Exploring Domain-Invariant Parameters for Source Free Domain Adaptation

Fan Wang*
Shandong University
fanwangsail@gmail.com

Zhongyi Han*
Shandong University
hanzhongyicn@gmail.com

Yongshun Gong
Shandong University
ysgong@sdu.edu.cn

Yilong Yin†
Shandong University
ylyin@sdu.edu.cn

Abstract

Source-free domain adaptation (SFDA) newly emerges to transfer the relevant knowledge of a well-trained source model to an unlabeled target domain, which is critical in various privacy-preserving scenarios. Most existing methods focus on learning the domain-invariant representations depending solely on the target data, leading to the obtained representations are target-specific. In this way, they cannot fully address the distribution shift problem across domains. In contrast, we provide a fascinating insight: rather than attempting to learn domain-invariant representations, it is better to explore the domain-invariant parameters of the source model. The motivation behind this insight is clear: the domain-invariant representations are dominated by only partial parameters of an available deep source model. We devise the Domain-Invariant Parameter Exploring (DIPE) approach to capture such domain-invariant parameters in the source model to generate domain-invariant representations. A distinguishing method is developed correspondingly for two types of parameters, i.e., domain-invariant and domain-specific parameters, as well as an effective update strategy based on the clustering correction technique and a target hypothesis is proposed. Extensive experiments verify that DIPE successfully exceeds the current state-of-the-art models on many domain adaptation datasets.

1. Introduction

Unsupervised domain adaptation (UDA) has been gaining momentum in the past decade, effectively addressing the distribution shift problem across domains. Thanks to the free access to labeled source data, previous UDA studies have achieved remarkable achievements [10, 25, 49]. However, the source data is unavailable in various privacy-preserving scenarios: data privacy protection laws and data silos in clinical practice [38]. Moreover, the fully test-time adaptation [44] assumes that the model could be sensitive to changing conditions, e.g., domain shift, during testing without the training data. In such practical limitations, source-free domain adaptation (SFDA) relaxes the source data requirement and leverages the source model's knowledge for domain adaptation.

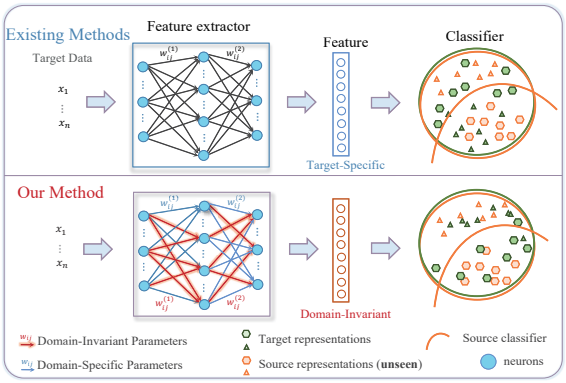


Figure 1. The comparison between existing methods and our method. Existing methods (top) optimize the model parameters without distinction, obtaining the target-specific representations which would not be well adapted to the source classifier, i.e., some samples are classified wrongly. Our method (bottom) would obtain domain-invariant representations by exploring domain-invariant parameters, guaranteeing the generalization of the source model.

The fundamental challenge of SFDA is that the domain-invariant presentations are challenging to be explored directly depending solely on the target data, as previous works have attempted to do. Both SHOT [23] and PPDA [17] utilize various techniques, e.g., entropy functions and self-

*Contribute to this work equally
†Corresponding author

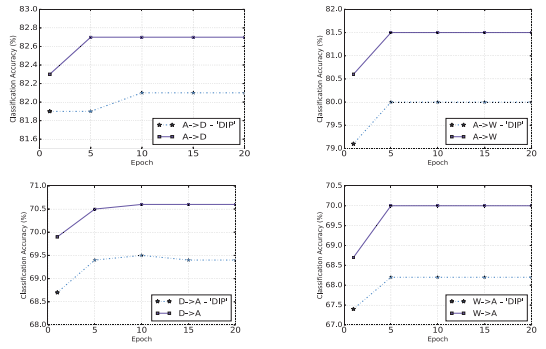


Figure 2. Accuracy (%) on Office-31 with or without DIP ('-DIP'). The model parameterized by the source model is finetuned on the pseudo labels predicted by itself.

supervised loss, to finetune the source model. To some extent, SHOT has made some progress on learning domain-invariant representations. But it ignores the domain shift, and the fixed source classifier module could not recognize the learned representations well (see Fig. 1).

In this paper, we give a novel insight: *In SFDA, exploring domain-invariant parameters stored in the source model is more feasible than exploring domain-invariant representations directly.* The insight is inspired by the lottery ticket hypothesis [8] which has demonstrated the significance of partial parameters in deep networks for generalization. Similarly, we discover that only partial parameters in the source model, termed domain-invariant parameters (DIP), are significant for the domain-invariant representations. On the contrary, the other parameters, termed domain-specific parameters, would tend to fit domain-specific information and hurt the generalization. As shown in Fig. 2, the model with exploring domain-invariant parameters produces better results on four tasks of Office-31 in our proposed method, *Domain-Invariant Parameter Exploring* (DIPE).

DIPE aims to explore domain-invariant parameters stored in the source model to generate domain-invariant representations and relieve domain shift. Three essential parts support DIPE to capture domain-invariant parameters precisely. First, to judge whether a parameter is domain-invariant or domain-specific, we propose a domain-balanced identifying criterion that simultaneously observes the active parameters in source and target models. Second, based on an intuition that the learned representations become closer to the domain-invariant ones as the training process proceeds, we suggest that the proportion of domain-invariant parameters should increase with the number of iterations. Third, we design an effective update strategy for these two types of parameters by the self-supervised loss based on clustering correction from the source and target hypotheses. Specifically, for domain-invariant parameters, we perform an active update. For domain-specific ones, we perform a passive update which will penalize their values to be near zero and gradually make them lose activity.

We summarize our main contributions as follows:

- To the best of our knowledge, we, for the first time, explore domain-invariant parameters stored in the given source model, opening up a new perspective in SFDA.
- We propose a novel DIPE framework for exploring domain-invariant parameters, and introduce a domain-balanced identifying criterion to determine domain-invariant and domain-specific parameters.
- A simple and general technique, clustering correction, is proposed to promote the learning process.

2. Related Work

2.1. Unsupervised Domain Adaptation

UDA methods have achieved great success in recent years. These methods can be grouped into four categories: importance estimation, moment matching, pseudo labeling, and adversarial learning. (1) The core idea of importance estimation is to measure the distance of the source samples from the overlapping distributions between source and target domains, consequently optimizing the importance weighted objective function [39]. (2) The moment matching tries to minimize the discrepancy in high dimensional statistics across domains [15, 26]. (3) The pseudo labeling utilizes pseudo labels of the target samples to realize standard supervised learning [34, 37, 48]. (4) The main idea behind adversarial training is to introduce a domain discriminator to distinguish the samples between two domains for learning the domain-invariant representations [9,25,36]. However, the success of all these methods depends on the accessed source data, which is unsafe and often unrealistic because the source data may be private and decentralized.

2.2. Source-Free Domain Adaptation

As data privacy protection has been drawing attention, SFDA is considered in the literature gradually. A few SFDA works can be divided into model-finetuning-based works and data-generation-based works. (1) Model-finetuning-based works attempt to explore domain-invariant representations by finetuning the source model. [23] attempts to learn representations that would align the source distribution by information maximization and self-supervision loss. [17] finetunes the source model by reliable pseudo labels of target samples based on entropy functions. However, they rely only on the target data and do not sufficiently consider the given model's source information, resulting in limited performance. (2) The core idea of data-generation-based methods is to generate source or target data to achieve standard domain adaptation. [22] proposes the 3C-GAN framework to generate target data with annotations. Although 3C-GAN achieves a certain performance gain, it needs enor-

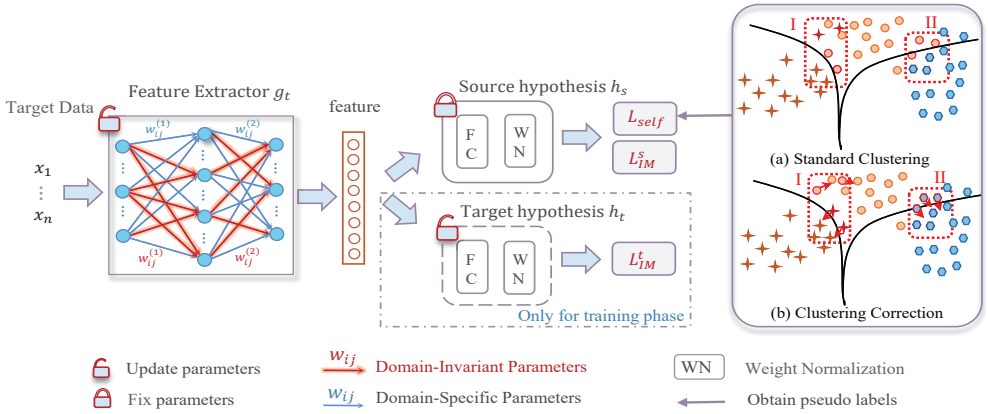


Figure 3. The framework of domain-invariant parameter exploring (DIPE). We visualize the parameters of the feature extractor, where the blue links indicate domain-specific parameters, which we need to deactivate gradually, and the red ones indicate domain-invariant parameters, which we need to emphasize. Here the L_{self} is calculated on the pseudo labels from clustering correction.

mous computing resources and cannot be applied to complex target tasks. Recently, more challenging settings concerning online SFDA [44] and federated SFDA [31] are also discussed.

2.3. Lottery Ticket Hypothesis

The lottery ticket hypothesis [8] proves that overparameterized DNNs contain winning tickets (parameters) that are significant for generalization. It indicates that partial parameters stored in the network contribute little to generalization and appear redundant. Although the lottery ticket hypothesis motivates our idea, this study is fundamentally different from it. Instead of searching a sparse sub-network with competitive generalization performance compared with the original network, we hope to explore the domain-invariant parameters and reduce the side effect of domain-specific information. These domain-invariant parameters would further generate domain-invariant representations, which is challenging to obtain in SFDA.

3. Domain-Invariant Parameter Exploring

In this section, we first show necessary notations of SFDA. Then, we propose the domain-balanced identifying criterion, introduce the domain-balanced identifying criterion to determine the proportion of domain-invariant parameters, and present the effective update strategy for two types of parameters by a newly-designed clustering correction in a self-supervised way. The framework of domain-invariant parameter exploring (DIPE) is illustrated in Fig. 3.

3.1. Learning Setup

The main difference between UDA and SFDA is that SFDA cannot utilize source data strictly during the training process in privacy-preserving scenarios, i.e., the source data D_s cannot be obtained directly. Instead, a well-trained source model $f_s : \mathcal{X}_s \to \mathcal{Y}_s$ trained on D_s and n_t unlabeled data $\{x_t^i\}_{i=1}^{n_t}$ from target domain D_t are given, where $x_t^i \in \mathcal{X}_t$. In the multi-classification task, $\mathcal{Y} \in \{1, ..., K\}$, and K represents the number of the classes. Here, $D_s \sim p$, $D_t \sim q$, and the distributions p and q are similar but different. The goal of SFDA is to predict the labels $\{y_t^i\}_{i=1}^{n_t}$ in the target domain without the source data.

3.2. Source Model Generation

In realistic scenarios, the source model provided by a third party might not be acquired in the laboratory, so we simulate the third party to train the source model $f_s : \mathcal{X}_s \to \mathcal{Y}_s$ with the labeled source data. In addition, to encourage the source data to lie in tight clusters, we utilize the label smoothing technique following [23]. Accordingly, the objective function is

$$L_s^{\text{ls}}(f_s; \mathcal{X}_s, \mathcal{Y}_s) = -\mathbb{E}_{(x_s,y_s)\in(\mathcal{X}_s,\mathcal{Y}_s)} \sum q_k^{\text{ls}} \log \delta_k(f_s(x_s)), \tag{1}$$

where $\delta_k(\boldsymbol{a}) = \frac{\exp(\boldsymbol{a}_k)}{\sum_i \exp(\boldsymbol{a}_i)}$ denotes the k-th element in the softmax output of a K-dimensional vector $\boldsymbol{a}$, and $q_k^{\text{ls}} = (1-\epsilon)q_k + \frac{\epsilon}{K}$. q_k is the one-of-K encoding of y_s, and ϵ is the smoothing parameter.

As shown in Fig. 3, the target model parameterized by the above source model consists of three modules: the feature extractor g_t: $\mathcal{X}_t \to \mathbb{R}^d$, the source hypothesis (fixed classifier) h_s, and the target hypothesis (trainable classifier) h_t, i.e., $f_t(x) = h_t(g_t(x))$. Here d is the dimension of the input feature. We propose the domain-invariant parameter exploring (DIPE) to explore domain-invariant parameters in the feature extractor (g_t) and generate domain-invariant representations that could be well adapted to the source hypothesis (h_s). DIPE can mitigate the negative effects of target-oriented information excessively, which is overlooked by

the previous methods. Further, the trainable target hypothesis (h_t) is introduced to cooperate with the source hypothesis, avoiding source-oriented information. Next we will describe how to explore the domain-invariant parameters.

3.3. Domain-Balanced Identifying Criterion

The principle of identifying domain-invariant parameters is to find those critical parameters that play a decisive role in exploring domain-invariant representations. In the forward propagation, there are partial parameters that are relatively large and have the same positive and negative role in the same position in the source and target models. These parameters are more active and play a co-directional role in representation extraction. Thus we term them as domain-invariant parameters (DIP). In contrast, the inconsistency between the positive and negative parameters of the source model and the target model under training at the same location indicates that they act in opposite directions, we thus term these parameters as domain-specific parameters (DSP). Based on the analysis above, we design a domain-balanced identifying criterion as follows. At the t-th iteration, denote by $w_i^s(t)$ stored in the source model and $w_i^t(t)$ stored in the target model at the same position. The judgment criterion is denoted by g_i, *i.e.*,

$$g_i = |w_i^s(t) + w_i^t(t)|, \quad i \in [m], \tag{2}$$

where m is the parameter number of the feature extractor and target hypothesis. If the value of g_i is large, w_i^t is viewed as a domain-invariant parameter. Otherwise, w_i^t is regarded as a domain-specific one that tends to fit domain-specific information. By the way, the target hypothesis with exploring domain-invariant parameters aims to avoid updating the target-oriented gradients, further promoting the learning of domain-invariant parameters.

3.4. Identify the Proportion of DIP

Intuitively, with network training, the representations gradually tend to be domain-invariant. So we determine the proportion of domain-invariant parameters with the increase of training iterations. Specifically, we denote by τ the dynamic proportion of domain-invariant parameters, which is defined by

$$\tau = 1 - d\frac{2\exp(\frac{-10c}{\mathrm{T_m}})}{1.0 + \exp(\frac{-10c}{\mathrm{T_m}})}, \tag{3}$$

where c represents current iterations, and T_m represents the maximal iterations, $\tau \in [1-d, 1]$.

3.5. Updating Parameters with Different Rules

Updating the parameters for different types is a fine-grained version of the conventional parameter fine-tuning-based strategies. The domain-invariant parameters can be further found and updated to emphasis, weakening the potential influence of domain-specific ones, thus producing domain-invariant representations. Therefore, for domain-invariant parameters, we utilize standard stochastic gradient descent (SGD) [27] algorithm to update, termed as **active update** rule,

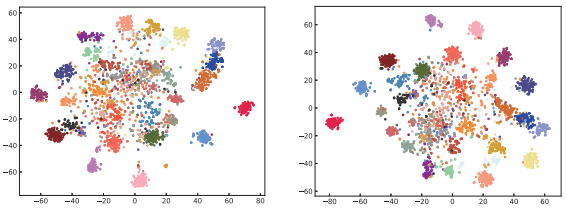
$$W_{IP}(t+1) \leftarrow W_{IP}(t) - \eta(\frac{\partial L(W_{IP}(t))}{\partial W_{IP}(t)} + \lambda W_{IP}(t)), \tag{4}$$

where $\lambda \in \mathbb{R}_+$ is a regularization parameter which equals to the weight decay coefficient with SGD, $\eta > 0$ is the learning rate, and t indicates the t-th iteration. W_{IP} represents the set of domain-invariant parameters, and W_{SP} represents the set of domain-specific ones.

For domain-specific parameters, we update them only utilizing a regularization item, i.e., weight decay, termed as **passive update** rule [46]. Moreover, we use the standard sgn function to replace the normal weight decay by sgn (W_{SP}), which would make the value of W_{SP} converge to zero or near zero faster as the network training. The passive update rule is defined by

$$W_{SP}(t+1) \leftarrow W_{SP}(t) - \eta\lambda \mathrm{sgn}(W_{SP}(t)), \tag{5}$$

In practice, the label of target data is unavailable, and the poor quality of the pseudo-labels predicted by the model itself would learn the wrong information. Therefore, we propose the clustering correction to correct the pseudo labels.



(a) D→A (b) W→A

Figure 4. The t-SNE visualization of target features.

Clustering Correction Fig. 4 shows the t-SNE visualization of target features on the source model. It indicates that the same class of the target data can still form a cluster in the embedding space under domain shift. Moreover, many domain adaptation works [5, 21, 30, 40] have verified the effectiveness of clustering. So the clustering helps explore the inherent structure of the target data. Further, we propose the clustering correction to obtain more accurate pseudo labels and support the optimization of supervised loss.

Clustering correction aims to correct the error-prone pseudo labels by exploring the relationship between representations. Specifically, clustering correction first utilizes the deep k-means clustering [23] to predict the pseudo la-

bels of target samples, in which the model output and clustering results are utilized as supervisory information. Then, it corrects the samples with ambiguous pseudo labels that are located in the decision boundary.

First, clustering correction obtains the pseudo-labels by weighted k-means clustering [23]:

$$\hat{y}_t = \arg\min_k D_f(g_t(x_t), c_k)\,, \tag{6}$$

where $D_f(\cdot,\cdot)$ denotes the cosine distance of two variables, g_t denotes the learned representations, c_k denotes the class centroids which can robustly and more reliably characterize the distribution of different classes within the target domain.

Then, clustering correction searches the multiple neighbors (closest to the cosine distance) for each sample and corrects ambiguous pseudo labels. In detail, if the majority of the neighbors of this sample have the same pseudo label as itself, the sample's label is maintained. Else, the fuzzy sample's label is corrected to the label of the majority of its neighbors, such as the samples at the decision boundary in the upper right corner of Fig. 3. The red, orange, and blue colors represent three classes, respectively. The ambiguous labels of uncertain samples (red border) are corrected by

$$\hat{y}_t = \text{maxcommon}(\hat{y}_{t1}, \hat{y}_{t2}, ..., \hat{y}_{tn})\,, \tag{7}$$

where n represents the number of neighbors.

Based on the accurate pseudo labels of target samples, we optimize a standard supervised loss by

$$L_{self} = \mathbb{E}_{(x_t,\hat{y}_t)\in(\mathcal{X}_t,\hat{\mathcal{Y}}_t)} \sum_{k=1}^{K} \mathbb{1}_{[k=\hat{y}_t]} \log\delta_k(h_s(g_t(x_t)))\,. \tag{8}$$

Besides, for promoting the target outputs reliable and globally diverse, we apply the information maximization (IM) [11] loss which is composed of entropy minimization (L_{ent}) and Kullback–Leibler divergence (L_{div}):

$$\begin{aligned} L_{ent}(f_{ts};\mathcal{X}_t) &= -\mathbb{E}_{x_t\in\mathcal{X}_t} \sum_{k=1}^{K} \delta_k(f_{ts}(x_t)) \log(\delta_k(f_{ts}(x_t)))\,, \\ L_{div}(f_{ts};\mathcal{X}_t) &= \sum_{k=1}^{K} \hat{p}_k \log(\hat{p}_k) = D_{KL}(\hat{p}, \frac{1}{K}\mathbb{1}(K)) - \log(K)\,, \\ L^s_{IM} &= L_{ent}(f_{ts};\mathcal{X}_t) + L_{div}(f_{ts};\mathcal{X}_t)\,, \end{aligned} \tag{9}$$

where $f_{ts}(x) = h_s(g_t(x))$ is the K-dimensional output of each target sample, $\mathbb{1}(K)$ is a K-dimensional vector with all ones, and $\hat{p} = \sum_{x_t\in\mathcal{X}_t}[\delta(f_{ts}(x_t)]$ is the mean output embedding of the whole target data. In addition, to prevent the learned feature from being source-oriented, we calculate the L^t_{IM} from target hypothesis, where $f_t(x) = h_t(g_t(x))$.

In summary, the overall optimization goal for updating two types of parameters is stated as

$$L(W,S) = L^s_{IM} + \gamma L^t_{IM} + \beta L_{self}\,. \tag{10}$$

where $\beta > 0$ and $\gamma > 0$ are balancing hyper-parameters. The overall procedure of the proposed method DIPE is summarized in Algorithm 1.

Algorithm 1 DIPE Algorithm.

Input: source model $f_s = g_s \circ h_s$, target data $\{x^i_t\}_{i=1}^{n_t}$;
Parameters: maximum number of epochs E, trade-off parameters d, β, γ;
Initialization: freeze source hypothesis h_s and copy the parameters to g_t and h_t;

```
Let epoch = 1, iter_num = 0;
while epoch ≤ E do
    obtain pseudo labels based clustering correction with Eq. (6-7);
    while iter_num < n_b do
        Fetch mini-batch D̂_t from D_t;
        Calculate L(W, S) loss with Eq (8), (9),(10);
        Divide W into W_IP and W_SP with Eq (2), (3);
        Update W_IP with Eq (4);
        Update W_SP with Eq (5);
    end while
end while
```

4. Experiment

4.1. Experimental Setup

Digits is a standard UDA dataset that supports digit recognition with diverse domains. Following the protocol in [14], we utilize three subsets: SVHN (**S**) [29], MNIST (**M**) [18], and USPS (**U**) [16]. Each domain has 10 classes.

Office-31 [33] is a small-scale UDA dataset consisting of three diverse domains: Amazon (**A**), Dslr (**D**), and Webcom (**W**). Each domain has 31 classes.

Office-Home [43] is a more challenging UDA dataset consisting of four domains: Artistic images (**Ar**), Clip Art images(**Cl**), Product images (**Pr**), and Real-world images (**Re**). Each domain has 65 classes.

VisDA-C [32] is a simulation-to-real dataset with two extremely distinct domains: **Synthetic** images, and **Real** images. Each domain has 12 classes. The source domain contains 152 thousand images generated by rendering 3D models, and the target domain has 55 thousand real images sampled from Microsoft COCO [24].

We compare our designed DIPE algorithm with state-of-the-art methods: (1) **ResNet-50**, **ResNet-101** [12]; (2) UDA: Domain Adversarial Network (**DANN**) [10], Adversarial Discriminative Domain Adaptation (**ADDA**) [41], Conditional domain adversarial networks (**CDAN**) [25], Cycle-Consistent Adversarial Domain Adaptation (**CyCADA**) [14], Cluster Alignment with a Teacher (**CAT**) [6], Sliced Wasserstein Discrepancy (**SWD**) [19], Stepwise Adaptive Feature Norm (**SAFN**) [47], Batch Spectral Penalization (**BSP**) [2], Adversarial Dropout Regularization (**ADR**) [35], Margin Disparity Discrepancy (**MDD**) [49], Gradually Vanishing Bridge (**GVB-GD**) [4], Stochastic classifiers (**STAR**) [28], Structurally Regularized Deep

Table 1. Accuracy (%) on Digits.

Method	S→M	U→M	M→U	Avg
Source only [14]	67.1±0.6	69.6±3.8	82.2±0.8	73.0
ADDA [42]	76.0±1.8	90.1±0.8	89.4±0.2	85.2
ADR [35]	95.0±1.9	93.1±1.3	93.2±2.5	93.8
CyCADA [14]	89.2	98.0	95.6	94.3
CDAN [25]	90.4±0.4	96.5±0.1	95.6±0.4	94.2
rRevGrad+CAT [6]	98.8±0.0	96.0±0.9	94.0±0.7	96.3
SWD [19]	98.9±0.1	97.1±0.1	98.1±0.1	98.0
source model only	69.2	87.8	79.1	78.7
SHOT [23]	99.0±0.0	99.0±0.0	97.7±0.1	98.6
MA [22]	**99.4±0.1**	**99.3±0.1**	97.3±0.2	**98.7**
DIPE	99.0±0.0	99.0±0.0	**98.2±0.1**	**98.7**

Table 2. Accuracy (%) on Office-31 (ResNet-50).

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg
ResNet-50 [13]	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DANN [10]	79.7	82.0	68.2	96.9	67.4	99.1	82.2
SAFN+ENT [47]	90.7	90.1	73.0	98.6	70.2	99.8	87.1
rRevGrad+CAT [6]	90.8	94.4	72.2	98.0	70.2	**100.0**	87.6
CDAN [25]	92.9	94.1	71.0	98.6	69.3	**100.0**	87.7
DSBN+MSTN [1]	92.2	92.7	71.7	**99.0**	74.4	**100.0**	88.3
CDAN+BSP [2]	93.0	93.3	73.6	98.2	72.6	**100.0**	88.5
CDAN+BNM [3]	92.9	92.8	73.5	98.8	73.8	**100.0**	88.6
MDD [49]	93.5	94.5	74.6	98.4	72.2	**100.0**	88.9
CDAN+TransNorm [45]	94.0	**95.7**	73.4	98.7	74.2	**100.0**	89.3
GVB-GD [4]	95.0	94.8	73.4	98.7	73.7	**100.0**	89.3
SRDC [40]	95.8	**95.7**	**76.7**	99.2	77.1	**100.0**	**90.8**
source model only	79.5	77.2	62.2	96.1	62.5	98.6	79.4
SHOT [23]	94.8	88.2	73.6	98.4	75.5	99.8	88.4
MA [22]	92.7	93.7	75.3	98.5	**77.8**	99.8	89.6
DIPE	**96.6**	93.1	75.5	98.4	77.2	99.6	90.1

Clustering (**SRDC**) [40]; (3) Source-free domain adaptation: Source Free Domain Adaptation (**SFDA**) [17], Source Hypothesis Transfer (**SHOT**) [23], Model Adaptation (**MA**) [22]. Note that SFDA, SHOT and MA are the previous best source-free domain adaptation methods.

We implement our algorithm in PyTorch. As for some necessary parameters, we set momentum to 0.9, weight decay to $1e^{-3}$, learning rate to $\eta_0 = 1e^{-2}$ for the new layers and the layers learned from scratch in all experiments except $\eta_0 = 1e^{-3}$ for VisDA-C. We further adopt the same learning rate scheduler $\eta = \eta_0(1+10p)^{-0.75}$, where p is changed from 0 to 1. Moreover, we set the batch size to 64, initialize $\beta = 0.3$, $\gamma = 0.3$, $n = 4$, $\epsilon = 0.1$ for all experiments except $\beta = 0.1$ for Digit, and initialize d =0.5 for all experiments expect d =0.6 for Office-Home.

4.2. Results

4.2.1 Results on Digit Recognition

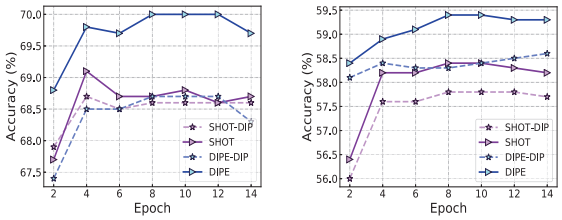
Table 1 reports the classification accuracy of DIPE and other algorithms on the digits. DIPE obtains the best average accuracy on these three tasks compared to all methods. MA receives the same results compared to DIPE at the cost of tremendous computation. DIPE also improves the accuracy of the source model by 20%, demonstrating its effectiveness.

4.2.2 Results on Object Recognition

Tables 2, 3, and 4 report the classification accuracy on three object recognition benchmarks: Office-31, VisDA-C, and Office-Home, respectively. It is clear to see that DIPE significantly outperforms the state-of-the-art methods in Office-Home, improving the average accuracy from 71.3% to 72.5 % without accessing the source data. Meanwhile, DIPE performs the best among 6 out of 12 tasks compared to all methods. For the large-scale and challenging synthesis-to-real VisDA-C dataset, DIPE still achieves the best per-class accuracy. Inside Office-31, DIPE also achieves the best performance on A→D. These results show that by exploring domain-invariant parameters in the feature extractor, we can obtain more domain-invariant representations that further align the unseen source distribution.

4.3. Ablation Studies

Effect of Domain-Invariant Parameters (DIP). We conducted experiments on Office-Home in several methods, e.g., the source model finetuned by pseudo labels predicted by itself, SHOT [23], and our proposed DIPE, aiming to verify DIP's effectiveness. Here '-DIP' indicates that the DIP is not explored in the experiments. In Table 5, we can observe a significant improvement of about 1.0% in challenging tasks, e.g., Cl→Ar, Pr→Cl, and Re→Cl, and a weak improvement in simple tasks, which indicates that exploring DIP is more effective for challenging tasks in SFDA. Moreover, it is clear that exploring DIP brings better performance on all introduced SFDA methods, which verifies that exploring DIP is essential for SFDA. Now the MA [22] is not involved as it requires reproducing the generative and adversarial framework with lots of computation costs. In addition, through the change of accuracy with increasing epoch in Fig. 5, we can observe that exploring DIP not only improves the performance but also stabilizes the effect.



(a) Cl→Ar (b) Re→Cl

Figure 5. Ablation studies on the DIP.

Effect of Clustering Correction. Fig. 6 (a) and (b) demonstrate the advantage of the clustering correction on the challenging tasks in Office-31. (a) represents the accuracy of pseudo labels at the first epoch, where the cluster correcting receives the best result. (b) shows that the

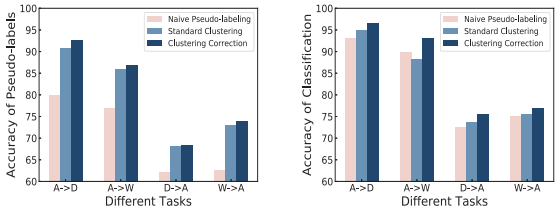
Table 3. Accuracy (%) on VisDA-C (ResNet-101).

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg
ResNet-101 [13]	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DANN [10]	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
ADR [35]	94.2	48.5	84.0	72.9	90.1	74.2	**92.6**	72.5	80.8	61.8	82.2	28.8	73.5
CDAN [25]	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
CDAN+BSP [2]	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SAFN [47]	93.6	61.3	84.	70.6	94.1	79.0	91.8	79.6	89.9	55.6	**89.0**	24.4	76.1
SWD [19]	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
DSBN+MSTN [1]	94.7	86.7	76.0	72.0	**95.2**	75.1	87.9	81.3	91.1	68.9	88.3	45.5	80.2
STAR [28]	95.0	84.0	**84.6**	**73.0**	91.6	91.8	85.9	78.4	**94.4**	84.7	87.0	42.2	82.7
source model only	58.7	20.4	48.2	70.6	63.9	12.1	82.1	18.3	76.4	32.8	87.1	7.4	48.2
SFDA [17]	81.5	79.4	80.3	61.8	92.3	91.9	84.5	82.7	86.5	58.4	74.2	43.5	76.4
MA [22]	94.8	73.4	68.8	74.8	93.1	**95.4**	88.6	**84.7**	89.1	84.7	83.5	48.1	81.6
SHOT [23]	94.8	**87.7**	77.6	53.0	94.0	94.8	82.2	82.6	90.6	87.7	85.5	**58.0**	82.4
DIPE	**95.2**	87.6	78.8	55.9	93.9	95.0	84.1	81.7	92.1	**88.9**	85.4	**58.0**	**83.1**

Table 4. Accuracy (%) on Office-Home (ResNet-50).

Method	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Avg
ResNet-50 [13]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [10]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN [25]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
CDAN+BSP [2]	52.0	68.6	76.1	58.0	70.3	70.1	58.6	50.2	77.6	72.2	59.3	81.9	66.3
SAFN [47]	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
MDD [49]]	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
CDAN+BNM [3]	56.2	73.7	79.0	63.1	73.6	74.0	62.4	54.8	80.7	72.4	58.9	83.5	69.4
GVB-GD [4]	**57.0**	74.7	79.8	64.6	74.1	74.6	65.2	**55.1**	81.0	74.6	**59.7**	84.3	70.4
SRDC [40]	52.3	76.3	**81.0**	69.5	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	**85.0**	71.3
source model only	45.8	67.4	74.1	52.5	61.8	64.7	51.7	42.3	73.8	64.9	47.6	78.2	60.4
SFDA [17]	48.5	71.3	75.6	63.9	69.0	72.1	62.4	43.5	76.0	70.4	50.1	76.1	64.9
SHOT [23]	55.3	78.1	80.5	68.7	76.0	**78.8**	65.7	52.2	82.4	73.1	57.5	84.2	71.0
DIPE	56.5	**79.2**	80.7	**70.1**	**79.8**	**78.8**	67.9	**55.1**	**83.5**	74.1	59.3	84.8	**72.5**

clustering correction also improves the final classification accuracy, suggesting that more accurate pseudo labels can further promote the exploring of domain-invariant parameters. Meanwhile, the results of the last row in Table 6 on Office-Home also verify its effectiveness.



(a) (b)

Figure 6. Ablation studies on the clustering correction.

Effect of Loss Functions. We perform ablation studies to demonstrate the effect of three loss functions in Eq. (10) on the several tasks in Office-Home. As shown in Table 6, we verify that the naive pseudo labeling (PL) [20] is not suitable for the SFDA, then we demonstrate the importance of three loss functions in Eq. (10), since there are noticeable performance gains after adding each loss in them.

Table 6. Ablation study (%) on Office-Home (ResNet-50).

Method	Cl→Ar	Cl→Pr	Cl→Re	Re→Ar	Avg
source model only	52.5	61.8	64.7	64.9	61.0
naive pseudo labeling (PL) [20]	59.4	66.6	70.5	67.7	66.1
L^s_{IM}	66.9	74.7	76.0	73.1	72.7
L^s_{IM} + L^t_{IM}	67.5	75.3	77.4	73.1	73.3
L^s_{IM} + L^t_{IM} + PL [20]	67.0	74.3	75.6	73.1	72.5
L^s_{IM} + L^t_{IM} + Self-supervised PL [23]	68.7	78.1	78.4	73.2	74.6
L^s_{IM} + L^t_{IM} + Cluster Correction	70.1	79.8	78.8	74.1	75.7

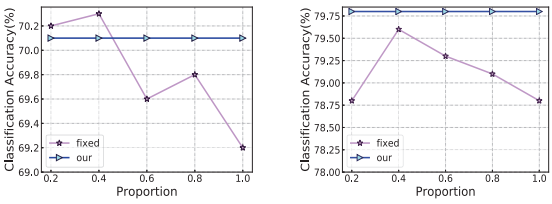
Effect of Domain-Invariant Parameter Proportion. We perform ablation studies to dissect the proportion of domain-invariant parameters on Cl→Ar and Cl→Pr tasks. When the proportion of domain-invariant parameters is set to zero, the accuracy is very low, and it does not converge as all parameters are passively updated, so we do not include this result in the graph. In contrast, when the proportion is set to one, the accuracy is still low in Fig. 7 because all parameters are not treated differently. We can see that the proportion designed incrementally could obtain the best results closely in different tasks.

Effect of Loss Weight. We show the classification accuracy with different γ and β of Eq. (10) in Fig. 8.

Feature Visualization. Fig. 9 (a) and (b) show the t-SNE embedding [7] of target representations on the first 5 classes in a challenging task (Cl→Ar) from the source

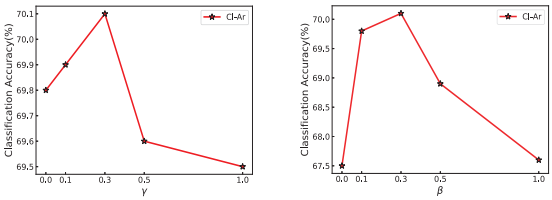
Table 5. Ablation study (%) on Office-Home (ResNet-50) with or without exploring DIP.

Method	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Avg
model - DIP	49.7	72.0	76.0	57.6	66.3	69.3	55.3	45.9	76.3	66.5	51.6	79.7	63.9
model	49.8	73.2	76.3	59.4	66.6	70.5	56.5	46.1	76.9	67.7	52.2	79.9	64.6↑
SHOT - DIP	55.3	78.1	80.5	68.7	76.0	78.8	65.7	52.2	82.4	73.1	57.5	84.2	71.0
SHOT	56.0	78.2	80.9	69.3	75.6	78.9	66.4	53.9	82.5	73.2	58.9	84.0	71.5↑
DIPE - DIP	**57.0**	78.8	80.6	69.2	78.8	**78.8**	**67.9**	54.1	82.9	73.1	58.4	84.6	72.0
DIPE	56.5	**79.2**	**80.7**	**70.1**	**79.8**	**78.8**	**67.9**	**55.1**	**83.5**	**74.1**	**59.3**	**84.8**	**72.5**↑



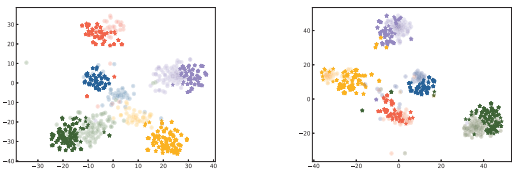
(a) Cl→Ar (b) Cl→Pr

Figure 7. Ablation studies on the effect of domain-invariant parameters' proportion.



(a) Cl→Ar (b) Cl→Ar

Figure 8. Ablation studies on the loss weight.



(a) source model (b) ours

Figure 9. The t-SNE visualization of target representations for the first 5 class classification task. Stars '*' in dark colors denote unseen source data and circle 'o' in light colors denote target data. Different colors represent different classes.

and learned models. It is clearly shown that the target feature representations learned by DIPE are more consistent (b) across domains than the given source model (a), verifying DIPE's effectiveness.

5. Conclusion

In this paper, we propose a novel method to explore domain-invariant parameters stored in the well-trained source model for Source Free Domain Adaptation (SFDA). It effectively alleviates the domain shift problem since the learned domain-invariant parameters can promote learning domain-invariant representations. Extensive experiments on image classification have demonstrated that our method could achieve more accurate performance in various privacy-preserving applications. The idea behind domain-invariant parameters exploring is simple and orthogonal to other methods. One can extend our work to various practical SFDA algorithms. Thus our approach opens up a new perspective for SFDA. In future work, better parameter judgment criteria and update strategies can be investigated.

Broader Impact

Recently the success of domain adaptation algorithms has depended on the large scale of labeled source data, which is impractical in privacy-preserving scenarios. The positive impact of our work is to improve the robustness and generalizability of deep neural networks where the domain shift meets the data privacy protection. While we show improved performance relative to state-of-the-art, the negative transfer could still occur. Therefore our approach should not be used in mission-critical applications or to make essential decisions without human oversight.

Limitations

While we can verify that exploring domain-invariant parameters is critical for SFDA through the effect improvement, the existence of the domain-invariant parameters is challenging to prove due to the lacking of the theoretical guarantees and interpretability of deep networks. Further, SFDA relies on the well-trained source model that may be impaired by some causes, e.g., the source model training process. In these unforeseen circumstances, the robustness of SFDA methods would face serious challenges.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (62176139, 61876098), the Major Basic Research Project of Natural Science Foundation of Shandong Province (ZR2021ZD15), the Young Elite Scientists Sponsorship Program by CAST.

References

[1] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019. 6, 7

[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. PMLR, 2019. 5, 6, 7

[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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3941–3950, 2020. 6, 7

[4] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12455–12464, 2020. 5, 6, 7

[5] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 4

[6] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9944–9953, 2019. 5, 6

[7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 647–655. JMLR.org, 2014. 7

[8] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018. 2, 3

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 2

[10] 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. 1, 5, 6, 7

[11] Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. 2010. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 5

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 7

[14] 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. 5, 6

[15] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608, 2006. 2

[16] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994. 5

[17] Youngeun Kim, Donghyeon Cho, and Sungeun Hong. Towards privacy-preserving domain adaptation. *IEEE Signal Processing Letters*, 27:1675–1679, 2020. 1, 2, 6, 7

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[19] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019. 5, 6, 7

[20] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 7

[21] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9757–9766, June 2021. 4

[22] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020. 2, 6, 7

[23] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data ? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 1, 2, 3, 4, 5, 6, 7

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[25] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1647–1657, 2018. 1, 2, 5, 6, 7

[26] 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. PMLR, 2017. 2

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 4

[28] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9111–9120, 2020. 5, 7

[29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[30] Yingwei Pan, Ting Yao, Yehao Li, Chong-Wah Ngo, and Tao Mei. Exploring category-agnostic clusters for open-set domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 4

[31] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054*, 2019. 3

[32] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 5

[33] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 5

[34] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997. PMLR, 2017. 2

[35] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. *arXiv preprint arXiv:1711.01575*, 2017. 5, 6, 7

[36] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 2

[37] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016. 2

[38] Serban Stan and Mohammad Rostami. Privacy preserving domain adaptation for semantic segmentation of medical images. *arXiv preprint arXiv:2101.00522*, 2021. 1

[39] Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008. 2

[40] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 4, 6, 7

[41] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 5

[42] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 6

[43] 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. 5

[44] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 1, 3

[45] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable normalization: Towards improving transferability of deep neural networks. 2019. 6

[46] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 4

[47] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1426–1435, 2019. 5, 6, 7

[48] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3801–3809, 2018. 2

[49] 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. PMLR, 2019. 1, 5, 6, 7