Denoised Maximum Classifier Discrepancy for Source-Free Unsupervised Domain Adaptation

Tong Chu, Yahao Liu, Jinhong Deng, Wen Li, Lixin Duan*

School of Computer Science and Engineering & Shenzhen Institute for Advanced Study,
University of Electronic Science and Technology of China
{uestcchutong, lyhaolive, jhdeng1997, liwenbnu, lxduan}@gmail.com

Abstract

Source-Free Unsupervised Domain Adaptation (SFUDA) aims to adapt a pre-trained source model to an unlabeled target domain without access to the original labeled source domain samples. Many existing SFUDA approaches apply the self-training strategy, which involves iteratively selecting confidently predicted target samples as pseudo-labeled samples used to train the model to fit the target domain. However, the self-training strategy may also suffer from *sample selection bias* and be impacted by the *label noise* of the pseudo-labeled samples. In this work, we provide a rigorous theoretical analysis on how these two issues affect the model generalization ability when applying the self-training strategy for the SFUDA problem. Based on this theoretical analysis, we then propose a new Denoised Maximum Classifier Discrepancy (D-MCD) method for SFUDA to effectively address these two issues. In particular, we first minimize the distribution mismatch between the selected pseudo-labeled samples and the remaining target domain samples to alleviate the sample selection bias. Moreover, we design a strong-weak self-training paradigm to denoise the selected pseudo-labeled samples, where the strong network is used to select pseudo-labeled samples while the weak network helps the strong network to filter out hard samples to avoid incorrect labels. In this way, we are able to ensure both the quality of the pseudo-labels and the generalization ability of the trained model on the target domain. We achieve state-of-the-art results on three domain adaptation benchmark datasets, which clearly validates the effectiveness of our proposed approach. Full code is available at https://github.com/kkkkkkon/D-MCD.

Introduction

Benefiting from the large amount of labeled training data available, deep neural networks have achieved promising results in many computer vision tasks. However, it is often highly costly to build a large-scale labeled dataset for deep neural network training. To this end, the Unsupervised Domain Adaptation (UDA) was devised, the goal of which to leverage a labeled source domain to help the training of models on a new unlabeled target domain, thus saving the cost of annotating training samples for the new domain.

While many methods have been proposed to solve the UDA problem (Ganin and Lempitsky 2015; Long et al. 2018; Saito et al. 2018; Dong et al. 2020; Liu et al. 2021; Deng et al. 2021; Dong et al. 2021), these approaches is needed to access the source domain data during the training process. This limits application of the UDA approach in many real-world scenarios. For example, in visual recognition tasks involving medical images, surveillance videos, or fingerprint images, accessing these data often introduces privacy issues.

To avoid accessing the source domain data in the domain adaptation process, a more challenging UDA setting has been is proposed, named Source-Free Unsupervised Domain Adaptation (SFUDA) (Liang, Hu, and Feng 2020). Under this setting, we are only given a source model pre-trained on the source domain and unlabeled samples in the target domain; the goal is to improve the performance of the model on the target domain without access to the original labeled source domain data.

One straightforward way to address SFUDA is to apply a self-training strategy, and many methods using this way have been proposed (Liang, Hu, and Feng 2020; Chen et al. 2021; Tian et al. 2021). The core concept involves using the trained model to select a set of confidently predicted samples from the target domain, which are likely to be correctly labeled, and then use these selected pseudo-labeled samples to refine the model. This process is then iterated such that the model can be gradually improved.

However, there are also risks associated with the self-training strategy. First, there exists a sample selection bias when selecting pseudo-labeled samples from the target domain, which inevitably limits the model's generalization ability on the entire target domain. Second, the pseudo-labeled samples often contain significant label noise, which also harms the model performance. While several heuristics designed have been proposed (Liang, Hu, and Feng 2020; Chen et al. 2021) to improve the label quality, existing works on this topic have not entirely resolved these two issues.

In this work, we provide a rigorous theoretical analysis of how the sample selection bias and the label noise of pseudo-labeled samples affect the target model's generalization ability when the self-training strategy is applied for the SFUDA problem. Building upon the generalization bound for the traditional UDA problem, we provide a generalization bound for the SFUDA problem. We prove that the generalization ability of the target model can be bounded by the target train-

*Corresponding author.

ing error with the pseudo-labeled samples, the label noise of the pseudo-labeled samples, the distribution mismatch between the selected pseudo-labeled samples and the remaining target samples, and other constant terms. This validates our analysis of the two risks when using self-training for the SFUDA problem.

Based on the generalization bound, we then propose a new SFUDA approach called Denoised Maximum Classifier Discrepancy (D-MCD). First, we remold the Bi-Classifier Determinacy Maximization (BCDM) to adapt the pre-trained source model to the target domain using unlabeled target domain samples, enabling us to obtain a good enough initial target model for self-training. We then begin the self-training process in which we explicitly consider the two risks discussed above. When training the target model with the selected pseudo-label samples, we also pay attention to the distribution mismatch between these selected samples and the remaining target domain samples. The BCDM approach is again applied during the self-training process to reduce this distribution mismatch, such that the generalization ability can be guaranteed on the entire target domain.

Furthermore, we design a strong-weak self-training paradigm to reduce the label noise in the selected pseudo-label samples. As the initial target model trained with pseudo-label samples often produces high-confidence but incorrect predictions, we additionally train another model from scratch with pseudo-label samples to help filter out these hard samples. This approach is motivated by the fact that a model tends to remember easy samples during the early stage of the training process; accordingly, the newly trained weak model is able to identify hard examples for the strong initial target model, thereby reducing its production of high-confidence and incorrect predictions. We implement this by gradually ensembling the model parameters of the weak model to the strong model until the weak model is sufficiently strongly trained.

In summary, the contributions of this paper are as follows:

- We provide a generalization bound for the SFUDA problem, which reveals the impacts of sample selection bias and the label noise of pseudo-labeled samples when applying self-training for the SFUDA problem.
- We propose a new D-MCD approach for the SFUDA problem, in which we simultaneously reduce the data distribution mismatch between the selected pseudo-labeled samples and the remaining target domain samples, and improve the label quality of pseudo-labeled samples by means of a strong-weak self-training paradigm.
- We evaluate our proposed approach on three domain adaptation benchmark datasets and achieve state-of-the-arts results.

Related Works

Unsupervised Domain Adaptation Conventional UDA methods reduce the domain discrepancy between the source and target domains in the feature space and rely on matching the high-order moments of the source domain and the target domain (Tzeng et al. 2014; Long et al. 2017; Gretton et al. 2012) or conducting adversarial training through the domain discriminator to learn domain-invariant features (Ganin and Lempitsky 2015; Long et al. 2018). In addition, there is also a special kind of adversarial method that depends on the adversarial training between the feature extractor and the classifier (Saito et al. 2018; Li et al. 2021; Lee et al. 2019; Lu et al. 2020). These methods decouple the source and the target domain data during the training process; that is, they enable estimation of the difference between the source domain and the target domain without having access to the source domain data. In addition to the inter-domain alignment method, some methods consider intra-domain alignment (Pan et al. 2020) or fit the target distribution in a straightforward manner (Wang and Breckon 2020; Liu et al. 2021). They are often dependent on the accuracy of prototype estimation and the accuracy of pseudo-label annotation.

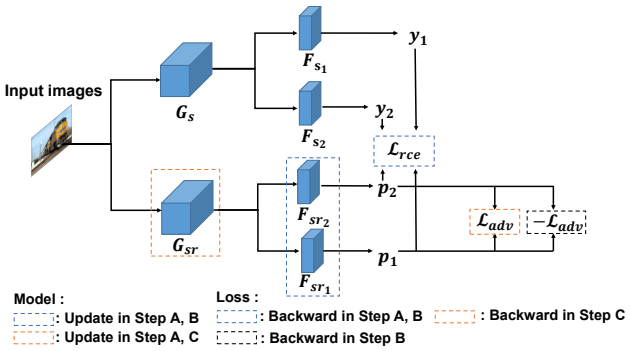

Figure 1: Overview of model adaptation. There are three steps (*i.e.*, step A, B, C) during the model adaptation, with different colors of dashed-line boxes indicating either a move backward or an update in the corresponding step.

Source-Free Unsupervised Domain Adaptation SFUDA focuses on adapting the model to the target domain without accessing the source domain data. Some SFUDA methods (Qiu et al. 2021; Tian et al. 2021) mainly focus on reconstructing the fake source distribution in the feature space according to the source hypothesis and further improve the generalization ability by aligning the target domain samples with the pseudo source domain samples. Another stream of SFUDA methods (Liang, Hu, and Feng 2020; Chen et al. 2021; Yang et al. 2020) exploit pseudo label prediction from the source model or prototype to adapt the model to the target domain so that the model is well-fitted to the target domain distribution.

Noise Label Learning Noise label learning refers to reducing the influence of noise labels and improving model performance when dataset label noise is present. The regularization method involves a regularization term in the training loss to avoid overfitting on noise labels (Wang et al. 2019; Müller, Kornblith, and Hinton 2019). Previous work (Arpit et al. 2017) has shown that deep networks tend to memorize easy samples first, and then memorize hard samples during the training process. Based on this observation, some approaches (Han et al. 2018; Yu et al. 2019) have achieved good results by filtering labels to reduce the accumulation of errors.

Revisiting Self-training for Source-Free Unsupervised Domain Adaptation

In the SFUDA problem, we are given a source model pretrained on the labeled source domain and an unlabeled target domain $\mathcal{D}$; moreover, $\hat{\mathcal{D}} = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$ where $\mathbf{x}_i^t$ is sampling from $\mathcal{D}$ and n_t is the total number of samples in the target domain. The goal of SFUDA is to adapt the source model to the unlabeled target domain $\mathcal{D}$ without having access to the original labeled source domain samples.

When applying the self-training strategy to solve the SFUDA problem, the target domain is divided into two subsets; a high-confidence sample set $\hat{\mathcal{D}}_h = \{\mathbf{x}_i^h\}_{i=1}^{n_h}$ and a low-confidence sample set $\hat{\mathcal{D}}_l = \{\mathbf{x}_i^l\}_{i=1}^{n_l}$. Usually, we have $\hat{\mathcal{D}} = \hat{\mathcal{D}}_h \cup \hat{\mathcal{D}}_l$ and $\hat{\mathcal{D}}_h \cap \hat{\mathcal{D}}_l = \emptyset$. Each high-confidence sample $\mathbf{x}_i^h$ in $\hat{\mathcal{D}}_h$ is then provided with a pseudo-label y_i by using the predictions made by a certain model (*e.g.*, the pretrained source model or the target model from the previous training stage). For ease of presentation, we here redefine the high-confidence set as $\hat{\mathcal{D}}_h = \{(\mathbf{x}_i^h, y_i)\}_{i=1}^{n_h}$ where y_i corresponds to the pseudo-label of x_i^h.

At first glance, when applying the self-training strategy for solving the SFUDA problem, we are facing a semi-supervised learning problem(Wang, Li, and Gool 2019), as we have a pseudo-labeled training set $\hat{\mathcal{D}}_h$ and an unlabeled training set $\hat{\mathcal{D}}_l$. However, the self-training problem poses additional challenges. Specifically, the selection of pseudo-labeled samples (*i.e.*, the high-confidence set) inevitably involves a sample selection bias; in other words, the sample distributions of the high-confidence set $\hat{\mathcal{D}}_h$ and the low-confidence set $\hat{\mathcal{D}}_l$ are usually different, meaning that a model trained with these selected pseudo-labeled samples cannot generalize well to the entire target domain. Moreover, as the labels of the pseudo-labeled samples are obtained from model predictions rather than human annotation, there is often considerable noise in these labels.

To verify our above analysis, we derive a generalization error bound for the SFUDA problem. In more detail, following the terminology proposed by (Li et al. 2021), we define h as a learnt hypothesis, and f_p (*resp.*, f_h) as a labeling function that outputs the pseudo-labels (*resp.*, ground truth labels) for the high-confidence target samples. We further define $\mathcal{E}_{\hat{\mathcal{D}}_h}(h, f_p)$ (*resp.*, $\mathcal{E}_{\hat{\mathcal{D}}_h}(f_h, f_p)$) as the empirical estimation of the discrepancy between the learnt hypothesis h (*resp.*, the ground-truth labeling function f_h) and the pseudo-labeling function f_p on the high-confidence samples. Let us represent the generalization error on the target domain of the learned hypothesis h as $\mathcal{E}_{\hat{\mathcal{D}}}(h)$; thus, the generalization bound for the SFUDA problem can be described as follows:

Theorem 1 *Given any $\delta \geq 0$, for any hypothesis $h \in \mathcal{H}$ where $\mathcal{H}$ is a hypothesis set, the following generalization bound holds with at least a probability of $1 - 3\delta$:*

$$\begin{aligned}\mathcal{E}_{\mathcal{D}}(h) \leq \mathcal{E}_{\hat{\mathcal{D}}_h}(h, f_p) + \mathcal{E}_{\hat{\mathcal{D}}_h}(f_p, f_h) + \\ (1-r) d_{h,\mathcal{H}}(\hat{\mathcal{D}}_h, \hat{\mathcal{D}}_l) + (1-r)\lambda + \Omega,\end{aligned} \tag{1}$$

where $d_{h,\mathcal{H}}(\hat{\mathcal{D}}_h, \hat{\mathcal{D}}_l)$ represents the distribution mismatch between the selected pseudo-labeled samples and the remaining target samples, λ and Ω are constant terms , and $r = \frac{n_h}{n_t}$ is the samples selection ratio for $\hat{\mathcal{D}}_h$.

The proof is provided in the Supplementary section. From the generalization, we can observe that, in addition to the constant term λ and Ω, the generalization error of the target hypothesis h is bounded by three terms: the target training error with the pseudo-labeled samples $\mathcal{E}_{\hat{\mathcal{D}}_h}(h, f_p)$, the label noise of the pseudo-labeled samples $\mathcal{E}_{\hat{\mathcal{D}}_h}(f_p, f_h)$, and the distribution mismatch between the selected pseudo-labeled samples and the remaining target samples $d_{h,\mathcal{H}}(\hat{\mathcal{D}}_h, \hat{\mathcal{D}}_l)$. This indicates that, in the process of self-training for the SFUDA problem, in addition to minimizing the training error using the pseudo-labeled samples (*i.e.*, the first term), it is necessary to pay attention to the noise in the pseudo-labels of the confidence samples (*i.e.*, the second term), as well as the distribution difference between the high-confidence and low-confidence samples (*i.e.*, the third term).

Denoised Maximum Classifier Discrepancy

Based on the analysis on the generalization bound for the SFUDA problem, we propose a new SFUDA approach, called Denoised Maximum Classifier Discrepancy (D-MCD), in which we improve the self-training strategy by reducing the noise in the pseudo-labels of the confidence samples along with the distribution difference between the confidence and non-confidence samples. Specifically, we base our D-MCD approach on the improved MCD (Saito et al. 2018) method BCDM (Li et al. 2021). As self-training usually requires an initial model that is strong enough to perform sufficiently well on the target domain, we first adapt the pretrained source model to the target domain using unlabeled target samples, referred to as the *Model Adaptation* phase. We then begin the self-training and simultaneously address the label noise and sample selection bias issues, referred to as the *Model Self-Training* phase.

Model Adaptation

The BCDM (Li et al. 2021) method was proposed to address the traditional unsupervised domain adaptation problem, in which the labeled source domain samples are available during the training process. The generalization bound satisfies:

$$\mathcal{E}_{\mathcal{T}}(h) \leq \mathcal{E}_{\widehat{\mathcal{S}}}(h) + d_{h,\mathcal{H}}(\widehat{\mathcal{S}}, \widehat{\mathcal{T}}) + \lambda + \hat{\Omega}, \tag{2}$$

where $d_{h,\mathcal{H}}(\mathcal{S}, \mathcal{T}) \triangleq \sup_{h' \in \mathcal{H}} \left(\mathrm{dis}_{\mathcal{S}}\left(h', h\right) - \mathrm{dis}_{\mathcal{T}}\left(h', h\right)\right)$, while λ and $\hat{\Omega}$ are constant terms.

The above bounds can be optimized by means of adversarial training between the classifiers f and feature extractor g. Therefore, the training process in the BCDM method can be summarized by the following three steps:

Step A Optimize the cross-entropy loss ℓ_1, ℓ_2 calculate by the model output for the source sample and source label to keep $\mathcal{E}_{\widehat{\mathcal{S}}}(h)$ and $\mathrm{dis}_{\mathcal{S}}\left(h', h\right)$ small enough so that the generalization bound in Eq. 2 still holds.

$$\min_{G, F_1, F_2} \ell_1\left(F_1\left(G\left(\mathbf{x}_s\right)\right), \mathbf{y}_s\right) + \ell_2\left(F_2\left(G\left(\mathbf{x}_s\right)\right), \mathbf{y}_s\right)$$

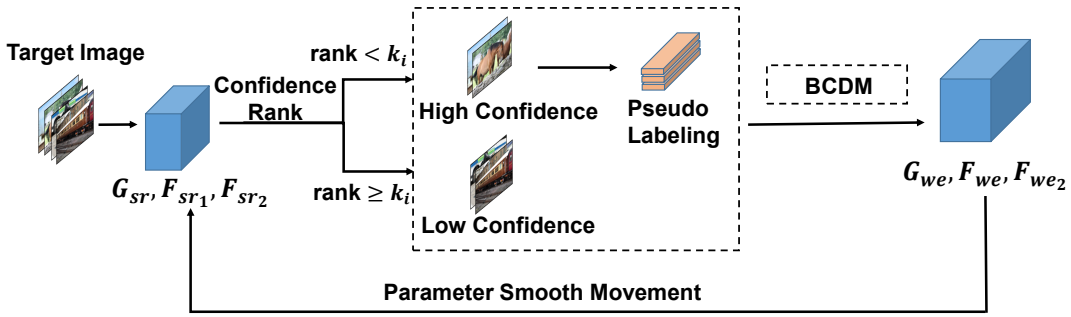

Figure 2: The pipeline of the strong-weak self-training paradigm. The target images are first fed into the strong model (*i.e.*, G_{sr}, F_{sr_1} and F_{sr_2}) to divide the target domain into a high-confidence and low-confidence sample split. These two sets of samples are used to train the weak model (*i.e.*, G_{we}, F_{we_1} and F_{we_2}) using BCDM (Li et al. 2021), meaning that it can also feed back to help the strong model filter out hard samples and thus avoid incorrect labels by parameter smooth movement.

Step B Adopt the CDD distance (Li et al. 2021) as $d(.,.)$ to measure the classifier output discrepancy (adversarial loss $\mathcal{L}_{adv}$). To maximize the CDD distance with the classifier and train with ℓ_1, ℓ_2 to maintain stability.

$$\min_{F_1,F_2} \ell_1 + \ell_2 - \gamma d(F_1\left(G\left(\mathbf{x}_t\right)\right), F_2\left(G\left(\mathbf{x}_t\right)\right))$$

Step C To minimize the CDD distance with feature extractor G.

$$\min_{G} \gamma d(F_1\left(G\left(\mathbf{x}_t\right)\right), F_2\left(G\left(\mathbf{x}_t\right)\right))$$

Remolding BCDM for SFUDA In the SFUDA problem, the labeled samples in the source domain are not accessible during training; accordingly, the BCDM cannot be directly applied to the SFUDA problem, since the loss ℓ_1 and ℓ_2 in Steps A and B cannot be optimized due to the lack of available labeled source domain samples.

According to the setting of the SFUDA problem, as the source model is trained on the source domain, it is reasonable to assume that the error performance of the model on the source domain is also extremely small, meaning that the above generalization bound still holds. However, due to the lack of original labeled source domain data, we cannot calculate the loss function ℓ_1, ℓ_2 in Steps A and B. Training only with steps B and C and without ℓ_1, ℓ_2 may cause the model's error on the source domain to increase. Thus, to remold BCDM for SFUDA, we cannot only train with adversarial training, it is also necessary to maintain the performance of the model in the source domain. Specifically, assume that we are given a pre-trained source domain includes two branches of classification heads F_{s_1}, F_{s_2}, and a common feature extractor G_s and we initialize model $F_{sr_1}, F_{sr_2}, G_{sr}$ with source model. To address this problem, we propose to replace these loss functions with reverse cross-entropy loss (RCE loss function):

$$\begin{aligned} \ell_{rce_1} &= -\sum_{k=1}^{K} p_1(k|\mathbf{x}_i^t) \log q_1(k|\mathbf{x}_i^t) \\ \ell_{rce_2} &= -\sum_{k=1}^{K} p_2(k|\mathbf{x}_i^t) \log q_2(k|\mathbf{x}_i^t) \end{aligned} \tag{3}$$

where $\mathbf{x}_t$ is target domain sample, while $q_1(k|\mathbf{x}_t)$ and $q_2(k|\mathbf{x}_t)$ are the respective the outputs of the source model from the classifier for the k-th class. Moreover, F_{s_1} and F_{s_2}, and the $p_1(k|\mathbf{x}_t)$ and $p_2(k|\mathbf{x}_t)$ are respectively the outputs of the trained model from the branch classifier F_{sr_1} and F_{sr_2}. As shown in Figure 1, we train Step A and Step B to optimize the RCE loss function between the current model output and the source model output.

By taking the soft labels from the pre-trained source model, the traditional cross-entropy (CE) loss can also be used as an alternative to the above RCE loss, because it can be used as a regular term to keep the model from collapsing during training. However, compared with the CE loss function, the RCE loss function pays attention not only to the consistency of the output and the label, but also to the confidence of the label. The RCE loss function has a large sample gradient for high-confidence labels and a small sample gradient for lower-confidence labels, as discussed below,

Properties of RCE Loss For distribution p, q, the RCE loss function (Wang et al. 2019) is calculated as follows:

$$\ell_{rce} = -\sum_{k=1}^{K} p(k|\mathbf{x}) \log q(k|\mathbf{x}) \tag{4}$$

we calculate the gradient of the RCE loss function to the j-th element z_j output by the neural network and fixing the probability p to analyze the influence of q on the gradient, we find that when the predicted probability q is a uniform probability vector, such as $[\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}]$, the gradient of the RCE loss function is:

$$\begin{aligned} \frac{\partial \ell_{rce}}{\partial z_j} &= p_j(\sum_{k=1}^{K} p_k \log q_k - \log q_j) \\ &= p_j(\sum_{k=1}^{K} p_k \log \frac{1}{K} - \log \frac{1}{K}) = 0 \end{aligned} \tag{5}$$

When the predicted probability q is a one-hot vector, the absolute value of the gradient reaches its maximum. Moreover, the gradient of the CE loss function to the j-th element z_j output by the neural network is as follows:

$$\frac{\partial \ell_{ce}}{\partial z_j} = p_j - q_j \tag{6}$$

This shows that the gradient of the CE loss function focuses only on consistency while ignoring confidence. Therefore, by optimizing the RCE loss function in each step, the model can not only learn according to the confidence of the soft label but is also able to maintain its performance on the high-confidence samples.

Model Self-Training

After obtaining a sufficiently good initial model, we next discuss how to improve the self-training process. In particular, when training the target model with the selected pseudo-labeled high-confidence samples, we design a strong-weak self-training paradigm, in which the strong and weak model training helps to remove high-confidence but incorrect labels, and also employ the BCDM (Li et al. 2021) method to reduce the distribution mismatch between the high-confidence and low-confidence samples. The overview of model self-training is presented in Fig. 2.

Strong-Weak Self-Training Paradigm We represent the strong model as $G_{sr}, F_{sr_1}, F_{sr_2}$ with parameter θ_{sr}. As it has been fine-tuned on the target domain, it typically outputs confident predictions for the target domain samples, even though a number of these predictions might be wrong. As shown by the analysis in Theorem 1, this will significantly increase the second noise label term in the generalization bound, and thus degrade the generalization performance of the model in the target domain.

To reduce the number of high-confidence but incorrect labels, we additionally train a weak model $G_{we}, F_{we_1}, F_{we_2}$ with parameter θ_{we} from a model that has not been trained on either the source or the target domain *e.g.*, an ImageNet pre-trained model. Our motivation is based on the observation that deep networks tend to memorize correctly labeled samples first (Arpit et al. 2017) and then memorize label noise samples during the training process. Therefore, the weak model will tend to first remember the high-confidence and correct label during the training process while ignoring the high-confidence but incorrect labels. We accordingly use the weak model help the strong model to filter out these high-confidence but incorrectly predicted samples. More specifically, during the training process, we use the method of smooth parameter movement to fuse the parameters of the weak model with those of the strong model at the end of each epoch.

$$\theta_{sr} = \alpha\theta_{sr} + (1-\alpha)\theta_{we} \tag{7}$$

In this way, the addition of parameters from the weak model helps to increase the confidence score of easy and correct samples, thus encouraging them to enter the high-confidence sample set. At the same time, the reduction of the original parameter of the strong model helps to reduce the confidence of high-confidence but incorrect samples, meaning that these samples will tend to be filtered out of the high-confidence sample set. After a period of training, due to the reduction of the influence of noisy labels, the weak model will continue to grow stronger, even to the extent that the predictive accuracy of its high-confidence samples will exceed that of the original strong model. Thus we abandon the original strong model and use the stronger current model for training. Specifically, we set $\alpha = 1$ when the model's cross-entropy loss function $\mathcal{L}_{ce} < 0.5$ for noise labels. By comparing the accuracy of the false labels of a fixed proportion of high-confidence samples before and after denoising on several datasets, we can confirm the effectiveness of our method (see Supplementary for the details).

Training process of D-MCD We separate the target domain $\mathcal{D}$ into a high-confidence domain $\mathcal{D}_h$ and a low-confidence domain $\mathcal{D}_l$. Moreover, we use the strong model to assign pseudo-labels to $\mathcal{D}_h$ samples. Therefore, we can use the UDA method BCDM (Li et al. 2021) to align the labeled domain $\mathcal{D}_h$ and the unlabeled domain $\mathcal{D}_l$. We refer to the steps A, B, C mentioned in the model adaptation chapter and iteratively perform the following steps:

Step 1 Similar to the step A mentioned above, we replace the source domain sample $\mathbf{x}_s$ and label y_s with high-confidence domain samples $\mathbf{x}^h$ and pseudo-label y_h.

Step 2 Similar to step B mentioned above, we replace the source domain sample $\mathbf{x}_s$ and label y_s with high-confidence domain samples $\mathbf{x}^h$ and pseudo-label y_h to calculate cross-entropy loss. Moreover, we calculate CDD distance with samples $\mathbf{x}^l$ of low-confidence domain instead of samples from all the target domain.

Step 3 Similar to the above-mentioned step C, we calculate CDD distance with samples $\mathbf{x}^l$ of low-confidence domain rather than all samples from all the target domain.

Details of Selecting High-confidence Samples After model adaptation, we obtain a strong enough model $G_{sr}, F_{sr_1}, F_{sr_2}$ that can better predict results on the target domain. To separate the target samples, we select CDD distance (Li et al. 2021) as the measure of the sample confidence level, since CDD distance can measure the consistency and confidence of the two classifier outputs.

Given a ranking of CDD distance for each sample in the target domain, hyper-parameter r is introduced as a ratio to separate the target images in a class-wise manner into high-confidence and low-confidence domain. Specifically, for each category, we select the top ratio r samples to construct a high-confidence domain and the remaining samples to construct a low-confidence domain.

In addition, to prevent the impact of unbalanced sample numbers when selecting by category, we estimate the expected sample interval for each category. We define the number of categories as K; thus, the expected number of high-confidence samples for the i-th category is $E_i(r) = r\frac{n_t}{K}$. For each category, we construct an interval $[a_i, b_i] = [E_i(r-c), E_i(r+c)]$, where c represents the balance ratio. So the number of samples selected for the i-th category as $k_i = \min(b_i, \max(k_i, a_i))$. This operation helps to ensure a balanced number of samples across each category in the constructed high-confidence domain.

Experimental Setup

Datasets We evaluate our method on three widely used UDA benchmark datasets: 1) VISDA (Peng et al. 2017), a

Method	Source-Free	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
ResNet-101 (He et al. 2016)	✗	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
MCD (Saito et al. 2018)	✗	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
CDAN (Long et al. 2018)	✗	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
SWD (Lee et al. 2019)	✗	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
STAR (Lu et al. 2020)	✗	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
BCDM (Li et al. 2021)	✗	95.1	87.6	81.2	73.2	92.7	95.4	86.9	82.5	95.1	84.8	88.1	39.5	83.4
SHOT (Liang, Hu, and Feng 2020)	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
MA (Li et al. 2020)	✓	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
G-SFDA (Yang et al. 2021)	✓	96.1	88.3	85.5	74.1	**97.1**	95.4	89.5	79.4	**95.4**	**92.9**	89.1	42.6	85.4
SSNLL (Chen et al. 2021)	✓	**97.2**	87.7	89.1	73.6	96.1	91.2	**92.7**	79.9	94.2	89.0	**90.4**	48.9	85.8
VDM-DA (Tian et al. 2021)	✓	96.9	89.1	79.1	66.5	95.7	96.8	85.4	83.3	96.0	86.6	89.5	56.3	85.1
CPGA (Qiu et al. 2021)	✓	95.6	**89.0**	75.4	64.9	91.7	97.5	89.7	83.8	93.9	93.4	87.7	**69.0**	86.0
D-MCD (ours)	✓	97.0	88.0	**90.0**	**81.5**	95.6	**98.0**	86.2	**88.7**	94.6	92.7	83.7	53.1	**87.5**

Table 1: Classification accuracy (%) on the VISDA dataset (ResNet-101). ✓ indicates the SFUDA method, and ✗ indicates the UDA method. Bold text indicates the best results.

Method	Source-Free	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg.
ResNet-50 (He et al. 2016)	✗	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 (Ganin and Lempitsky 2015)	✗	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
DAN (Long et al. 2015)	✗	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
CDAN (Long et al. 2018)	✗	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
SPL (Wang and Breckon 2020)	✗	54.5	77.8	**81.9**	65.1	78.0	**81.1**	66.0	53.1	**82.8**	69.9	55.3	**86.0**	71.0
SHOT (Liang, Hu, and Feng 2020)	✓	57.1	78.1	81.5	**68.0**	78.2	78.1	**67.4**	54.9	82.2	**73.3**	58.8	84.3	71.8
G-SFDA (Yang et al. 2021)	✓	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3
CPGA (Qiu et al. 2021)	✓	59.3	78.1	79.8	65.4	75.5	76.4	65.7	**58.0**	81.0	72.0	**64.4**	83.3	71.6
D-MCD (ours)	✓	**59.4**	**78.9**	80.2	67.2	**79.3**	78.6	65.3	55.6	82.2	**73.3**	62.8	83.9	**72.2**

Table 2: Classification accuracy (%) on the Office-Home dataset (ResNet-50). ✓ indicates the SFUDA method, and ✗ indicates the UDA method. Bold text indicates the best results.

large-scale challenging dataset with 12 classes; 2) Office-Home (Venkateswara et al. 2017), a medium-sized image classification dataset with four distinctive domains (Art (A), Clipart (C), Product (P), and RealWorld (R)); 3) Office31 (Saenko et al. 2010), a small-sized image classification dataset comprising three different domains (Amazon (A), DSLR (D), and Webcam (W))

Experiment Details We first train a model using the labeled source samples, then employ our proposed D-MCD method to improve the target model performance on the target domain, where only unlabeled target samples are available while the labeled source samples are inaccessible. We use the data transform method for high-confidence samples adopts from (French, Mackiewicz, and Fisher 2018) and we also adopt the consist loss for different data transform samples. For the Office31 dataset, we calculate probability using the ensemble of feature level probability and classifier output probability, and we generate the prototype following SHOT (Liang, Hu, and Feng 2020); moreover, to balance the model obtained by Model Adaptation training and Model Self-Training, the ensemble output of these two models will be used as the result.

Network Architecture We follow the network architecture presented in the BCDM (Li et al. 2021) method. The feature extractor is initialized with the ResNet50/101 model pre-trained on ImageNet (Deng et al. 2009), and we replace the last fully connected layer with the bottleneck layer. A classifier with three fully connected layers is used for the VISDA dataset, and a classifier with two fully connected layers is used for the Office-Home and Office31 datasets.

Network Hyper-parameters We set the following hyper-parameters for RCE loss $\beta = 0.1$ for Office and $\beta = 0.001$ for VISDA, $\gamma = 0.0025$ in training step B and C, $r = 0.4$ and $c = 0.2$ for VISDA and Office-Home and $r = 0.6$, $c = 0.1$ for Office31. We adopt the Stochastic Gradient Descent optimizer (SGD) with momentum 0.9 and weight decay 5×10^{-4} and the same learning rate scheduler $\eta = \eta_0 \cdot (1 + 10 \cdot p)^{-0.75}$ where p is the training progress changing from 0 to 1. For the VISDA dataset, the learning rates for the feature extractor and the feature classifier are set to 3×10^{-4} and 1×10^{-3} respectively. For the Office-Home and Office31 datasets, learning rate of the feature extractor is 3×10^{-3} and the learning rate of the feature classifier is 1×10^{-2}. Moreover, we exploit the same entropy loss (Long et al. 2016) following (Saito et al. 2018; Li et al. 2021)

Method	Source-Free	A → D	A → W	D → A	D → W	W → A	W → D	Avg.
ResNet-50 (He et al. 2016)	✗	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DANN (Ganin and Lempitsky 2015)	✗	79.7	82.0	68.2	96.9	67.4	99.1	82.2
DAN (Long et al. 2015)	✗	78.6	80.5	63.6	97.1	62.8	99.6	80.4
CDAN (Long et al. 2018)	✗	92.9	94.1	71.0	98.6	69.3	**100.0**	87.7
BCDM (Li et al. 2021)	✗	93.8	**95.4**	73.1	98.6	71.6	**100.0**	89.0
SHOT (Liang, Hu, and Feng 2020)	✓	94.0	90.1	74.7	98.4	74.3	99.9	88.6
MA (Li et al. 2020)	✓	92.7	93.7	75.3	98.5	**77.8**	99.8	89.6
VDM-DA (Tian et al. 2021)	✓	93.2	94.1	75.8	98.0	77.1	**100.0**	89.7
CPGA (Qiu et al. 2021)	✓	**94.4**	94.1	76.0	98.4	76.6	99.8	**89.9**
D-MCD (ours)	✓	94.1	93.5	**76.4**	98.8	76.4	**100.0**	**89.9**

Table 3: Classification accuracy (%) on the Office-Home dataset (ResNet-50).✓ indicates the SFUDA method, and ✗ indicates the UDA method. Bold text indicates the best results.

Model Adaptation	Matching Distribution	Strong-Weak Model	Acc.(%)
	✓	✓	74.2
✓	✓		83.9
✓		✓	86.1
✓	✓	✓	**87.5**

Table 4: Ablation study results on VISDA dataset.

Experimental Results

We list the classification accuracy results of the proposed D-MCD method on the VISDA, Office-Home, and Office31 datasets in Table 1, 2, 3 respectively. The experimental results show that the classification accuracy of our method is higher than the current state-of-the-art SFUDA approaches (Qiu et al. 2021; Liang, Hu, and Feng 2020; Li et al. 2020) on the three benchmark datasets. Taking the results on the VISDA dataset as an example, we can observe that our D-MCD method improves the ResNet-101 model by 30.1% in terms of accuracy. Furthermore, our method achieves 87.5% accuracy, which outperforming the SSNLL (Chen et al. 2021) by a notable margin of 1.7%. This demonstrates that our method can effectively address the sample selection bias by reducing the distribution mismatch between high-confidence and low-confidence samples, while also eliminating the label noise in the high-confidence sample by applying the strong-weak paradigm. In addition, our method also improves the domain adaptation accuracy compared with traditional UDA methods (*i.e.*, 83.4% *v.s.* 87.5%). A similar observation can be made for the results on Office-Home and Office31 can be found.

Ablation Study

We conduct our ablation study by isolating each key part of our D-MCD method: *i.e.*, model adaptation, matching distribution, and strong-weak paradigm. The results are summarized in Table 4. We can observe from the table that each component of D-MCD contributes to the promotion of model performance on the target domain. More specifically, after removing the model adaptation component, the performance decreases dramatically to 74.2%. This means that a good enough initial target model is an essential part of the self-training strategy and will significantly improve the accuracy on the target domain. Moreover, removing the matching distribution component also results in a decline in accuracy to 86.1%, which reveals that sample selection bias is a primary obstacle to the self-training strategy. Furthermore, when employing the strong-weak paradigm, the accuracy on the target domain is improved from 83.9% to 87.5% , showing that the strong-weak paradigm can effectively denoise the pseudo-label, and thus further promote the their quality.

Qualitative Results

We visualize the output of the source domain model, the model after model adaptation, and the model after model self-training with method (Van der Maaten and Hinton 2008), as shown in Supplementary. We first conduct model adaptation, each category presents a tighter cluster but is still inevitably injected with some label noise. After the model self-training, in which we reduce the label noise, the clusters are tighter and cleaner.

Conclusion

In this paper, we address the SFUDA problem from the perspective of self-training and determine that the self-training strategy for SFUDA typically suffers from sample selection bias and the label noise of the pseudo-labeled samples. We go on to conduct a rigorous theoretical analysis of how these two risks affect the model generalization ability on the target domain. Based on the theoretical analysis, we then propose a novel Denoised Maximum Classifier Discrepancy (D-MCD) approach for the SFUDA problem. Specifically, we first minimize the distribution mismatch between high-confidence samples and the remaining target domain samples to alleviate the sample selection bias.Subsequently, we devise a strong-weak self-raining paradigm to reduce the label noise in the high-confidence samples. Benefiting from our proposed D-MCD, we achieve state-of-the-art results on three domain adaptation benchmark datasets, which demonstrates the effectiveness of our proposed approach.

Acknowledgements

This work is supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, the National Natural Science Foundation of China (Grant No. 62176047), Beijing Natural Science Foundation (Z190023), and Sichuan Science and Technology Program, NO: 2021YFS0374.

References

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A. C.; Bengio, Y.; and Lacoste-Julien, S. 2017. A Closer Look at Memorization in Deep Networks. In *ICML*, 233–242.

Chen, W.; Lin, L.; Yang, S.; Xie, D.; Pu, S.; Zhuang, Y.; and Ren, W. 2021. Self-Supervised Noisy Label Learning for Source-Free Unsupervised Domain Adaptation. *CoRR*, abs/2102.11614.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 248–255.

Deng, J.; Li, W.; Chen, Y.; and Duan, L. 2021. Unbiased Mean Teacher for Cross-Domain Object Detection. In *CVPR*, 4091–4101.

Dong, J.; Cong, Y.; Sun, G.; Fang, Z.; and Ding, Z. 2021. Where and How to Transfer: Knowledge Aggregation-Induced Transferability Perception for Unsupervised Domain Adaptation. *TPAMI*, 1–1.

Dong, J.; Cong, Y.; Sun, G.; Zhong, B.; and Xu, X. 2020. What Can Be Transferred: Unsupervised Domain Adaptation for Endoscopic Lesions Segmentation. In *CVPR*, 4022–4031.

French, G.; Mackiewicz, M.; and Fisher, M. H. 2018. Self-ensembling for Visual Domain Adaptation. In *ICLR*.

Ganin, Y.; and Lempitsky, V. S. 2015. Unsupervised Domain Adaptation by Backpropagation. In *ICML*, 1180–1189.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. J. 2012. A Kernel Two-Sample Test. *MLJ*, 723–773.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I. W.; and Sugiyama, M. 2018. Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. In *NeurIPS*, 8536–8546.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Lee, C.; Batra, T.; Baig, M. H.; and Ulbricht, D. 2019. Sliced Wasserstein Discrepancy for Unsupervised Domain Adaptation. In *CVPR*, 10285–10295.

Li, R.; Jiao, Q.; Cao, W.; Wong, H.; and Wu, S. 2020. Model Adaptation: Unsupervised Domain Adaptation Without Source Data. In *CVPR*, 9638–9647.

Li, S.; Lv, F.; Xie, B.; Liu, C. H.; Liang, J.; and Qin, C. 2021. Bi-Classifier Determinacy Maximization for Unsupervised Domain Adaptation. In *AAAI*, 8455–8464.

Liang, J.; Hu, D.; and Feng, J. 2020. Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation. In *ICML*, 6028–6039.

Liu, Y.; Deng, J.; Gao, X.; Li, W.; and Duan, L. 2021. BAPA-Net: Boundary Adaptation and Prototype Alignment for Cross-Domain Semantic Segmentation. In *ICCV*, 8801–8811.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning Transferable Features with Deep Adaptation Networks. In *ICML*, 97–105.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional Adversarial Domain Adaptation. In *NeurIPS*, 1647–1657.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2016. Unsupervised Domain Adaptation with Residual Transfer Networks. In *NeurIPS*, 136–144.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *ICML*, 2208–2217.

Lu, Z.; Yang, Y.; Zhu, X.; Liu, C.; Song, Y.; and Xiang, T. 2020. Stochastic Classifiers for Unsupervised Domain Adaptation. In *CVPR*, 9108–9117.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? In *NeurIPS*, 4696–4705.

Pan, F.; Shin, I.; Rameau, F.; Lee, S.; and Kweon, I. S. 2020. Unsupervised Intra-Domain Adaptation for Semantic Segmentation Through Self-Supervision. In *CVPR*, 3763–3772.

Peng, X.; Usman, B.; Kaushik, N.; Hoffman, J.; Wang, D.; and Saenko, K. 2017. Visda: The Visual Domain Adaptation Challenge. *CoRR*, abs/1710.06924.

Qiu, Z.; Zhang, Y.; Lin, H.; Niu, S.; Liu, Y.; Du, Q.; and Tan, M. 2021. Source-free Domain Adaptation via Avatar Prototype Generation and Adaptation. In *IJCAI*, 2921–2927.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting Visual Category Models to New Domains. In *ECCV*, 213–226.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In *CVPR*, 3723–3732.

Tian, J.; Zhang, J.; Li, W.; and Xu, D. 2021. VDM-DA: Virtual Domain Modeling for Source Data-free Domain Adaptation. *TCSVT*, 1–1.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep Domain Confusion: Maximizing for Domain Invariance. *CoRR*, abs/1412.3474.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing Data Using T-SNE. *JMLR*, 9(11).

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. In *CVPR*, 5018–5027.

Wang, Q.; and Breckon, T. P. 2020. Unsupervised Domain Adaptation via Structured Prediction Based Selective Pseudo-Labeling. In *AAAI*, 6243–6250.

Wang, Q.; Li, W.; and Gool, L. V. 2019. Semi-supervised Learning by Augmented Distribution Alignment. In *ICCV*, 1466–1475.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric Cross Entropy for Robust Learning With Noisy Labels. In *ICCV*, 322–330.

Yang, S.; Wang, Y.; van de Weijer, J.; Herranz, L.; and Jui, S. 2020. Unsupervised Domain Adaptation without Source Data by Casting a Bait. *CoRR*, abs/2102.11614(2): 3.

Yang, S.; Wang, Y.; van de Weijer, J.; Herranz, L.; and Jui, S. 2021. Generalized Source-Free Domain Adaptation. In *ICCV*, 8978–8987.

Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I. W.; and Sugiyama, M. 2019. How does Disagreement Help Generalization Against Label Corruption? In *ICML*, 7164–7173.