Meta Adversarial Weight for Unsupervised Domain Adaptation

Chang Liu † Lichen Wang† Yun Fu†

Abstract

Despite great progress in supervised image recognition, a large performance drop is usually observed when deploying the model in the wild. Unsupervised domain adaptation (UDA) methods tackle the issue by aligning the source domain and the target domain. However, most existing adversarial based methods attempt to perform the alignment from a holistic view, ignoring the underlying class-level data structure in the target domain. As a result, the representations are distorted by adversarial alignment, leading to a negative transfer. Motivated by this issue, we first claim that this issue can be solved if there exists 'optimal' per-sample weights for adversarial alignment, and then devise a meta-learning framework to adaptively learn such adversarial weights. Specifically, we construct a meta-dataset with target-like distribution as meta knowledge, and use it to guide the learning of the optimal adversarial weights via a meta-learner. By this means, our framework can adaptively adjust the weights of all training samples in adversarial training based on the feedback from meta dataset and thus achieve the categorical-wise domain alignment. We conduct sufficient ablation studies and experiments to show the effectiveness of our approach. Our method is generic to existing domain alignment based methods and could achieve consistently improvements over three UDA classification benchmarks.

Keywords: Domain Adaptation, Adversarial Learning, Transfer Learning, Semi-supervised Learning

1 Introduction

The success of deep neural networks in recent years is mostly driven by a large amount of accessible labeled data. However, it is usually labor-intensive work to collect massive densely annotated data. To address this issue, unsupervised domain adaption (UDA) alleviates the dependency on large-scale labeled training datasets by transferring knowledge from relevant source domains with rich labeled data, e.g synthetic data via computer graphics technology. However, a performance drop is observed when the model trained with source domain data is applied to target domain data due to distribution discrepancy between source and target domain. This phenomenon is known as the domain shift problem, which poses a challenge to UDA as shown in Figure 1.

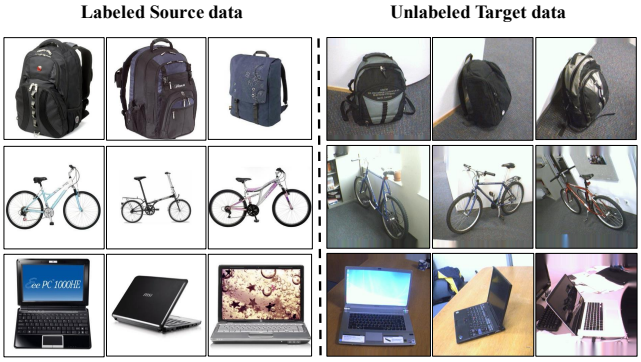

Figure 1: Examples from Office-31 [17] dataset for unsupervised domain adaptation (UDA). For training, labeled source data (Amazon) and unlabeled target data (Webcam) are used. For testing, only target data is evaluated. The source and target domain data share the same label space. The goal of UDA is to reduce the domain shift problem to learn a model that generalizes well on target data.

There are two directions on UDA for closing the discrepancy gap via aligning the feature distributions of the source and the target domain. One direction of UDA focuses on directly applying semi-supervised learning (SSL) techniques on UDA problems [28, 8] by considering target domain as the unlabeled data in SSL. Recently, pseudo-labeling-based (PL) methods [25] have been widely adopted from SSL and achieved state-of-the-art performance on UDA tasks. A typical way of PL is to generate pseudo-labels corresponding to the largest prediction probability of target samples and retrain the network on them in an iterative and supervised manner. However, as the network is biased towards source labeled data, the pseudo labels unavoidably contain noises. Retraining on those noisy supervision would propagate errors.

Another direction is to learn domain-invariant representations by alignment-based methods. Among them, adversarial alignment [20, 12] has been known as one of the most popular methods which reduce the discrepancy between two domains by training a domain-invariant feature extractor. The key idea is to play a two-player minimax game between a feature extractor and domain discriminator. Specifically, the domain discriminator is optimized to distinguish the samples from source and target domain while the feature extractor is trained to fool the domain discriminator. Though tremendous progress has been made in adversarial domain adaptation, the

†Northeastern University (liu.chang6@northeastern.edu, wanglichenxj@gmail.com, yunfu@ece.neu.edu)

major limitation is that it can only align the feature distributions globally without considering categorical information. As a result, the adversarial alignment would distort the discriminative structures of the original representations, leading to a negative transfer (aligning samples of two domains but different class together).

Many subsequent works [27, 2] try to address this issue by introducing their newly designed regularization. Specifically, [2] adds batch spectral regularization to suppress the top singular values such that eigenvectors with smaller singular values can also carry discriminability information. [27] proposes a asymmetric adversarial mechanism with a new domain classifier as regularization. [12] incorporates class prediction with feature to learn the domain discriminator and design an entropy-based weighting strategy to align the samples in a easy-to-hard manner.

Following the spirit of [12] in terms of sample weighting, we claim that the feature distortion issue caused by adversarial learning can be solved if there are "ideal" per-sample weights existed for adversarial alignment. Importantly, the 'ideal' adversarial weights should have following properties: 1) it will assign large weights to those poorly-aligned samples and small or even zero weights to those well-aligned samples. 2) For those poorly-aligned samples, the value of weight should be adaptively adjusted such that they can be aligned to the correct semantic class. However, none of existing methods can estimate such 'ideal' adversarial weights.

In this paper, we devise a Meta-learning framework to adaptively learn such ideal adversarial weights with the guidance from meta knowledge. The main idea is to build up a meta-dataset with target distribution as meta knowledge, and leverage it to guide the learning of a meta-leaner which takes each sample as input and outputs its estimated adversarial weights. Intuitionally, the meta-learner can be connected with original model in the sense that, if adversarial weights generated by the meta-learner are optimal, then the model trained with such weights for domain alignment should have low risk on the meta dataset. Formally, we formulate this problem in a bi-level optimization manner: 1) the meta-learner is iteratively optimized by the meta data to output the optimal adversarial weights. 2) the model is optimized by training data with the estimated optimal adversarial weights from the meta-learner for better domain alignment. Additionally, as target data does not have labels in UDA problems, we create our meta dataset with clean target-like distribution by mixing up source samples with high-quality pseudo-labeled target samples of the same class. This meta dataset can provide the meta-knowledge to the meta-learner for the better estimation of adversarial weights.

We summarize our contributions as follows:

- Motivated by the limitations of adversarial learning in terms of representation distortion, we propose a Meta-learning framework to solve the issue by learning "optimal" per-sample weights for adversarial alignment in UDA classification task. To our best knowledge, we conduct the first attempt to model a **learnable** per-sample weight for adversarial alignment and solve it in a learning-to-learn manner.
- To learn the optimal adversarial weights, we construct a meta dataset with target-like distribution as meta-knowledge, and use it to guide the learning of a meta-leaner which takes training samples as input and outputs their adversarial weights. The meta-learner and model are connected in a bi-level optimization manner.
- We conduct extensive experiments and ablation studies to thoroughly verify the effectiveness of our proposed Meta Adversarial Weight. Furthermore, our method is generic and can be plugged into various domain alignment based UDA methods to boost their performance.

2 Related Work

2.1 Discrepancy-based domain adaptation Many works focus on extracting domain-invariant feature representations. Some previous works suggest different metrics to measure domain discrepancy explicitly and promote the feature alignment by minimizing these measure like MMD [21]. With the popularity of Generative Adversarial Network, a trend of adversarial domain adaptation is rising [5, 12]. Specifically, adversarial training based methods involve a domain discriminator as an implicit way to measure domain discrepancy and domain adaptation is achieved by encouraging the network to confuse the well-trained discriminator to generate domain-invariant features. However, adversarial alignment without proper regularization would distort the feature representations. Several previous methods try to solve this issue with their designed regularization loss. Specifically, [2] adds batch spectral penalization to suppress the top singular values such that eigenvectors with smaller singular values can also carry discriminability information. [18] uses a contrastive loss to achieve instance-level transfer for better discriminability.

In this paper, we solve the representation distortion from a new perspective by estimating the optimal per-sample weights for adversarial alignment in a learning-to-learn manner.

2.2 Pseudo-labeling in Domain Adaptation Inspired by cluster assumption, pseudo-labeling can realize the class-wise alignment across domains. Specifically, it iteratively generates pseudo-labels for the target samples with high prediction probability and retrain the network based on those pseudo-labels along with labeled source data. This technique has been widely employed for UDA on classification [25] and semantic segmentation [31] tasks. However, pseudo-labeling methods do not have a theoretical guarantee for reducing domain discrepancy and usually suffer from noisy target

pseudo labels which would mislead the network training.

In this paper, we combines the merits of adversarial learning and pseudo-labeling in the sensen that our method could not only minimize the domain discrepancy across domains with theoretical guarantee, but also maintains the discriminative structures by our proposed meta adversarial weights.

2.3 Meta-learning in DA Meta-Learning (learning to learn) has experienced a recent resurgence in few-shots learning (e.g MAML[4]). The goal of MAML is to learn a good network initialization for unseen few-shot tasks by training on several different tasks. Following the spirits of MAML, meta-learning is introduced in transfer learning setting by Li et al. [10, 9] to solve domain generalization, multi-source DA, and semi-supervised DA setting. Specifically, they split the different domains of labeled training set into meta-train set and met-test set, and optimize the network in a two-stage meta-learning manner for a better initialization. [23] first introduce meta-learning into the UDA setting. Instead of splitting the training set into two stages, they use domain alignment loss in meta-training stage while classification loss in meta-testing stage such that two objectives can be optimized in a more coordinated way for better representation.

In this paper, thought our method is also based on meta-learning, our motivation, meta-optimization design, and goal are substantially different. To specify, our method is motivated by the representations distortion issue of adversarial learning, and propose to solve it by estimating the optimal adversarial weights via meta-learning. Our meta objective does not directly optimize the representations but aims at estimating a better adversarial weights.

2.4 Sample Weighting methods Our method is also related to sample weighting methods. In noisy label learning, Ren et al. [16] introduced learning to re-weight scheme for assigning the low weights for the noisy samples. [19] improves the stability of [16] by using an additional meta-learner for sample weighting.

In comparison, our method is substantially different from [16, 19] in terms of task, and meta-dataset construction and meta-learner design. Specifically, we re-weights the adversarial alignment to prevent features distortion while they re-weights the classification loss for noisy labels. Further, as there is no validation set in UDA setting, we construct a meta dataset with target-like distribution by mix-up augmentation while they directly use a held-out clean validation set. Our meta-learner takes both feature and prediction as input and generate adversarial weights while their meta-learner takes training loss as a scalar input.

3 Preliminary

3.1 Problem Definition In unsupervised domain adaptation (UDA) problem, we are given a source domain $\mathbb{D}_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)|_{i=1}^{N_s}\}$ of N_s labeled source examples and a target domain $\mathbb{D}_t = \{(\boldsymbol{x}_i^t)|_{i=1}^{N_t}\}$ of N_t unlabeled target examples. Note that source and target domain share the same label space. The joint distributions of source and target domain are not identically and independently distributed, specifically $P(\boldsymbol{x}^s, \boldsymbol{y}^s) \neq Q(\boldsymbol{x}^t, \boldsymbol{y}^t)$. The objective of UDA is to train a deep neural network $G(\cdot|\boldsymbol{\theta})$ on labeled source data $(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)$ drawn from $\mathbb{D}_s$ and unlabeled target data $\boldsymbol{x}_i^t$ drawn from $\mathbb{D}_t$ such that the model $G(\cdot|\boldsymbol{\theta})$ can generalize well on target domain. In details, network $G(\cdot|\boldsymbol{\theta}) = C \circ F(\cdot|\boldsymbol{\theta})$ is comprised of a feature extractor $F(\cdot|\boldsymbol{\theta})$ and a classifier $C(\cdot|\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ denotes network parameters.

In general, training a network $G(\cdot|\boldsymbol{\theta})$ on source domain only leads to sub-optimal performance as the domain gap issue is unsolved. The source supervised objective function is in the form of:

$$\mathcal{L}_s(F, C) = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{L}_{ce}\left(C(F(\boldsymbol{x}_i^s|\boldsymbol{\theta})), \boldsymbol{y}_i^s\right). \tag{3.1}$$

As the consequence, domain alignment methods are incorporated with Eq. 3.1 to mitigate the domain shift problem.

3.2 Adversarial Domain Adaptation Adversarial domain alignment methods, derived from Domain adversarial neural network (DANN) [5], have been arguably one of the most effective approaches to reduce the domain shift problem in UDA. The key idea of DANN is to learn transferable feature space by playing a two-player minimax game between a feature extractor $F(\cdot|\boldsymbol{\theta})$ and an newly introduced domain discriminator $D(\cdot|\boldsymbol{\theta})$. Specifically, the domain discriminator $D(\cdot|\boldsymbol{\theta})$ is optimized to distinguish the samples from source and target domain while the feature extractor $F(\cdot|\boldsymbol{\theta})$ is trained to fool the domain discriminator $D(\cdot|\boldsymbol{\theta})$. By these means, the feature extractor $F(\cdot|\boldsymbol{\theta})$ can learn a transferable representation across domains. Jointly training the $F(\cdot|\boldsymbol{\theta})$ with the category classifier $C(\cdot|\boldsymbol{\theta})$ on labeled source data, a discriminative representation $\mathbf{f} = F(\cdot|\boldsymbol{\theta})$ could be obtained across categories. Formally, the optimization process of DANN is formulated as follows:

$$\min_{F,C} \max_{D} \mathcal{L}_s(F, C) - \lambda \mathcal{L}_{adv}(F, D), \tag{3.2}$$

$$\begin{aligned} \mathcal{L}_{adv}(F, D) = &\frac{w_i^s}{N_s} \sum_{i=1}^{N_s} \log D(\boldsymbol{f_i^s}) + \\ &\frac{w_j^t}{N_t} \sum_{j=1}^{N_t} \log[1 - D(\boldsymbol{f_j^t})], \end{aligned} \tag{3.3}$$

where λ is a trade-off hyperparameter, $\boldsymbol{f_i^s}$ and $\boldsymbol{f_i^t}$ are the features, and w_i^s, w_j^t are the per-sample weights for source sample i and target sample j respectively. In DANN, w_i^s and w_j^t are set as one for all samples.

3.3 Conditional Adversarial Domain Adaptation As the joint distributions of two domains are different $P(\boldsymbol{x}^s, \boldsymbol{y}^s) \neq Q(\boldsymbol{x}^t, \boldsymbol{y}^t)$, only adapting the feature representation $\boldsymbol{f}$ without considering categorical structure is not enough. Specifically, there is a failure mode in DANN. When the features from two domains but different classes are aligned together, the discriminator is confused but categorical alignment is mismatched. To insert the discriminative class strucutres into the adversarial alignment, conditional adversarial domain adaptation (CDAN) is proposed to adapt both feature representation $\boldsymbol{f} = F(\cdot|\boldsymbol{\theta})$ and class prediction $\boldsymbol{g} = C(\cdot|\boldsymbol{\theta})$ with a joint representation $\boldsymbol{h} = (\boldsymbol{f}, \boldsymbol{g})$. In this sense, the Eq. 3.3 can be extended as follows:

$$\mathcal{L}_{cadv}(F, D) = \frac{w_i^s}{N_s} \sum_{i=1}^{N_s} \log D(\boldsymbol{f_i^s}, \boldsymbol{g_i^s}) + \frac{w_j^t}{N_t} \sum_{j=1}^{N_t} \log[1 - D(\boldsymbol{f_j^t}, \boldsymbol{g_j^t})]. \tag{3.4}$$

As a result, the adversarial learning can incorporate class structures into the domain alignment for better generalization.

4 Limitations of Adversarial Domain Adaptation

Existing domain adaptation methods are based on a theory proposed by [1]. Formally, let $\mathcal{H}$ denote the hypothesis space and $h \in \mathcal{H}$ denote the classifier, we can formulate the upper bound of target generalization error ϵ_t as:

$$\epsilon_t(h) \leq \epsilon_s(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda^*, \forall h \in \mathcal{H}, \tag{4.5}$$

where $\epsilon_s(h)$ is the source generalization error of h, $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$ is the $\mathcal{H}\Delta\mathcal{H}$-distance which measures the divergence between the source and target feature distributions, and $\lambda^* = \epsilon_s(h^*) + \epsilon_t(h^*)$ denotes the error of an ideal joint hypothesis h^* on source and target domains.

4.1 Error of ideal joint hypothesis In adversarial domain adaptation, $\mathcal{H}\Delta\mathcal{H}$-distance is minimized by learning a domain-invariant representation while $\epsilon_s(h)$ is minimized by training the classifier with labeled source samples. Previous adversarial methods generally assume that λ^* is small, but it is not the case according to the analysis from [11]. Specifically, to compute λ^*, they first fixed the feature representations learned by: ResNet-50 [6] and DANN [5] where ResNet-50 denotes the method training with Eq. 3.1 only. Then, they trained a new ideal classifier over the fixed representations with labeled source data and labeled target data, and obtain the errors λ^* on source and target domain. As a consequence, *λ^* of DANN is substantially worse than the λ^* of ResNet-50, indicating that the adversarial alignment would distort the discriminative structures of the original representations.*

4.2 Importance of class prediction in CDAN we found that the effectiveness of CDAN is largely correlated with the correctness of class prediction $\boldsymbol{g}$ according to our preliminary experiment. Specifically, when the class prediction $\boldsymbol{g}$ in Eq. 3.4 is 100 % correct with respect to ground truth, the classification accuracy on target data would achieve 100 % as well on Office-31 [17] dataset. Inversely, *if the class predictions $\boldsymbol{g}$ of target samples are incorrect, CDAN would wrongly align the source and target samples of different category together, leading to a negative transfer.*

5 Method

Motivated by the limitations of existing adversarial alignment methods above, we propose a meta-learning framework to learn a per-sample weight for adversarial alignment such that discriminative representations can be preserved, and name it as meta adversarial weight (MAW). In this section, 1) we first explain why estimating ideal per-sample weights helps adversarial domain adaptation from representations distortion, and then formulate the ideal weights estimation problem in a learning-to-learn manner via a meta-leaner. 2) Next, we introduce how we build up the meta-dataset with target-like distribution without access to target labels, and use it as meta knowledge to guide the search for the optimal per-sample weights. 3) Last, we illustrate bi-level meta-optimization procedure for updating a meta-learner to estimate the optimal adversarial weights.

5.1 Meta Adversarial Weight for Domain Alignment For existing adversarial domain adaptation, all source and target samples are equally involved in adversarial alignment. As a result, it might distort the original discriminative representations. To specify, training with labeled source domain only in Eq. 3.1 could already align a portion of target samples to their semantic counterpart in source domain. However, adding adversarial alignment for those well-aligned target samples might push them from the correct class into an incorrect one, leading to a negative transfer. Additionally, the value of weight λ in Eq. 3.2 also plays an important role in balancing the discriminativeness and transferability. If the weight is too large, adversarial alignment would take over the whole training and overlook the goal of learning discriminative representations.

We claim that the aforementioned issues can be solved if there are ideal per-sample weights existed for adversarial alignment, and they should have two properties as follows:

- After training with Eq. 3.1, only those poorly-aligned target samples are assigned the weights for adversarial alignment while those already well-aligned target samples would have very small or zero weights.
- For those poorly-aligned target samples, the value of adversarial weights can be adaptively adjusted such that those samples can be aligned to their correct classes.

The key challenge is how to estimate the per-sample adversarial weights such that they have such two properties. Instead of heuristically choosing the weights by conditional entropy [12] or disagreement of two classifiers [14], we devise a Meta-learning framework to adaptively learn the adversarial weights under the guidance of meta knowledge.

The main idea is to build up a meta-dataset with target distribution $\mathbb{D}_m = \{(\boldsymbol{x}_i^m, \boldsymbol{y}_i^m)|_{i=1}^{N_m}\}$ as meta knowledge, and use it to guide the learning of a meta-learner $M(\cdot|\boldsymbol{\phi})$ which estimates the optimal adversarial weights. Specifically, the meta-learner $M(\cdot|\boldsymbol{\phi})$ takes the feature and class prediction of a sample as input and output a adversarial weight for the sample in the form of $w_i = M(\boldsymbol{f_i}, \boldsymbol{g_i}|\boldsymbol{\phi})$. We can link our original model $G(\cdot|\boldsymbol{\theta})$ with the meta-learner $M(\cdot|\boldsymbol{\phi})$ in a bi-level optimization manner. *The intuition is that if the adversarial weights generated by the meta-learner $M(\cdot|\boldsymbol{\phi})$ are optimal, then a model $G(\cdot|\boldsymbol{\theta})$ trained with such weights for domain alignment should have low risk on the meta dataset $\mathbb{D}_m$.* Formally, the optimal adversarial weights estimation via the meta-learner $M(\cdot|\boldsymbol{\phi})$ can be formulated as the following bi-level optimization problem,

$$\min_{\boldsymbol{\phi}} \quad \mathbb{E}_{i\in\mathbb{D}_m} L_{ce}(x_i^m, y_i^m; \theta^*(\boldsymbol{\phi})) \text{ with} \tag{5.6}$$

$$\theta^*(\boldsymbol{\phi}) \leftarrow \arg\min_{\theta} \mathbb{E}_{i\in\mathbb{D}_{s,t}} L_s - \lambda L_{adv}(x_i^s, x_i^t; \theta, \boldsymbol{\phi}), \tag{5.7}$$

where Eq. 5.7 is extended from Eq. 3.2 except that the adversarial weights for samples in Eq. 3.3 are estimated by the meta-learner $w_i = M(\boldsymbol{f_i}, \boldsymbol{g_i}|\boldsymbol{\phi})$ instead of setting to one. We term this framework as Meta Adversarial Weight (MAW).
Meta-leaner Architecture: $M(\cdot|\boldsymbol{\phi})$ takes feature and class prediction of a sample as input by concatenation and feed it into a two-layer neural network with dimensions of $(C + xdim, hdim), (hdim, C)$ respectively where C is the number of classes, $xdim$ is the dimension of feature and $hdim$ is the dimension of hidden units. ReLU is used between layers as the activation function and output is with the Sigmoid function to guarantee the weight located between $[0, 1]$.

5.2 Meta-dataset Construction

5.2.1 Dilemma in UDA The typical pipeline of machine learning has a validation set which shares the same data distribution with testing set and can be served for hyperparameters selection. However, this pipeline is controversial to UDA setting where no validation set is available and testing set (target domain data) does not have labels to involve network training. Therefore, it is an open problem to obtain the clean target-like distribution data for model selection.

5.2.2 Creating clean target-like distribution As there are no labels for target data, we apply the widely-used pseudo-labeling strategy on target samples to obtain high-quality pseudo labels by setting a predefined threshold. The pseudo labels of target samples are selected only when the following criterion is meet:

$$\tilde{\boldsymbol{y}_i^t} = \mathbb{1}[\max(C(F(\boldsymbol{x}_i^t|\boldsymbol{\theta}))) > \tau], \tag{5.8}$$

where $\mathbb{1}$ is an indicator function and τ is a pre-defined threshold value (empirically setting to 0.9).

As pseudo labels are not 100% clean, we leverage mix-up augmentation strategy [24] to create virtual samples by mixing up the source domain samples $\mathbb{D}_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)|_{i=1}^{N_s}\}$ and selected pseudo labeled target domain samples $\tilde{\mathbb{D}}_t = \{(\boldsymbol{x}_i^t, \tilde{\boldsymbol{y}_i^t})|_{i=1}^{\tilde{N}_t}\}$. Specifically, we randomly sample a pair of source sample $(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)$ and target samples $(\boldsymbol{x}_i^t, \tilde{\boldsymbol{y}_i^t})$ of same class and mix them up to obtain the meta sample $(\boldsymbol{x}_i^m, \boldsymbol{y}_i^m)$ as follows:

$$\begin{aligned} \boldsymbol{x}_i^m &= \alpha \boldsymbol{x}_i^s + (1-\alpha)\boldsymbol{x}_i^t, \\ \boldsymbol{y}_i^m &= \boldsymbol{y}_i^s, \end{aligned} \tag{5.9}$$

where α follows a uniform distribution $U(0, 1)$.

Consequently, we can build up such a meta-dataset $\mathbb{D}_m = \{(\boldsymbol{x}_i^m, \boldsymbol{y}_i^m)|_{i=1}^{N_m}\}$ with target-like distribution by incorporating high-quality pseudo-labeled target samples while maintaining the cleanness by mixing with labeled source samples. Our method use this meta-dataset $\mathbb{D}_m$ as meta knowledge to guide the learning of the meta-learner for estimating the optimal adversarial weights.
Discussion. It is worth noting that there are some substantial differences between our method and either pseudo-labeling [31] or mix-up [24]. **Functionally**, pseudo-labeling or mix-up are generally used for model training and help to learn a smooth and discriminative decision boundary. In comparison, our meta dataset is not involved in model training but served as meta-knowledge to estimate the optimal adversarial weights via meta-leaner. **Conceptually**, we could consider our meta dataset as a validation set in machine learning to select the per-sample weights as a bunch of hyperparameters. In our case, per-sample weights can be estimated in a learning-to-learn manner during training. **Algorithmically**, unlike mix-up which mixes up two random samples, our meta-dataset mixes up two randomly sampled source examples and pseudo-labeled target examples of same class. The purpose is to ensure that the meta-sample is target-like while maintaining a certain degree of label cleanness. **Future work**, image-to-image translation method [7] can also help create clean target

distribution data by translating the source data with target style. As it will introduce extra computational cost, we leave it as future work.

5.3 Meta-Optimization for MAW To illustrate the bi-level meta-optimization in Eq. 5.6 and Eq. 5.7, we introduce three steps to perform one iteration of optimization for model $G(\cdot|\boldsymbol{\theta})$ and meta-learner $M(\cdot|\boldsymbol{\phi})$.

Pseudo Model Update: At each iteration t, we uniformly sample a batch B_s from source data $\mathbb{D}_s$, a batch B_t from target data $\mathbb{D}_t$ and feed them into the model $G(\cdot|\boldsymbol{\theta})$ to update the model parameters θ.

(5.10)
$$\theta^*(\boldsymbol{\phi^t}) \leftarrow \theta^t - \eta \frac{\partial \mathbb{E}_{i\in\mathbb{D}_{s,t}} L_s(x_i^s, y_i^s; \theta) - \lambda L_{adv}(x_i^s, x_i^t; \theta^t, \boldsymbol{\phi^t})}{\partial \theta^t}$$

This step is considered as the pseudo update for model as θ^* can be rolled back to previous iteration t if a better meta-learner $\boldsymbol{\phi}$ is obtained.

Meta-Update on Meta-leaner: We sample a batch B_m from meta dataset $\mathbb{D}_m$ via Eq. 5.9. Then we feed the meta batch B_m into the model θ^* that obtained in the previous step to update the parameters of meta-learner $M(\cdot|\boldsymbol{\phi})$:

(5.11)
$$\boldsymbol{\phi^{t+1}} \leftarrow \boldsymbol{\phi^t} - \beta \frac{\partial \mathbb{E}_{i\in\mathbb{D}_m} L_{ce}(x_i^m, y_i^m; \theta^*(\boldsymbol{\phi^t}))}{\partial \phi^t}$$

As a result, the updated meta-learner $\boldsymbol{\phi^{t+1}}$ should be better than the previous updated one $\boldsymbol{\phi^t}$, in the sense that it results in smaller classification errors on the meta dataset by estimating more optimal adversarial weights.

Real Model Update: We apply the obtained new meta-learner $\boldsymbol{\phi^{t+1}}$ to conduct the real model update given the same training batches from Eqn. 5.10:

(5.12)
$$\theta^{t+1} \leftarrow \theta^t - \eta \frac{\partial \mathbb{E}_{i\in\mathbb{D}_{s,t}} L_s(x_i^s, y_i^s; \theta) - \lambda L_{adv}(x_i^s, x_i^t; \theta^t, \boldsymbol{\phi^{t+1}})}{\partial \theta^t}$$

As summarize in Alg. 1, we pursue the ideal adversarial weights generated by the optimal meta-leaner $\boldsymbol{\phi}$ for Eqn 5.12 in terms of minimizing the loss on meta dataset in Eqn 5.11.

Algorithm 1 Optimization for Meta Adversarial Weight

Input: source dataset $\mathbb{D}_s$, unlabeled target dataset $\mathbb{D}_t$
Build up meta-dataset at Eq. 5.9
for $t = 1, 2, \cdots, t_{all}$ **do**
  1: Sample two batch B_s, B_t from $\mathbb{D}_s$, $\mathbb{D}_t$ respectively
  2: Pseudo Model Update $\theta^*(\boldsymbol{\phi^t})$ at Eq. 5.10
  3: Sample a meta-batch B_m from $\mathbb{D}_m$
  4: Meta-Update on Meta-learner $\boldsymbol{\phi^{t+1}}$ at Eq. 5.11
  5: Real Model Update θ^{t+1} at Eqn. 5.12
end

6 Experiment

6.1 Datasets We conduct experiments on three domain adaptation classification benchmarks: Office-31 [17], Office-Home [22] and ImageCLEF-DA. **Office-31** is a commonly used dataset for unsupervised domain adaptation. It includes 4652 images of 31 classes from three domains: Amazon (A), Webcam (W) and DSLR (D). **ImageCLEF-DA** consists of 12 common classes shared by three public datasets (domains): Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). **Office-Home** is a more challenging benchmark than Office-31. It consists of images of everyday objects organized into four domains: artistic images (Ar), clip art (Cl), product images (Pr), and real-world images (Rw). It contains 15,500 images of 65 classes.

6.2 Implementation details We follow the standard protocol of UDA([5, 26]) to use all labeled source samples and all unlabeled target samples as training data. The reported testing results are the average accuracy over three random repeats with center-crop images. We adopt ResNet-50 [6] on Office-31, ImageCLEF-DA and Office-Home dataset, fine-tuned from the ImageNet pre-trained model. We use Pytorch as implementation framework. We adopt Stochastic Gradient Descent (SGD) optimizer with learning rate of 1×10^{-3}, weight decay 5×10^{-4}, momentum 0.9 and batch size 32. For optimization, we first pre-train the model based on source data only in Eqn 3.1. Then, we train our MAW framework based on the three steps optmization in Eq 5.10, Eq 5.11 and Eq 5.12. The threshold value τ for high-quality pseudo label is fixed as $\tau = 0.90$ and the trade-off hyperparameter in Eq.3.2 is set to $\lambda = 0.2$ for all datasets. Note that the results of existing methods in Table 1, 2, 3 refer to their respective papers.

6.3 Comparison with State-of-the-Arts In this section, we select several state-of-the-art methods in UDA such as DMRL [24], MDD [29], BNM [3], SymNets [30] and AADA [27]. Further, we also compare our methods with other regularizers that are designed for improving adversarial alignment to a state-of-the-art performance, such as BSP [11], MetaAlign [23] and ILA-DA [18].

6.3.1 Results on Office-31 Results based on ResNet-50 are shown in Table 1. 1) Our method can be served as a plugged in module to boost DANN [5] and CDAN [12] by a significant margin. 2) Compared to other regularizers for adversarial DA such as BSP[11], MetaAlign [23] and ILA-DA [18], our method shows the consistent improvements over them on different DA methods. 3) Comparing to the general state-of-the-art methods such as SAFN [26], DMRL [24], MDD [29], our method outperforms it substantially by 1.1%.

https://www.imageclef.org/2014/adaptation

Table 1: Experiment results on Office-31 classification using ResNet-50. Best(**bold**), second best (underline).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50 [6]	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
ADDA [20]	86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN [13]	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
MRENT [32]	88.0±0.4	98.6±0.1	100.0±0.0	87.4±0.8	72.7±0.2	71.0±0.4	86.4
SAFN [26]	90.1±0.8	98.6±0.2	99.8±0.0	90.7±0.5	73.0±0.2	70.2±0.3	87.1
DMRL [24]	90.8±0.3	99.0±0.2	100.0±0.0	93.4±0.5	73.0±0.3	71.2±0.3	87.9
MDD [29]	94.5±0.3	98.4±0.1	100.0±0.0	93.5±0.2	74.6±0.3	72.2±0.1	88.9
DANN [5]	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
+ BSP [11]	93.0±0.2	98.0±0.2	100.0±0.0	90.0±0.4	71.9±0.3	73.0±0.3	87.7
+ MetaAlign [23]	93.9±0.4	98.7±0.2	100.0±0.	91.6±0.3	73.7±0.2	74.1±0.2	88.7
+MAW (ours)	92.8 ±0.1	98.6±0.0	100.0 ±0.0	92.6 ±0.2	75.9 ±0.1	**76.3±0.2**	89.4
CDAN[12]	94.1±0.1	98.6±0.1	100.0±0.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
+ BSP [11]	93.3±0.2	98.2±0.2	100.0±0.0	93.0±0.2	73.6±0.3	72.6±0.3	88.5
+ILA-DA [18]	**95.7±0.0**	**99.2±0.0**	100.0±0.0	93.3 ±0.0	72.1±0.0	75.4±0.0	89.3
+MAW (ours)	94.2±0.2	98.7±0.1	**100.0±0.0**	**94.4±0.3**	**76.5±0.2**	75.8±0.4	**90.0**

Table 2: Experiment results on Office-Home for unsupervised domain adaptation (ResNet-50). Best(**bold**), second best (underline).

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
ResNet-50 [6]	34.9	50	58	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
AADA [27]	54.0	71.3	77.5	60.8	70.8	71.2	59.1	51.8	76.9	71.0	57.4	81.8	67.0
SymNets [30]	47.7	72.9	78.5	64.2	71.3	74.2	63.6	47.6	79.4	73.8	50.8	82.6	67.2
SAFN [26]	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
BNM [3]	52.3	**73.9**	**80.0**	63.3	**72.9**	**74.9**	61.7	49.5	79.7	70.5	53.6	82.2	67.9
MDD [29]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
DANN [5]	45.6	59.3	70.1	47	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
+MetaAlign [23]	48.6	69.5	76.0	58.1	65.7	68.3	54.9	44.4	75.3	68.5	50.8	80.1	63.3
+BSP [11]	51.4	68.3	75.9	56.0	67.8	68.8	57.0	49.6	75.8	70.4	57.1	80.6	64.9
+MAW (ours)	52.6	72.4	78.4	61.1	72.2	72.4	59.4	**52.2**	79.0	**73.5**	58.7	**82.8**	67.8
CDAN [12]	49.0	69.3	74.5	54.4	66	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
+BSP [11]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
+MetaAlign [23]	55.2	70.5	77.6	61.5	70.0	70.0	58.7	**55.7**	78.5	73.3	**61.0**	81.7	67.8
+ MAW (ours)	**55.9**	72.8	78.8	62.2	72.5	73.5	60.4	53.2	**80.2**	**74.2**	59.4	**83.6**	**68.9**

6.3.2 Results on Office-Home Result based on ResNet-50 are reported in Table 2. Similarly conclusion can be drawn that MAW shows consistent improvements on different adversarial methods such as DANN and CDAN, and outperforms other state-of-the-art regularizers such as BSP[11], MetaAlign[23] and ILA-DA[18].

6.3.3 Results on ImageCLEF-DA Results based on ResNet-50 are shown in Table 3. Comparing to state-of-the-art adversarial methods DMRL [24] and AADA[27], our MAW improves DANN to 88.8 % and improves CDAN to 89.5 % accuracy, and outperforming other adversarial methods by a substantial margin.

6.4 Analysis

6.4.1 Feature visualization We visualize the target feature embeddings of (a) source model, (b) DANN and (c) **DANN + MAW** on Office-31 W→ A via t-SNE ([15]) in Fig.2 (a-c). We can qualitatively observe that **DANN + MAW** could learn more discriminative target features than DANN.

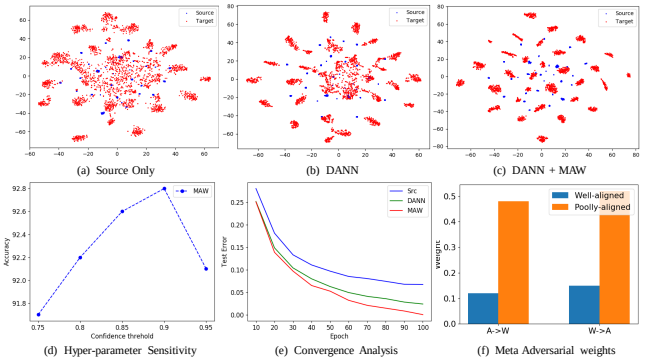

Figure 2: Analysis: (a-c) The t-SNE visualization of target feature (red) on Office-31 $W \rightarrow A$. (d) Hyperparameter sensitivity on Confidence threshold on Office-31 A → W. (e) Convergence analysis w.r.t Test error on Office-31 A → W. (f) Visualization on Meta adversarial weights on Office-31.

6.4.2 Hyper-parameter sensitivity on Confidence Threshold We conduct the hyper-parameter sensitivity analysis on the the threshold value τ for selecting pseudo label in Eqn. 5.8 on Office31 A → W. As Fig.2 (d) shows, our method is robust to confidence threshold while it is

Table 3: Experiment results on ImageCLEF-DA classification using ResNet-50

Method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
ResNet-50[6]	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DANN[5]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
CDAN[12]	76.7	90.6	**97.0**	90.5	74.5	93.5	87.1
SAFN[26]	78.0	91.7	96.2	91.1	77.0	94.7	88.1
DMRL[24]	77.3	90.7	97.4	91.8	76.0	94.8	88.0
AADA[27]	79.2	92.5	96.2	91.4	76.1	94.7	88.4
DANN + MAW	79.5	91.5	96.3	92.1	77.9	95.6	88.8
CDAN + MAW	**79.7**	**92.6**	96.9	**92.8**	**78.6**	**96.1**	**89.5**

recommended to choose τ around 0.9. If τ is too high, the meta dataset will only have limited amount of samples such that it fails to capture the target distribution. If τ is too small, meta set would contain too many noisy target samples.

6.4.3 Convergence Comparison In order to validate whether our method can guide the existing adversarial method towards class-wise alignment, we quantitatively visualize the change of entropy and test error with respect to training epochs for three candidates method. From Fig. 2(e), we observe similar trend for entropy and test error in the sense that our MAW could converge much faster than DANN thanks to the guidance from unbiased task risk estimator.

6.4.4 Visualization on Meta Adversarial Weight To verify whether the learned adversarial weights match our hypothesis, we first evaluate the source only model on target domain on Office-31 dataset. We denote the correctly classified target samples as "Well-aligned" samples while wrongly classified target samples as "Poorly-aligned" samples. After training with our MAW framework, we use meta-learner to generate the weights for all target samples, and take an average of the weights for the "Well-aligned" and "Poorly-aligned" samples respectively on A $\rightarrow$ W and W $\rightarrow$ A task. We found that our meta-learner would assign small weights to the "Well-aligned" samples while large weights to the "Poorly-aligned" samples. It demonstrates that our learned meta adversarial weights follow the first property of 'ideal' adversarial weights.

6.4.5 Ablation Study To differentiate our MAW with pseudo-labeling (PL) and mix-up methods, we conduct ablation studies by jointly training DANN with either PL or mix-up in a naive way. From table 4, our DANN + MAW achieves the best performance at 89.4 % and outperforms either PL or mix-up by a significant margin. We claim that our MAW could better achieve the categorical adversarial alignment with the guidance from our meta-learned weights compared to those naive jointly training.

7 Conclusion and Future work

In this work, we devise a meta-learning framework to adaptively learn optimal adversarial weights to help the feature distortion issue from adversarial alignment. Specifically, we construct a meta-dataset with target-like distribution as meta knowledge, and use it to guide the learning of the optimal adversarial weights via a meta-learner. By this means, our framework can adaptively adjust the weights of all training samples in adversarial training based on the feedback from meta dataset and thus achieve the categorical-wise domain alignment.

Table 4: Ablation studies using Office-31 based on ResNet-50. Please refer to Section 6.4.5 on what each component represents.

Method	Avg
DANN	82.2
+PL	84.7
+Mixup	87.4
+ MAW	**89.4**

For future work, 1) we could use image-to-image translation method to construct the meta dataset with target distribution. 2) Besides estimating the adversarial weights, we could also leverage the meta knowledge to correct the class predictions of target samples such that it could benefit both conditional adversarial DA and pseudo labeling methods.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

[2] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1081–1090, 2019.

[3] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. pages 3941–3950, 2020.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

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

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[7] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[8] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–100, 2019.

[9] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 382–403. Springer, 2020.

[10] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[11] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*, pages 4013–4022, 2019.

[12] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

[13] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, pages 2208–2217. JMLR. org, 2017.

[14] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.

[15] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[16] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

[17] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, 2010.

[18] Astuti Sharma, Tarun Kalluri, and Manmohan Chandraker. Instance level affinity-based transfer for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5361–5371, 2021.

[19] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379*, 2019.

[20] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[21] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[22] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

[23] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16643–16653, 2021.

[24] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. Dual mixup regularized learning for adversarial domain adaptation. In *European Conference on Computer Vision*, pages 540–555. Springer, 2020.

[25] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 5423–5432, 2018.

[26] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *IEEE International Conference on Computer Vision*, pages 1426–1435, 2019.

[27] Jianfei Yang, Han Zou, Yuxun Zhou, Zhaoyang Zeng, and Lihua Xie. Mind the discriminability: Asymmetric adversarial domain adaptation. In *European Conference on Computer Vision*, pages 589–606. Springer, 2020.

[28] Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 781–797. Springer, 2020.

[29] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019.

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

[31] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*, 2018.

[32] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *IEEE International Conference on Computer Vision*, pages 5982–5991, 2019.