domain adaptation (PDA) is a unsupervised domain adaptation problem which assumes that the source label space subsumes the target label space. A critical challenge of PDA is the negative transfer problem, which is triggered by learning to match the whole source and target domains. To mitigate negative transfer, we note a fact that, it is impossible for a source sample of outlier classes to find a target sample of the same category due to the absence of outlier classes in the target domain, while it is possible for a source sample of shared classes. Inspired by this fact, we exploit the cycle inconsistency, *i.e.*, category discrepancy between the original features and features after cycle transformations, to distinguish outlier classes apart from shared classes in the source domain. Accordingly, we propose to filter out source samples of outlier classes by weight suppression and align the distributions of shared classes between the source and target domains by adversarial learning. To learn accurate weight assignment for filtering out outlier classes, we design cycle transformations based on domain prototypes and soft nearest neighbor, where center losses are introduced in individual domains to reduce the intra-class variation. Experiment results on three benchmark datasets demonstrate the effectiveness of our proposed method.

Keywords: Unsupervised partial domain adaptation, negative transfer, cycle inconsistency

1 Introduction

Deep neural networks have achieved remarkable success in various machine learning problems and applications [31, 22, 41, 50, 17]. Usually, training a deep neural network requires large amounts of labeled data and assumes that the training data follow identical distribution as the test ones. Therefore, networks may degrade drastically when applied in new scenarios, where the training and test

◇ indicates equal contributions and ‡ indicates the corresponding author.

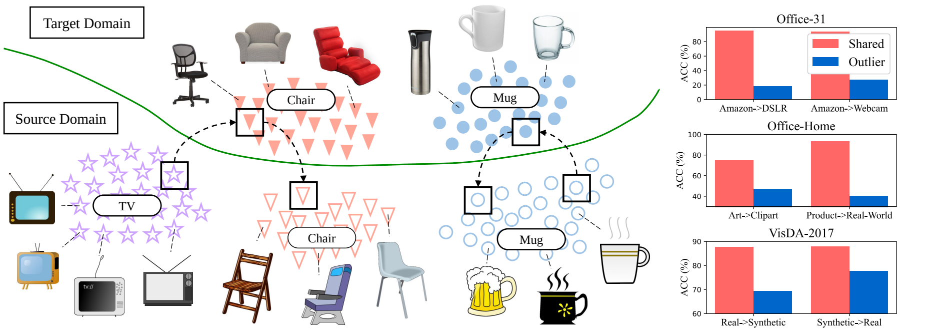
data follow very different distributions [69, 59]. By utilizing both labeled data from the source domain and unlabeled data from the target domain, *unsupervised domain adaptation (UDA)* [42, 15] attempts to safely transfer knowledge from the label-sufficient source domain to the label-free target domain, such that networks can generalize to the target domain. Existing UDA methods typically mitigate the distribution shift by minimizing the discrepancy between the source and target domains [15, 43, 59, 60].

UDA always assumes that the source and target domains share identical label space. Such an assumption would not be held in real-world applications since the labels of target data are unknown. In this paper, we focus on a variant of UDA, namely *unsupervised partial domain adaptation (PDA)*, which is challenging but realistic in real-world applications [4, 5, 72]. Compared with UDA, PDA does not assume that the source domain has identical label space as the target domain, but assumes that the target label space is a subset of the source label space. In this case, we term the classes absent in the target domain as *outlier classes* and the other classes as *shared classes.* A critical challenge of PDA is how to mitigate *negative transfer*, which causes that a transfer model performs even worse than a source-only model which is trained solely in the source domain.

We demonstrate an example about the negative transfer problem in PDA. Consider applying an adversarial learning method for standard UDA (*e.g.*, DANN [15]) in PDA scenarios. By confusing a domain discriminator, the adversarial learning method makes the distribution of the source domain (consisting of both shared and outlier classes) similar to the distribution of the target domain (consisting of only shared classes). However, in ideal cases, the distribution of source outlier classes should be dissimilar to the target distribution, since the outlier classes in the source domains are absent in the target domain. As a result, the adversarial learning method makes some target samples of the shared classes indistinguishable from the source samples of outlier classes, which hinders transferable discriminative feature learning and triggers negative transfer.

To mitigate the negative transfer problem, we attempt to match only the shared classes between the source and target domains. To this end, we should filter out source samples of outlier classes when applying adversarial learning. However, since no label information is available in the target domain, it is challenging to identify which classes are outlier for the target domain. To tackle the challenge, we explore the differences between outlier and shared classes. We note a fact that, it is *impossible* for a source sample of outlier classes to find a target sample of the same category due to *the absence of outlier classes in the target domain* while it is *possible* for a source sample of shared classes to find one, and it is also *possible* for a target sample to find a source sample of the same category since all target classes are shared across domains. Inspired by this fact, we develop a solution to distinguish outlier classes apart from shared classes based on *cycle inconsistency* modeling.

Specifically, we design a cycle transformation for source samples. The proposed cycle transformation first transforms a source sample feature into the target domain and then transforms it back into the source domain. We hold an



(a) Illustration of the Cycle Inconsistency

(b) Accuracy of Cycle Transformations

Fig. 1. (a) A PDA example with two shared classes and one outlier class. In PDA, due to the absence of outlier classes in the target domain, a source sample of outlier classes *cannot* find a target sample of the same category, while it is *possible* for a source sample of shared classes to find one. Inspired by this fact, we design a cycle transformation to distinguish outlier classes from shared classes. Our assumption is that, source samples of outlier classes are *more likely* to alter their category after cycle transformations compared with source samples of shared classes, with appropriate transformation functions. (b) Assumption verification using source-only models on three real-world datasets. We conduct cycle transformations on source samples by two cross-domain feature transformations, which is implemented by searching the most similar samples across domains in feature space. Then we calculate the accuracy of cycle transformations, *i.e.*, the proportion of samples to keep their categories after cycle transformations. The empirical results show that the accuracy of samples in shared classes is much higher than samples in outlier classes, which verifies our assumption. Best viewed in color.

assumption that, source samples of outlier classes are *more likely* to alter their category after cycle transformations compared with source samples of shared classes, with appropriate transformation functions (see empirical verification in Fig. 1). Accordingly, we propose a weighted adversarial learning method with a novel weight assignment scheme based on *cycle inconsistency*, *i.e.*, the category discrepancy between the original features and features after cycle transformations. Our method filters out source samples in outlier classes by sample weight suppression and aligns the distributions of shared classes between the source and target domains iteratively. With such a filter-and-align manner, our method gradually learns accurate transformation functions based on feature similarity for exploiting cycle inconsistency. For accurate sample weight assignment, we design cross-domain feature transformation functions based on domain prototypes and soft nearest neighbor to alleviate unexpected category alteration under large intra-class variation. For further improving the accuracy of cross-domain feature transformations, we adopt center losses within individual domains to reduce the intra-class variation. We conduct quantitative and qualitative experiments on three benchmark datasets, which demonstrates the effectiveness of our method.

2 Related Work

2.1 Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) is one of the most classical transfer learning tasks [48]. UDA aims to transfer knowledge from the label-sufficient source domain to the label-free target domain, such that the transfer model can generalize to the target domain. A critical challenge of the UDA problem is how to diminish the distribution discrepancy between the source and target domains.

Typically, existing UDA methods explore domain invariance based on feature alignment. A mainstream type of feature alignment methods explicitly minimizes well-defined statistical distances (*e.g.*, Maximum Mean Discrepancy) between the source and target domains [60, 42, 58, 71, 10]. Inspired by GANs [17], another mainstream type introduces an auxiliary domain discriminator and makes the feature extractor confuse the domain discriminator in an adversarial learning manner. Usually, these methods design different criteria for training domain discriminators [15, 59, 74, 66, 9]. Moreover, some works focus on specific differences between domains for implicit alignment [65, 28]. Häusser et al. propose to reinforce associations between domains directly in feature space [21, 30, 7].

In addition to methods based on feature alignment, generative methods introduce GANs for synthesizing labeled target data and align the two domains in both pixel and feature levels [1, 55, 27, 23, 47]. Among them, Hoffman et al. [23] propose to utilize the cycle consistency constraint for better synthesis without cross-domain pairs, inspired by CycleGAN [75]. Furthermore, some methods attempt to explore domain-specific information [53, 52, 37, 56, 44, 13]. For example, Saito et al. [52] and Liang et al. [37] assign pseudo labels to selected samples in the target domains, and Long et al. [44] and Shu et al. [56] apply the entropy minimization principle from the semi-supervised learning literature [18, 73].

Although UDA makes generalization in label-free domains possible, it is not realistic in real-world applications. UDA always assumes that the source and target domains share identical label space, which is too rigorous as label information of the target domain is unknown. Therefore, recent works make attempts to relax the assumption. For example, *open-set domain adaptation* assumes there are private classes in the target domain, which requires models to recognize both known and unknown classes in the target domain [3, 54, 40, 32]. *Universal domain adaptation* further assumes that both the source and target domains have private classes, leading to a large category gap between domains [70, 14, 34]. In addition, UDA usually assumes that both the source and target data are accessible, which violates the data privacy policy in some cases. Therefore, some works explore source-free settings with the source data unavailable [33, 36, 38, 68, 67].

2.2 Partial Domain Adaptation

Unsupervised partial domain adaptation (PDA) is an extreme case of imbalanced unsupervised domain adaptation [24]. PDA assumes that the source label space subsumes the target label space, which is more realistic than the standard

UDA. A critical challenge of PDA is the negative transfer problem triggered by learning to match the whole source and target domains. Typically, to mitigate negative transfer, existing methods introduce weighting schemes to filter out source samples of outlier classes and then apply adversarial learning methods for domain alignment. SAN [4] and PADA [5] propose class-wise weighting schemes according to the statistics of label predictors. IWAN [72] and ETN [6] assign sample-wise weights by introducing extra domain discriminators and label predictors, respectively. TWINs [46] proposes to estimate the label distribution using two classifiers. Apart from methods based on weighting schemes, there are PDA methods of other types. For example, Chen et al. [8] propose a reinforced data selector based on reinforcement learning, Liang et al. [39] propose a balanced and uncertainty-aware method which augments the small target domain to match the large source domain, Hu et al. [26] propose to maximize the distribution divergence between outlier and shared classes beyond aligning shared classes across domains, and Xiao et al. [63] propose to promote positive transfer by aligning the distributions of implicit semantic topics across domains.

In this paper, we propose a novel weighted adversarial learning method which filters out the source samples of outlier classes by cycle inconsistency. Existing weighted adversarial learning methods also make attempts to filter out the source samples of outlier classes. However, these methods usually calculate the weights of source samples *in indirect ways*, which apply extra auxiliary networks on top of features to infer the category gap between domains (*e.g.*, domain discriminator in IWAN [72], label predictor in ETN [6]). The quality of sample weights significantly depends on the performance of the auxiliary networks, as there are representation gaps between the feature extractor and auxiliary networks. By contrast, the proposed method exploits the category discrepancy between original and cycle-transformed features, which *directly* exploits the property of feature space without auxiliary networks and is more straightforward.

3 Methodology

In *unsupervised partial domain adaptation (PDA)*, the source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ consists of n_s labeled samples from $|\mathcal{C}_s|$ classes and the target domain $\mathcal{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$ consists of n_t unlabeled samples from $|\mathcal{C}_t|$ classes. The two domains follow *different but related* input distributions, *i.e.*, $p_s(\mathbf{x}_i^s) \neq p_t(\mathbf{x}_i^t)$, which is termed *domain gap*. Different from the standard *unsupervised domain adaptation (UDA)* problem, PDA further assumes a specific *category gap* between domains. In PDA, the label space of the target domain $\mathcal{C}_t$ is a subset of the label space of the source domain $\mathcal{C}_s$, *i.e.*, $\mathcal{C}_t \subset \mathcal{C}_s$. We term the classes absent in the target domain as outlier classes and the other classes as shared classes. The goal of PDA is to learn a generalizable model in the small target domain by transferring knowledge from the large source domain. In this paper, we aim to learn a transferable classification model, which is composed of a feature extractor $F : \mathcal{X} \rightarrow \mathcal{Z}$ and a label predictor $G : \mathcal{Z} \rightarrow \mathcal{Y}$.

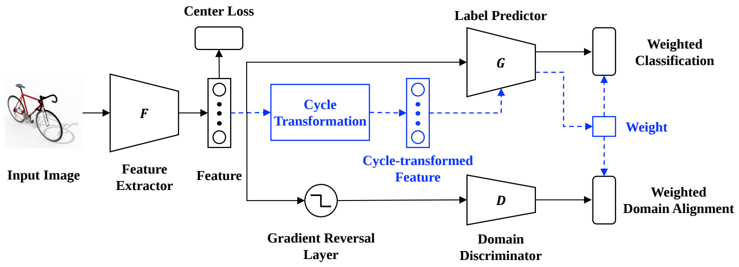


Fig. 2. An overview of the proposed weighted adversarial learning method based on cycle inconsistency. The blue components indicate the calculation process of sample weight, which is based on the category discrepancy between the original and cycle-transformed features. Three losses are involved during training, namely weighted classification, weighted domain alignment and center loss. Best viewed in color.

A critical challenge in PDA is the negative transfer problem, which is triggered by learning to match the whole source and target domains. To mitigate negative transfer, we design a cycle transformation consisting of two cross-domain feature transformations to distinguish outlier classes from shared classes. Our basic assumption is that, source samples of outlier classes are *more likely* to alter their category after cycle transformations compared with source samples of shared classes, with appropriate transformation functions. According to the assumption, we propose a weighted adversarial learning method with a cycle-inconsistency-based weighting scheme, which filters out source samples of outlier classes and aligns shared classes between the two domains iteratively. Next, we elaborate the proposed method, whose overview is given in Fig. 2.

3.1 Weighted Adversarial Learning for PDA

In this subsection, we illustrate the weighted adversarial learning framework. The framework is based on Domain Adversarial Neural Network (DANN) [15], which is one of the most widely used adversarial learning methods for the standard UDA problem. DANN introduces a domain discriminator and develops a two-player game for exploring domain invariance. Two losses are involved in DANN, namely classification losses for discriminating categories and domains, respectively. By confusing the domain discriminator, the feature extractor extracts domain-invariant features.

Although DANN is effective in UDA, it triggers the negative transfer problem in PDA. In principle, DANN implicitly learns to align the whole source and target domains. However, since there are outlier classes in the source domain in PDA, aligning the whole domains confuses some target samples with the source samples in outlier classes, leading to a loss of discriminative power in the target domain. Therefore, we should distinguish source samples of outlier classes apart from shared classes in adversarial learning. To this end, we assign a weight to each source sample in losses. Denoted the domain discriminator by $D : \mathcal{Z} \to \{0, 1\}$, the losses of weighted adversarial learning framework are given as follows:

$$\begin{aligned}\mathcal{L}^w_{cls}(\theta_f, \theta_g) &= \frac{1}{n_s}\sum_{i=1}^{n_s} w^s_i \mathcal{L}_{ce}(G(F(\mathbf{x}^s_i)), y^s_i) + \frac{\lambda_e}{n_t}\sum_{j=1}^{n_t} E(G(F(\mathbf{x}^t_j))),\\ \mathcal{L}^w_{adv}(\theta_f, \theta_d) &= \frac{1}{n_s}\sum_{i=1}^{n_s} w^s_i \mathcal{L}_{ce}(D(F(\mathbf{x}^s_i)), d^s_i) + \frac{1}{n_t}\sum_{j=1}^{n_t} \mathcal{L}_{ce}(D(F(\mathbf{x}^t_j)), d^t_j),\end{aligned} \tag{1}$$

where θ_f, θ_g and θ_d are parameters of the feature extractor F, label predictor G and domain discriminator D, respectively. $\mathcal{L}_{ce}(\cdot)$ is the cross-entropy loss, and $d^s_i = 0$ and $d^t_j = 1$ are domain labels. $E(\cdot)$ is the entropy function (entropy minimization encourages the low-density separation between classes [18]), and λ_e is a trade-off hyperparameter. w^s_i is the weight of the i-th source sample. Ideally, samples in shared classes have large weights and samples in outlier classes have zero weights. In this case, the adversarial learning ignores the source samples of outlier classes and aligns the distributions of shared classes between the two domains. Also, the weights are introduced in the classification loss for concentrating on classifying the shared classes. Next, we illustrate how to quantify the sample weights by *cycle inconsistency.*

3.2 Exploring Outlier Classes by Cycle Inconsistency

To quantify the sample weights, we exploit a key difference between source samples in shared and outlier classes by designing a cycle transformation. The cycle transformation consists of two cross-domain feature transformations by searching the most similar samples across domains in feature space. Specifically, given a source sample feature, we first transform it into the target domain and then transform it back into the source domain (*i.e.*, use the most similar feature cross domains as the transformed feature). We assume that, source samples of outlier classes are *more likely* to alter their category after cycle transformations compared with source samples of shared classes, if an appropriate similarity metric is learned. Empirically, we verify the assumption using source-only models as feature extractors on three real-world datasets, as shown in Fig. 1b. The assumption is inspired by a fact, *i.e.*, if a source sample belongs to outlier classes, applying the source-to-target transformation always alters its category. Since no label information is available in the target domain during training, we cannot verify the category alternation for filtering out outlier classes, thus we consider transforming the feature after source-to-target transformation back into the source domain.

According to the above assumption, we propose to exploit the *cycle inconsistency*, namely category discrepancy between the original and cycle-transformed features, to filter out source samples of outlier classes. Specifically, in each iteration, we assign weights to source samples based on the cycle inconsistency:

$$w^s_i = G(T_{t\to s}(T_{s\to t}(F(\mathbf{x}^s_i))))[y^s_i], \tag{2}$$

where $T_{s\to t} : \mathcal{Z} \to \mathcal{Z}$ and $T_{t\to s} : \mathcal{Z} \to \mathcal{Z}$ are functions for source-to-target and target-to-source feature transformations, respectively. $G(\mathbf{z})[c] : \mathcal{Z} \to [0, 1]$ denotes the c-th element of the classification probability vector given feature $\mathbf{z}$. In Eq. (2), we first extract the feature of $\mathbf{x}_i^s$ using the feature extractor F, then apply cycle transformation by $T_{s\to t}$ and $T_{t\to s}$, and finally get the cycle-transformed feature. We use the classifier G to predict the probability of category alternation after the cycle transformation as it is trained with labeled source samples. If the cycle-transformed feature has lower probability at its original category (*i.e.*, y_i^s), the sample $\mathbf{x}_i^s$ is more likely from the outlier classes (and vice versa).

Remark. The prerequisite of the proposed method is an appropriate feature similarity metric, based on which our method can find similar samples of the same categories across domains with acceptable accuracy for cross-domain feature transformations. In our method, the feature similarity metric is gradually learned by network training, and the cross-domain feature transformations gradually become more accurate as a result. The accuracy of cross-domain feature transformations is guaranteed by two factors. First, we use a source-only model as the initialization, since the model trained solely with source samples can distinguish target samples of different classes to some extent. Such an initialization scheme guarantees the accuracy of cross-domain similar sample search at the beginning of training (as shown in Fig. 1b). Second, our method uses a filter-and-align manner, which alternates between filtering out source samples of outlier classes by weight suppression and aligning the distributions of shared classes between the two domains by adversarial learning. As the training goes, our method gradually aligns shared classes across domains and thus the cross-domain similar sample search is gradually more accurate. In Sec. 3.3, we introduce prototype-based cross-domain feature transformation functions, which improve the accuracy of cross-domain similar sample search.

As the shared classes in the source and target domains gradually align during training, the classifier G gradually obtains classification power in the target domain. Therefore, if the classifier is confident in the prediction for the transformed feature $T_{s\to t}(F(\mathbf{x}_i^s))$, we can use the classifier to estimate the probability of category alternation after the source-to-target feature transformation $T_{s\to t}$, which contributes to filtering out source samples of outlier classes. Accordingly, by exploiting the category discrepancy between original features and features after source-to-target transformations, we propose a mixed strategy beyond cycle-inconsistency-based sample weighting, which is given as follows:

$$w_i^s = G(T_{t\to s}(T_{s\to t}(F(\mathbf{x}_i^s))))[y_i^s] + \lambda_w e_i^s G(T_{s\to t}(F(\mathbf{x}_i^s)))[y_i^s], \tag{3}$$

where λ_w is a trade-off hyperparameter and the entropy-aware weight $e_i^s = 1 + \exp\{-E(G(T_{s\to t}(F(\mathbf{x}_i^s))))\}$ indicates the classification confidence of the transformed feature $T_{s\to t}(F(\mathbf{x}_i^s))$. By using Eq. (3), our method considers the inconsistency in both cycle transformations and source-to-target transformations when the classifier is confident in its predictions, which contributes to more accurate sample weight assignment.

3.3 Prototype-based Cross-Domain Feature Transformation

In previous sections, we have introduced the cycle transformation consisting of two cross-domain feature transformations, which are implemented by searching the most similar sample across domains in the whole feature space. However, such an exhaustive searching process is too time-consuming and not practical for training. In addition, the exhaustive searching process will introduce noise into cross-domain feature transformations, especially when classes have large intra-class variation in feature space. For example, if a sample belongs to a class with large intra-class variation in feature space, its feature may fall close to the classification boundary (or even be misclassified). Therefore, for a sample of shared classes, the large variation improves the probability of finding a sample of another category in the cross-domain similar sample search. In ideal cases, the adopted transformation functions keep the categories of samples in shared classes. Besides, for samples of shared classes, variation between the original and cycle-transformed features is permitted if the category is not altered.

To this end, we propose an efficient and accurate cross-domain feature transformation method based on domain prototypes (dynamically updated) [62, 64, 37]. Specifically, to abstract the dataset, we obtain $|\mathcal{C}_s|$ and K domain prototypes in the source and target domains by class-wise feature mean and K-means clustering, respectively. Considering hyperparameter tuning in practice, we set $K = |\mathcal{C}_s|$ as the number of target classes is unknown. The sets of prototypes in the source and target domains are denoted by $\{\mathbf{c}_k^s\}_{k=1}^{|\mathcal{C}_s|}$ and $\{\mathbf{c}_k^t\}_{k=1}^{K}$, respectively. Given the domain prototypes, we conduct cross-domain feature transformations by using the most similar prototypes across domains, which fall away from the classification boundaries. And, the cost of one feature transformation comes from calculating sample similarity at the feature level for only $|\mathcal{C}_s|/K$ times. Therefore, we reduce the computation cost of the exhaustive search and the noise induced by the large intra-class variation. Furthermore, to improve the representation power of transformed features, we propose cross-domain feature transformation functions based on soft nearest neighbor [16, 57, 12] as follows:

$$T_{s\to t}(\mathbf{z}^s) = \sum_{k=1}^{K} \frac{e^{\mathrm{sim}(\mathbf{z}^s, \mathbf{c}_k^t)}}{\sum_{l=1}^{K} e^{\mathrm{sim}(\mathbf{z}^s, \mathbf{c}_l^t)}} \mathbf{c}_k^t, \; T_{t\to s}(\mathbf{z}^t) = \sum_{k=1}^{|\mathcal{C}_s|} \frac{e^{\mathrm{sim}(\mathbf{z}^t, \mathbf{c}_k^s)}}{\sum_{l=1}^{|\mathcal{C}_s|} e^{\mathrm{sim}(\mathbf{z}^t, \mathbf{c}_l^s)}} \mathbf{c}_k^s, \tag{4}$$

where $\mathbf{z}^s/\mathbf{z}^t$ denotes a feature vector in the source/target domain, and $\mathrm{sim}(\cdot, \cdot)$ is a function measuring the similarity between features. In our experiments, we adopt the negative square Euclidean distance as the similarity function for cross-domain feature transformations, *i.e.*, $\mathrm{sim}(\mathbf{z}^s, \mathbf{c}_k^t) = -\|\mathbf{z}^s - \mathbf{c}_k^t\|_2^2$.

In each training iteration, domain prototypes are updated using on-the-fly features in the current batch. Specifically, the updating rules are given as follows:

$$\mathbf{c}_k^s \leftarrow \lambda_m \mathbf{c}_k^s + \bar{\lambda}_m \frac{\sum_{i=1}^{B} \delta(y_i^s = k)\mathbf{x}_i^s}{\sum_{i=1}^{B} \delta(y_i^s = k)}, \; \mathbf{c}_k^t \leftarrow \lambda_m \mathbf{c}_k^t + \bar{\lambda}_m \frac{\sum_{j=1}^{B} \delta(\hat{y}_j^t = k)\mathbf{x}_j^t}{\sum_{j=1}^{B} \delta(\hat{y}_j^t = k)}, \tag{5}$$

where $\bar{\lambda}_m = 1 - \lambda_m$, y_i^s is the ground-truth label, $\hat{y}_j^t = \arg\max_{k=1}^{K} \mathrm{sim}(F(\mathbf{x}_j^t), \mathbf{c}_k^t)$ indicates the cluster assignment, B is the batch size, and λ_m is the momentum hyperparameter controlling the update rate. $\delta(\text{condition})$ is the indicator function, *i.e.*, $\delta(\text{condition}) = 1$ if the condition is satisfied and $\delta(\text{condition}) = 0$ otherwise. We do not update the prototypes absent in the current batch.

As discussed in previous works [62], classes will distribute in radial pattern in feature space, as the model is trained with a linear layer on top of the feature extractor and cross-entropy losses. Accordingly, the large intra-class variation in feature space negatively affects the source-to-target feature transformations and clustering in the target domain. Therefore, we adopt center losses within individual domains to reduce the intra-class variation, which are given as follows:

$$\mathcal{L}_{ctr}(\theta_f) = \frac{1}{2n_s}\sum\nolimits_{i=1}^{n_s} \|F(\mathbf{x}_i^s) - \mathbf{c}_{y_i^s}^s\|_2^2 + \frac{1}{2n_t}\sum\nolimits_{j=1}^{n_t} \|F(\mathbf{x}_j^t) - \mathbf{c}_{\hat{y}_j^t}^t\|_2^2. \tag{6}$$

By applying Eq. (6), each sample will be pushed closer to the corresponding prototype (*i.e.*, the ground-truth one in the source domain or the nearest one in the target domain) and classes will tend to distribute in sphere pattern in feature space. As a result, the center losses improve the compactness of classes and thus improve the accuracy of cross-domain feature transformations.

By cooperating the cycle-inconsistency-based weighting scheme with center losses, the overall objective of the proposed weighted adversarial learning method is given as follows:

$$\min_{\theta_f,\theta_g}\max_{\theta_d}\ \mathcal{L}_{cls}^w(\theta_f,\theta_g) - \lambda_{adv}\mathcal{L}_{adv}^w(\theta_f,\theta_d) + \lambda_{ctr}\mathcal{L}_{ctr}(\theta_f), \tag{7}$$

where λ_{adv} and λ_{ctr} are trade-off hyperparameters. The above minimax game is implemented by Gradient Reversal Layer [15], and the network parameters and domain prototypes are optimized in an alternating manner.

4 Experiment

4.1 Setups

- Datasets. We use three benchmark datasets in our experiments, namely Office-31, Office-Home and VisDA-2017. Office-31 [51] is a small-sized standard domain adaptation benchmark which consists of three domains, namely Amazon (A), DSLR (D) and Webcam (W). It contains 31 categories of objects in office setting, and the 10 categories shared with Caltech-256 [19] are taken as the target categories. Office-Home [61] is a medium-sized benchmark which consists of four domains, namely Artistic images (A), Clip Art (C), Product images (P), and Real-World images (R). It contains 65 categories of objects in office and home settings, and the first 25 categories (in alphabetical order) are taken as the target categories. VisDA-2017 [49] is a challenging large-scale dataset consisting of real (Re.) and synthetic (Sy.) images of 12 object categories, where the first 6 categories (in alphabetical order) are taken as the target categories. On VisDA-2017, we use center crop (rather than random crop) for training following previous works. There are 6, 12 and 2 transfer settings in these datasets, respectively. Classification accuracy (ACC) is used for evaluation.

Table 1. Comparison with the state-of-the-art methods on Office-31 and VisDA-2017 in terms of ACC (%). † indicates existing weighted adversarial learning methods for PDA. * indicates that the source-only model is used as initialization. The best result is marked as **bold red**, and the second best result is marked as *italic blue*.

Method	Office-31							VisDa-2017		
	A→D	A→W	D→A	D→W	W→A	W→D	Avg.	Re.→Sy.	Sy.→Re.	Avg.
ResNet-50 [22]	83.44	75.59	83.92	96.27	84.97	98.09	87.05	64.30	45.30	54.80
ADDA [59]	83.41	75.67	83.62	95.38	84.25	99.85	87.03	-	-	-
CDAN+E [43]	77.07	80.51	93.58	98.98	91.65	98.09	89.98	-	-	-
RTN [44]	66.90	75.30	85.60	97.10	85.70	98.30	84.80	72.90	50.00	61.45
†PADA [5]	89.17	88.70	94.61	99.77	95.79	**100.00**	94.67	69.46	62.76	66.11
†SAN [4]	94.27	93.90	94.15	99.32	88.73	99.36	94.96	69.70	49.90	59.80
†IWAN [72]	88.54	89.94	93.84	99.77	94.75	99.36	94.37	*78.18*	63.87	*71.02*
†ETN [6]	95.03	94.52	*96.21*	**100.00**	94.64	**100.00**	96.73	69.69	63.99	66.84
†MWPDA [25]	95.12	96.61	95.02	**100.00**	95.51	**100.00**	97.05	-	-	-
SSPDA [2]	90.87	91.52	90.61	92.88	94.36	98.94	93.20	-	-	-
DRCN [35]	86.00	88.05	95.60	**100.00**	95.80	**100.00**	94.30	73.20	58.20	65.70
RTNet [8]	*97.60*	96.20	92.30	**100.00**	95.40	**100.00**	96.90	-	-	-
BA3US [39]	**99.36**	*98.98*	94.82	**100.00**	94.99	98.73	*97.81*	-	-	-
DPDAN [26]	96.82	96.27	**96.35**	**100.00**	95.62	**100.00**	97.51	-	*65.26*	-
A2KT [29]	96.79	97.28	96.13	**100.00**	*96.14*	**100.00**	97.72	-	-	-
AdvRew [20]	91.72	97.63	95.62	**100.00**	95.30	**100.00**	96.71	-	-	-
Source-only	76.86	74.46	86.60	97.97	86.71	98.94	86.92	63.13	51.90	57.51
*DANN (baseline) [15]	59.24	56.84	70.22	82.60	86.19	90.45	74.25	50.09	44.02	47.05
†*PADA [5]	89.17	95.03	94.82	99.77	95.69	99.79	95.71	65.84	58.12	61.98
†*IWAN [72]	86.84	91.30	94.02	**100.00**	94.82	99.79	94.46	73.47	57.79	65.63
†*ETN [6]	84.71	87.23	94.08	98.76	94.57	98.73	93.01	67.42	60.87	64.15
Ours	96.82	**99.66**	96.14	**100.00**	**96.56**	**100.00**	**98.19**	**86.50**	**69.75**	**78.13**

- Existing methods. In our experiments, we compare the proposed method with both standard UDA and state-of-the-art PDA methods. Among existing PDA methods, we pay close attention to the methods based on weighted adversarial learning, which adopt different weighting schemes (*e.g.*, PADA [5] based on label predictor, IWAN [72] based on auxiliary domain discriminator). Apart from weighted adversarial learning methods, we also compare with PDA methods of other types, *e.g.*, RTNet [8], BA3US [39], DPDAN [26].

- Implementation details. We adopt ResNet-50 [22] pre-trained on ImageNet [11] as the backbone and add a bottleneck layer of dimension 256 between the backbone and classification layer. All network parameters are optimized using mini-batch SGD with batch size of 36, momentum of 0.9 and weight decay of 0.001. The learning rates of the bottleneck layer, classification layer and domain discriminator are 10 times that of the backbone, which are set as 0.001 initially and adjusted following the rules in previous works [15, 43]. By default, the hyperparameters are set as $\lambda_e = 0.1$, $\lambda_{adv} = 1$, $\lambda_{ctr} = 0.2$ and $\lambda_m = 0.99$. For the mixed strategy, $\lambda_w = 1$ and e_i^s is normalized within each batch. We initialize our model by the source-only model. On Office-Home, we adopt the source center loss after the source-only pre-training for one epoch. We normalize the sample weights in a batch of size B by $w_i^s \leftarrow Bw_i^s / \sum_{i=1}^{B} w_i^s$ following the previous works [72, 6].

4.2 Results

- Comparison with the state-of-the-arts. Tables 1 and 2 summarize the results on Office-31, Office-Home and VisDA-2017. Overall, the proposed method outperforms the state-of-the-art methods on all the three datasets, and Ours obtains the best or second best performance on most transfer settings. On VisDA-

Table 2. Comparison with the state-of-the-art methods and ablation study on Office-Home in terms of ACC (%). † indicates existing weighted adversarial learning methods for PDA. * indicates that the source-only model is used as initialization. The best result is marked as **bold red**, and the second best result is marked as *italic blue*.

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-50 [22]	46.33	67.51	75.87	59.14	59.94	62.73	58.22	41.79	74.88	67.40	48.18	74.17	61.35
ADDA [59]	45.23	68.79	79.21	64.56	60.01	68.29	57.56	38.89	77.45	70.28	45.23	78.32	62.82
CDAN+E [43]	47.52	65.91	75.65	57.07	54.12	63.42	59.60	44.30	72.39	66.02	49.91	72.80	60.73
SAFN [65]	58.93	76.25	81.42	70.43	72.97	77.78	72.36	55.34	80.40	75.81	60.42	79.92	71.83
†PADA [5]	49.31	71.95	82.09	57.73	58.86	65.03	67.03	41.87	83.60	79.55	52.12	84.37	66.13
†SAN [4]	44.42	68.68	74.60	67.49	64.99	77.80	59.78	44.72	80.07	72.18	50.21	78.66	65.30
†IWAN [72]	59.28	74.49	82.99	61.40	64.43	70.96	68.93	53.49	83.78	78.30	59.60	80.73	69.87
†ETN [6]	59.24	77.03	79.54	62.92	65.73	75.01	68.29	55.37	84.37	75.72	57.66	84.54	70.45
†MWPDA [25]	55.39	77.53	81.27	57.08	61.03	62.33	68.74	56.42	86.67	76.70	57.67	80.06	68.41
SSPDA [2]	51.95	67.00	78.74	52.16	53.78	59.03	52.61	43.22	78.79	73.73	56.60	77.09	62.06
DRCN [35]	54.00	76.40	83.00	62.10	64.50	71.00	70.80	49.80	80.50	77.50	59.10	79.90	69.00
RTNet [8]	*63.20*	80.10	80.70	66.70	69.30	77.20	71.60	53.90	84.60	77.40	57.90	85.50	72.30
BA3US [39]	60.62	83.16	88.39	71.75	72.79	83.40	75.45	61.59	86.53	79.25	62.80	86.05	75.98
DPDAN [26]	59.40	-	79.04	-	-	-	-	-	81.79	76.77	58.67	82.18	-
A2KT [29]	62.54	83.92	86.69	65.44	74.96	75.04	67.40	55.14	84.37	73.25	60.51	84.09	72.78
AdvRew [20]	62.13	79.22	89.12	73.92	75.57	**84.37**	78.42	61.91	87.85	82.19	*65.37*	85.27	77.11
Source-only	46.45	69.04	79.79	57.45	58.04	65.54	59.35	38.23	76.31	69.76	45.27	76.06	61.77
*DANN (baseline) [15]	47.22	58.71	71.67	48.45	44.50	54.74	53.38	40.48	69.57	63.09	47.74	71.02	55.88
†*PADA [5]	49.97	70.78	82.18	59.44	59.35	66.91	68.84	44.78	83.42	78.70	55.02	84.33	66.98
†*IWAN [72]	59.34	81.49	85.64	68.07	71.75	74.51	71.84	57.15	83.86	77.32	62.37	83.16	73.04
†*ETN [6]	52.78	70.84	78.29	69.54	69.76	73.37	63.12	50.10	74.47	75.18	55.07	79.23	67.65
Ours-Cycle w/o $\mathcal{L}_{ctr}$	62.45	84.71	89.18	*76.40*	75.57	77.75	77.04	59.70	86.86	82.37	62.87	*85.77*	76.72
Ours-Cycle	62.51	*85.71*	90.17	74.75	75.57	82.66	77.96	62.87	86.36	*84.76*	63.76	85.60	*77.72*
Ours-Cycle-Hard	**64.84**	84.99	**90.72**	75.30	*75.69*	82.83	77.23	*63.10*	85.42	80.62	63.64	84.87	77.44
Ours-Src2Trg	60.48	85.66	89.23	73.92	72.89	79.85	**80.72**	56.72	**88.57**	80.26	62.75	84.99	76.33
Ours (Full)	61.73	**86.89**	*90.50*	**77.23**	**76.86**	*83.77*	*79.61*	**63.82**	*88.46*	**85.03**	**65.79**	**86.22**	**78.83**
Oracle	69.19	82.75	88.99	75.94	76.88	83.99	77.29	66.19	90.06	84.14	74.33	91.04	80.07

2017, our method obtains significant improvement over the state-of-the-arts, *i.e.*, 8.32% on Re.→Sy. and 4.49% on Sy.→Re., respectively. In addition, Ours outperforms all existing weighted adversarial learning methods (marked by †, *e.g.*, PADA [5], IWAN [72], etc.), which demonstrates the superiority of the proposed cycle-inconsistency-based sample weighting scheme. We also make a comparison with weighted adversarial learning methods using the source-only model as initialization (*i.e.*, *PADA, *IWAN and *ETN). Although such an initialization brings performance improvement in some cases (*e.g.*, *IWAN vs. IWAN on Office-Home), our method still outperforms them, which demonstrates that our performance improvement does not come from the initialization.

- Ablation study. On Office-Home, we conduct an in-depth analysis of our model components. We first compare the source-only model with DANN [15], a classical UDA method based on adversarial learning. Although using source-only models as initialization, DANN obtains worse performance, which is caused by negative transfer. By introducing the proposed cycle-inconsistency-based weighting scheme, our method (Ours-Cycle w/o $\mathcal{L}_{ctr}$) obtains significant performance improvement over the source-only model. The result demonstrates the effectiveness of the proposed cycle-inconsistency-based weighting scheme, which mitigates negative transfer. By further introducing center losses, Ours-Cycle obtains higher performance, which attributes to the reduction of intra-class variation. By using the vanilla nearest neighbor (prototype) search for cross-domain feature transformations, Ours-Cycle-Hard obtains slightly lower performance compared with that using the soft nearest neighbor (Ours-Cycle), which is because that the soft-nearest-neighbor-based transformations improve the representation

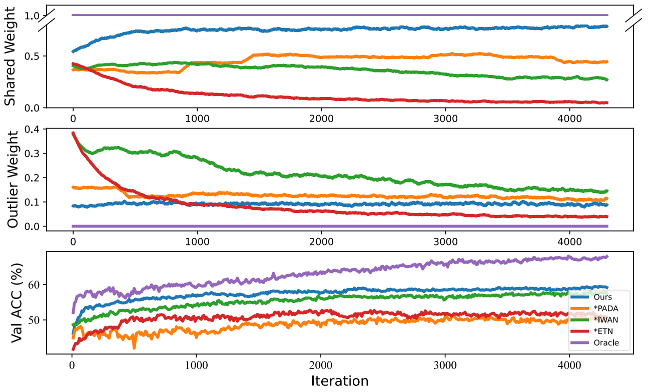
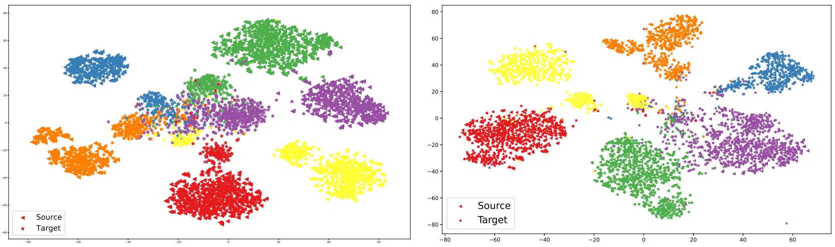


Fig. 3. Quantitative analysis of our method and existing weighted adversarial learning methods based on mean sample weights and validation ACCs during training on the A→C setting of Office-Home. Best viewed in color.

power of transformed features. If we cancel the cycle-back operation in the cycle-inconsistency-based weighting scheme (Ours-Src2Trg), the performance drops (*e.g.*, A→C, P→C), which demonstrates the effectiveness of cycle inconsistency modeling. This is because the model has weak classification power in the target domain at the early stages of training, resulting in inaccurate weight assignment. Furthermore, by adopting the mixed strategy, Ours (Full) obtains further improvement since it considers the inconsistency in both cycle transformations and source-to-target transformations. Also, we report the results of the oracle method (Oracle) as the upper bound, *i.e.*, *an ideal weighted adversarial learning method which assigns sample weights according to the ground-truth.* Compared with all existing methods, our performance is the closest to that of Oracle.

- Analysis of sample weights. Fig. 3 shows variation curves of the mean sample weight of our method during training on the A→C setting of Office-Home (the **blue** curves). Overall, our method assigns large weights to samples of shared classes and small weights to samples of outlier classes. At the beginning of training, the sample weights of shared classes are moderate (about 0.5) and the sample weights of outlier classes are small (about 0.1). As the training goes, the sample weights of shared classes gradually increase (from 0.5 to 0.8) while the sample weights of outlier classes keep stably low, and the validation ACC gradually increases as the shared classes across domains gradually align. Also, we compare our method with existing weighted adversarial learning methods, and we report the results of the oracle method (Oracle) as the upper bound. From the figure, we find that our method assigns much more accurate weights to samples compared with existing methods (*i.e.*, *PADA, *IWAN, *ETN). Specifically, our method assigns much higher weights to samples in shared classes and assigns relatively lower weights (with respect to shared weights) to samples in



(a) Source-only

(b) Ours

Fig. 4. Feature distribution by t-SNE of the (a) source-only model and (b) our model on VisDA-2017. The triangle and star markers denote the source and target samples, and different colors denote different categories. For better visualization, we only show the shared classes. Best viewed in color.

outlier classes. As a result, our method obtains higher validation ACC. Besides, our method performs closely with Oracle in terms of mean sample weights and obtains the closest performance to the upper bound compared with others.

- Feature distribution visualization. Fig. 4 shows the feature distribution by t-SNE [45]. From Fig. 4a, we find that the source-only model can discriminate target samples of different classes to some extent. However, the source-only model does not align the shared classes in the source and target domains and confuse the target samples from different classes at the central area. By contrast, as shown in Fig. 4b, our model aligns the two domains well and the classification boundaries are much clearer in the target domain.

5 Conclusion

To address unsupervised partial domain adaptation, in this work, we propose a weighted adversarial learning method with a novel sample weighting scheme. Our method exploits the cycle inconsistency, i.e., category discrepancy between original and cycle-transformed features, to distinguish outlier classes from shared classes in the source domain. Accordingly, our method filters out the source samples of outlier classes and aligns shared classes across domains iteratively. With such a filter-and-align manner, our method gradually learns accurate cycle transformation functions based on feature similarity. Extensive experiments demonstrate the effectiveness of our method. In the future, we will explore properties of the feature space in more practical settings, *e.g.*, universal domain adaptation.

Acknowledgements. This work was supported partially by the NSFC (U21A2-0471, U1911401, U1811461), Guangdong NSF Project (No. 2020B1515120085, 2018B030312002), Guangzhou Research Project (201902010037), and the Key-Area Research and Development Program of Guangzhou (202007030004).

References

1. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 95–104 (2017)
2. Bucci, S., D'Innocente, A., Tommasi, T.: Tackling partial domain adaptation with self-supervision. In: International Conference on Image Analysis and Processing. vol. 11752, pp. 70–81 (2019)
3. Busto, P.P., Gall, J.: Open set domain adaptation. In: IEEE International Conference on Computer Vision. pp. 754–763 (2017)
4. Cao, Z., Long, M., Wang, J., Jordan, M.I.: Partial transfer learning with selective adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2724–2732 (2018)
5. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: Proceedings of the European Conference on Computer Vision. pp. 139–155 (2018)
6. Cao, Z., You, K., Long, M., Wang, J., Yang, Q.: Learning to transfer examples for partial domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2985–2994 (2019)
7. Chen, C., Li, J., Zheng, Z., Huang, Y., Ding, X., Yu, Y.: Dual bipartite graph learning: A general approach for domain adaptive object detection. In: IEEE/CVF International Conference on Computer Vision. pp. 2683–2692 (2021)
8. Chen, Z., Chen, C., Cheng, Z., Jiang, B., Fang, K., Jin, X.: Selective transfer with reinforced transfer network for partial domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12703–12711 (2020)
9. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12452–12461 (2020)
10. Damodaran, B.B., Kellenberger, B., Flamary, R., Tuia, D., Courty, N.: DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation. In: Proceedings of the European Conference on Computer Vision. pp. 467–483 (2018)
11. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
12. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: Temporal cycle-consistency learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1801–1810 (2019)
13. French, G., Mackiewicz, M., Fisher, M.H.: Self-ensembling for visual domain adaptation. In: International Conference on Learning Representations (2018)
14. Fu, B., Cao, Z., Long, M., Wang, J.: Learning to detect open classes for universal domain adaptation. In: Proceedings of the European Conference on Computer Vision. pp. 567–583 (2020)
15. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**, 59:1–59:35 (2016)
16. Goldberger, J., Roweis, S.T., Hinton, G.E., Salakhutdinov, R.: Neighbourhood components analysis. In: Advances in Neural Information Processing Systems. pp. 513–520 (2004)
17. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)

18. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in Neural Information Processing Systems. pp. 529–536 (2004)
19. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007)
20. Gu, X., Yu, X., Yang, Y., Sun, J., Xu, Z.: Adversarial reweighting for partial domain adaptation. In: Advances in Neural Information Processing Systems (2021)
21. Häusser, P., Frerix, T., Mordvintsev, A., Cremers, D.: Associative domain adaptation. In: IEEE International Conference on Computer Vision. pp. 2784–2792 (2017)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
23. Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: CyCADA: Cycle-consistent adversarial domain adaptation. In: Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 1994–2003 (2018)
24. Hsu, T.H., Chen, W., Hou, C., Tsai, Y.H., Yeh, Y., Wang, Y.F.: Unsupervised domain adaptation with imbalanced cross-domain data. In: IEEE International Conference on Computer Vision. pp. 4121–4129 (2015)
25. Hu, J., Tuo, H., Wang, C., Qiao, L., Zhong, H., Jing, Z.: Multi-weight partial domain adaptation. In: 30th British Machine Vision Conference. p. 5 (2019)
26. Hu, J., Tuo, H., Wang, C., Qiao, L., Zhong, H., Yan, J., Jing, Z., Leung, H.: Discriminative partial domain adversarial network. In: Proceedings of the European Conference on Computer Vision. pp. 632–648 (2020)
27. Hu, L., Kan, M., Shan, S., Chen, X.: Duplex generative adversarial network for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1498–1507 (2018)
28. Jin, Y., Wang, X., Long, M., Wang, J.: Minimum class confusion for versatile domain adaptation. In: Proceedings of the European Conference on Computer Vision. pp. 464–480 (2020)
29. Jing, T., Xia, H., Ding, Z.: Adaptively-accumulated knowledge transfer for partial domain adaptation. In: ACM International Conference on Multimedia. pp. 1606–1614 (2020)
30. Kang, G., Wei, Y., Yang, Y., Zhuang, Y., Hauptmann, A.G.: Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems (2020)
31. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1106–1114 (2012)
32. Kundu, J.N., Venkat, N., Revanur, A., V., R.M., Babu, R.V.: Towards inheritable models for open-set domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12373–12382 (2020)
33. Kundu, J.N., Venkat, N., V., R.M., Babu, R.V.: Universal source-free domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4543–4552 (2020)
34. Li, G., Kang, G., Zhu, Y., Wei, Y., Yang, Y.: Domain consensus clustering for universal domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 9757–9766 (2021)
35. Li, S., Liu, C.H., Lin, Q., Wen, Q., Su, L., Huang, G., Ding, Z.: Deep residual correction network for partial domain adaptation. IEEE Trans. Pattern Anal. Mach. Intell. **43**(7), 2329–2344 (2021)

36. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In: Proceedings of the 37th International Conference on Machine Learning. vol. 119, pp. 6028–6039 (2020)
37. Liang, J., Hu, D., Feng, J.: Domain adaptation with auxiliary target domain-oriented classifier. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 16632–16642 (2021)
38. Liang, J., Hu, D., Wang, Y., He, R., Feng, J.: Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
39. Liang, J., Wang, Y., Hu, D., He, R., Feng, J.: A balanced and uncertainty-aware approach for partial domain adaptation. In: Proceedings of the European Conference on Computer Vision. pp. 123–140 (2020)
40. Liu, H., Cao, Z., Long, M., Wang, J., Yang, Q.: Separate to adapt: Open set domain adaptation via progressive separation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2927–2936 (2019)
41. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440 (2015)
42. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on Machine Learning. vol. 37, pp. 97–105 (2015)
43. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 1647–1657 (2018)
44. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: Advances in Neural Information Processing Systems. pp. 136–144 (2016)
45. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research **9**(11) (2008)
46. Matsuura, T., Saito, K., Harada, T.: TWINs: Two weighted inconsistency-reduced networks for partial domain adaptation. CoRR **abs/1812.07405** (2018)
47. Murez, Z., Kolouri, S., Kriegman, D.J., Ramamoorthi, R., Kim, K.: Image to image translation for domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4500–4509 (2018)
48. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
49. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: VisDA: The visual domain adaptation challenge. CoRR **abs/1710.06924** (2017)
50. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. pp. 91–99 (2015)
51. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Proceedings of the European Conference on Computer Vision. pp. 213–226 (2010)
52. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 2988–2997 (2017)
53. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3723–3732 (2018)

54. Saito, K., Yamamoto, S., Ushiku, Y., Harada, T.: Open set domain adaptation by backpropagation. In: Proceedings of the European Conference on Computer Vision. pp. 156–171 (2018)
55. Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R.: Generate to Adapt: Aligning domains using generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 8503–8512 (2018)
56. Shu, R., Bui, H.H., Narui, H., Ermon, S.: A DIRT-T approach to unsupervised domain adaptation. In: International Conference on Learning Representations (2018)
57. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. pp. 4077–4087 (2017)
58. Sun, B., Saenko, K.: Deep CORAL: Correlation alignment for deep domain adaptation. In: ECCV 2016 Workshops. pp. 443–450 (2016)
59. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2962–2971 (2017)
60. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep Domain Confusion: Maximizing for domain invariance. CoRR **abs/1412.3474** (2014)
61. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5385–5394 (2017)
62. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Proceedings of the European Conference on Computer Vision. pp. 499–515 (2016)
63. Xiao, W., Ding, Z., Liu, H.: Implicit semantic response alignment for partial domain adaptation. In: Advances in Neural Information Processing Systems (2021)
64. Xie, S., Zheng, Z., Chen, L., Chen, C.: Learning semantic representations for unsupervised domain adaptation. In: Proceedings of the 35th International Conference on Machine Learning. pp. 5419–5428 (2018)
65. Xu, R., Li, G., Yang, J., Lin, L.: Larger Norm More Transferable: An adaptive feature norm approach for unsupervised domain adaptation. In: IEEE/CVF International Conference on Computer Vision. pp. 1426–1435 (2019)
66. Yang, J., Zou, H., Zhou, Y., Zeng, Z., Xie, L.: Mind the Discriminability: Asymmetric adversarial domain adaptation. In: Proceedings of the European Conference on Computer Vision. pp. 589–606 (2020)
67. Yang, S., Wang, Y., van de Weijer, J., Herranz, L., Jui, S.: Exploiting the intrinsic neighborhood structure for source-free domain adaptation. CoRR **abs/2110.04202** (2021)
68. Yang, S., Wang, Y., van de Weijer, J., Herranz, L., Jui, S.: Generalized source-free domain adaptation. In: IEEE/CVF International Conference on Computer Vision. pp. 8978–8987 (2021)
69. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems. pp. 3320–3328 (2014)
70. You, K., Long, M., Cao, Z., Wang, J., Jordan, M.I.: Universal domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2720–2729 (2019)
71. Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., Saminger-Platz, S.: Central moment discrepancy (CMD) for domain-invariant representation learning. In: International Conference on Learning Representations (2017)

72. Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 8156–8164 (2018)
73. Zhang, Y., Zhang, H., Deng, B., Li, S., Jia, K., Zhang, L.: Semi-supervised models are strong unsupervised domain adaptation learners. CoRR **abs/2106.00417** (2021)
74. Zhang, Y., Liu, T., Long, M., Jordan, M.I.: Bridging theory and algorithm for domain adaptation. In: Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 7404–7413 (2019)
75. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision. pp. 2242–2251 (2017)