Denoised Maximum Classifier Discrepancy for Source-Free Unsupervised Domain Adaptation

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

Data Intelligence Group, University of Electronic Science and Techology 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, *i.e.*, iteratively selecting confidently predicted target samples as pseudo-labeled samples to train the model to fit the target domain. However, the self-training strategy may also suffer from the *sample selection bias* and 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 self-training strategy for the SFUDA problem. Benefiting from the theoretical analysis, we then propose a new Denoised Maximum Classifier Discrepancy (D-MCD) for SFUDA to effectively address these two issues. In particular, we first minimize the distribution mismatch between the selected pseudo-labeled samples and the rest target domain samples to alleviate the sample selection bias. Besides, 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 pseudo-label 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.

Introduction

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

While many methods have been proposed to solve the UDA problem (Tzeng et al. 2014; Long et al. 2017; Gretton et al. 2012; Ghifary, Kleijn, and Zhang 2014), it 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 the visual recognition tasks of medical images, surveillance videos, or fingerprint images, accessing these data often brings privacy issues.

To avoid accessing the source domain data in the domain adaptation process, a more challenging UDA problem is proposed named Source-Free Unsupervised Domain Adaptation (SFUDA) (Liang, Hu, and Feng 2020). 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.

A straightforward way to address SFUDA is applying the self-training strategy, and many methods were proposed along this line (Liang, Hu, and Feng 2020; Chen et al. 2021; Lao, Jiang, and Havaei 2021; Tian et al. 2021). The main idea is to use 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 iterated such that the model can be gradually improved.

However, the self-training strategy also has risks. First, there exists a sample selection bias when selecting pseudo-labeled samples from the target domain. This inevitably limits the model generalization ability on the entire target domain. Second, the pseudo-labeled samples often contain considerable label noises, which also harms the performance of the model. While a few heuristics were proposed (Liang, Hu, and Feng 2020; Chen et al. 2021; Lao, Jiang, and Havaei 2021; Tian et al. 2021) to improve the label quality, these two issues have not been clearly analyzed in those works.

In this work, we provide a rigorous theoretical analysis on how the sample selection bias and the label noise of pseudo-labeled samples affect the model generalization ability of the target model when applying the self-training strategy 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 training error with the pseudo-labeled samples, the label noise of the pseudo-labeled samples, the distribution mismatch between the selected pseudo-labeled samples and

*Corresponding author.

the rest of target samples, and other constant terms. This validates our analysis on 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). In particular, we first remold the Bi-Classifier Determinacy Maximization (BCDM) to adapt the pre-trained source model to the target domain using unlabeled target domain samples to obtain a good enough initial target model for self-training. Then, we start the self-training process in which we explicitly consider the before mentioned two risks. On one hand, 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 rest 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.

On the other hand, 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. The motivation behind this is that a model tends to remember easy samples during the early stage of the training process. So the newly trained weak model is able to identify hard examples for the strong initial target model to avoid producing high-confidence and incorrect predictions. We implement this by gradual ensembling the model parameters of the weak model to the strong model until the weak model is trained strong enough.

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 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 rest target domain samples, and improve the label quality of pseudo-labeled samples with 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 source and target domain in the feature space and relies 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; Ghifary, Kleijn, and Zhang 2014) or adversarial training through the domain discriminator to learn domain invariant features (Ganin and Lempitsky 2015; Long et al. 2018). In addition, there is a special kind of adversarial method that does not depend on the domain discriminator

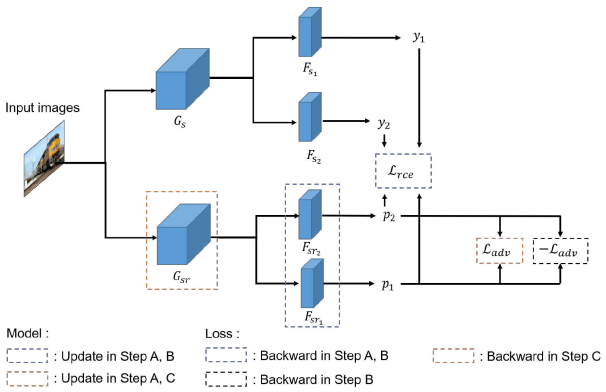


Figure 1: The overview of model adaptation. There are three steps (*i.e.*, step A, B, C) during the model adaptation, different colors dotted box indicates different backward or update in the corresponding step. Best viewed in colors.

and utilizes 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, to estimate the difference between the source domain and the target domain without using the source domain data. In addition to the method of inter-domain alignment, some methods consider intra-domain alignment (Pan et al. 2020) or fit target distribution straightforwardly (Wang and Breckon 2020; Deng et al. 2021; Liu et al. 2021). They often depends on the accuracy of prototype estimation and the accuracy of pseudo-label annotation.

Source-Free Unsupervised Domain Adaptation The 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 of the model 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; Lao, Jiang, and Havaei 2021) exploits pseudo label prediction from the source model or prototype to adapt the model to the target domain such that making the model fit the target domain distribution well.

Noise Label Learning Noise label learning refers to reducing the influence of noise labels and improving the performance of the model under the dataset label noise. The regularization method is to add a regular term in the training loss to avoid the sample overfitting the noise label (Wang et al. 2019; Müller, Kornblith, and Hinton 2019). Previous work (Wang, Li, and Gool 2019) has shown that deep networks can memorize easy samples first, and gradually memorize hard samples during the training process. Based on this observation, some approaches (Han et al. 2018; Yu et al. 2019) based on filtering labels to reduce the accumulation of errors have also achieved a good result.

Revisit 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 $\mathcal{D}$ and $\hat{\mathcal{D}} = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$ 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 access to the original labeled source domain samples.

When applying the self-training strategy for solving 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 the prediction of a certain model (*e.g.*, the pretrained source model or the target model from the previous training stage). For ease of presentation, we redefine the high-confidence set as $\hat{\mathcal{D}}_h = \{(\mathbf{x}_i^h, y_i)\}_{i=1}^{n_h}$ where the y_i corresponding to pseudo label of x_i^h.

At the first glance, when applying the self-training strategy for solving the SFUDA problem, we are facing a semi-supervised learning problem, 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 has more 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, making the model trained with these selected pseudo-labeled samples cannot well generalize to the entire target domain. Moreover, as the labels of the pseudo-labeled samples are obtained from model predictions instead of human annotation, there is often considerable noise in these labels, which may also degrade the performance of the target model.

To verify our above analysis, we derive a generalization error bound for the SFUDA problem. In particular, following the terminology in (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 also 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 $\mathcal{E}_{\hat{\mathcal{D}}}(h)$, then 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 $1 - 3\delta$,*

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

where $d_{h,\mathcal{H}}(\hat{\mathcal{D}}_h, \hat{\mathcal{D}}_l)$ is the distribution mismatch between the selected pseudo-labeled samples and the rest 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. From the generalization, we can observe that, despite 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 rest 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), and 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 Denoisd Maximum Classifier Discrepancy (D-MCD), in which we improve the self-training strategy by additionally reducing the noise in the pseudo-labels of the confidence samples and the distribution difference between the confidence and non-confidence samples.

Specifically, we build up our D-MCD approach on the improved MCD(Saito et al. 2018) method BCDM (Li et al. 2021). As self-training usually requires a strong enough initial model which performs sufficiently well on the target domain, we first adopt the pretrained source model to the target domain using unlabeled target samples, referred to as the *Model Adaptation* phase. Then, we start the self-training and simultaneously address the label noise and sample selection bias issues, referred to the *Model Self-Training* phase.

Model Adaptation

The BCDM (Li et al. 2021) method was proposed for the traditional unsupervised domain adaptation problem, where the labeled source domain samples are available during the training process. The generalization bound satisfied:

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

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

It is assumed that the model performs well in the source domain, so $\mathcal{E}_{\mathcal{S}}(h)$ and $\mathrm{dis}_{\mathcal{S}}\left(h', h\right)$ are small. To minimize $\mathrm{dis}_{\mathcal{T}}\left(h', h\right)$, assuming that hypothesis $h = f_1 \circ g$ and $h' = f_2 \circ g$ and replace sup with max and for any hypothesis and the above inequality still holds for any feature extractor g. The objective function can be rewritten as follows:

$$\min_{g} \max_{f_1, f_2} \mathrm{dis}_{\mathcal{T}}\left(f_1 \circ g, f_2 \circ g\right) \tag{3}$$

The above bounds can be optimized through the adversarial training between the classifiers f and feature extractor g.

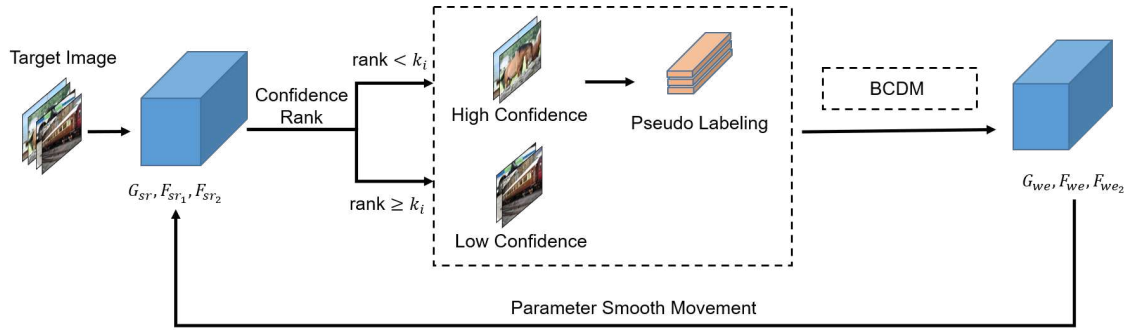


Figure 2: The pipeline of the strong-weak self-training paradigm. The target images are first fed into the strong model (*i.e.*, G_{sr}, F_{sr_1} and F_{sr_2}) to divide the target domain into high-confidence and low-confidence samples split. These two sets of samples are used for training the weak model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

Therefore, the training process in the BCDM method can be summarized as the following three steps:

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

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

Step B The CDD distance (Li et al. 2021) adopted as $d(.,.)$ to measure the classifier output discrepancy. Follow Eq. 3 , to maximize the CDD distance with classifier and train with ℓ_1, ℓ_2 to keep stable.

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

Step C Follow Eq. 3, minimize the CDD distance with feature extractor G.

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

Remold BCDM for SFUDA In the SFUDA problem, the labeled samples in the source domain are not accessible during the training problem, the BCDM cannot be directly applied to the SFUDA problem, since the loss ℓ_1 and ℓ_2 in Step A and B cannot be optimized due to the lack of 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 the error performance of the model on the source domain is also extremely small, so 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 Step A and B. Training only with step B, C without ℓ_1, ℓ_2 may cause the model's error on the source domain to increase. So that to remold BCDM for SFUDA, we cannot only train with adversarial training but also needs to maintain the performance of the model in the source domain.

In particular, assuming we are given a pre-trained source domain M_s, which includes two branches of classification heads F_{s_1}, F_{s_2}, and a common feature extractor G_s and we init model $F_{sr_1}, F_{sr_2}, G_{sr}$ with source model. To cope with this problem, we propose to replace these loss functions with reverse cross entropy loss (RCE loss function):

$$\begin{aligned} \ell_{rce_1} &= -\sum_{k=1}^{K} p_1(k|\mathbf{x}_i^t) \log q_1(k|\mathbf{x}_i^t) \\ \ell_{rce_2} &= -\sum_{k=1}^{K} p_2(k|\mathbf{x}_i^t) \log q_2(k|\mathbf{x}_i^t) \end{aligned} \tag{4}$$

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

With the soft-labels from the pre-trained model M_s, the traditional cross-entropy loss can also be used as an alternative to the above reverse cross entropy loss because it can be used as a regular term to keep the model from collapsing during training. However, compared with the cross entropy loss function, the reverse cross entropy 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 address confidence, as discussed below:

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

$$\ell_{rce} = -\sum_{k=1}^{K} p(k|\mathbf{x}) \log q(k|\mathbf{x}) \tag{5}$$

The gradient of the RCE loss function to the j-th element z_j output by the neural network is:

$$\frac{\partial \ell_{rce}}{\partial z_j} = p_j(\sum_{k=1}^{K} p_k \log q_k - \log q_j) \tag{6}$$

Fixing the probability p and analyzing the influence of q on the gradient, we find that when the predicted probability

q is a uniform probability vector such as $[\frac{1}{K}, \frac{1}{K}, ..., \frac{1}{K}]$, the gradient of the RCE loss function is:

$$\frac{\partial \ell_{rce}}{\partial z_j} = p_j(\sum_{k=1}^{K} p_k \log \frac{1}{K} - \log \frac{1}{K}) = 0 \tag{7}$$

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

$$\frac{\partial \ell_{rce}}{\partial z_j} = p_j - q_j \tag{8}$$

It shows that the gradient of the cross-entropy loss function only focuses on consistency and ignores confidence. Therefore, by optimizing the RCE loss function in step A, the model not only can learn according to the confidence of the soft label but also is able to maintain its performance on the high-confidence samples.

Model Self-Training

After obtaining a sufficiently good initial model, we now discuss how to improve the self-training process. In particular, when training the target model with selected pseudo-labeled high-confidence samples, we design a strong-weak self-training paradigm where 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 samples and the low-confidence samples. The overview of model self-training is shown in Fig. 2.

Strong-Weak Self-Training Paradigm We treat the initial model as the strong model. In particular, 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 usually outputs confident predictions for the target domain samples, even the predictions might be wrong. In other words, there are some high-confidence but incorrect labels in the predicted results. From 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 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 the source or the target domain, *e.g.*, an ImageNet pre-trained model. Our motivation is based on an observation that deep networks can memorize label correct samples first (Arpit et al. 2017) and gradually memorize label noise samples during the training process. Therefore, the weak model tends to first remember the high-confidence and correct label during the training process, and ignore the high-confidence but incorrect label. Therefore, we use the weak model to help the strong model to filter out these high-confidence but incorrectly predicted samples. Specifically, during the training process, we use the method of smooth parameter movement to fuse the parameters of the weak model with the parameters of the strong model at the end of each epoch.

$$\theta_{sr} = \alpha\theta_{sr} + (1-\alpha)\theta_{we} \tag{9}$$

In this way, the addition of parameters from the weak model helps 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, so that these samples 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 continues to become stronger, and even the predictive accuracy of high-confidence samples exceeds the original strong model. So 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 label of a fixed proportion of high-confidence samples before and after denoising on several datasets, we confirm the effectiveness of our method (see Supplementary for the details).

The 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, use the strong model mentioned in the previous section to assign pseudo-labels to high-confidence samples. Therefore, we can use the traditional UDA method BCDM (Li et al. 2021) to align the labeled source domain with the unlabeled target domain. We refer to the steps A, B, C mentioned in the model adaptation chapter and iteratively perform the following steps:

Step 1 Compared with the step A mentioned earlier, 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 Compared with step B mentioned earlier, 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. And we calculate CDD distance with samples $\mathbf{x}^l$ of low-confidence domain instead of the sample of target domain.

Step 3 Compared with step C mentioned earlier, we calculate CDD distance with samples $\mathbf{x}^l$ of low-confidence domain instead of the sample of target domain.

In general, we just replaced the source domain with a high-confidence sample with pseudo-labels and replaced the target domain with an unlabeled sample.

Details of Selecting High-confidence Samples After model adaptation, we obtain a strong enough model $G_{sr}, F_{sr_1}, F_{sr_2}$ for better predicting result on the target domain, so we use this strong model to split the target domain. For the index of separating samples, we select the CDD (Classifier Determinacy Disparity) (Li et al. 2021) distance as the measure of the confidence level of the sample. The CDD distance can measure the consistency between the output of the classifier and the confidence of the output. In other words, CDD distance equals 0 only if the output of each classifier is one-hot and consistent.

Given a ranking of scores from CDD distance for each sample in the target domain, hyper-parameter r is introduced as a ratio to class-wise separate the target images into a

Table 1: Classification accuracy (%) on **VISDA** dataset (ResNet-101). ✓indicates the SFUDA method, and ✗indicates the UDA method; The bold result in the table represents the best result.

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 2: Classification accuracy (%) on **Office-Home** dataset (ResNet-50). In the Source-Free part of the figure, ✓indicates the SFUDA method, and ✗indicates the UDA method; The bold result in the table represents the best result.

Method	Source-Free	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 (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 3: Classification accuracy (%) on **Office31** Dataset (ResNet-50). ✓indicates the SFUDA method, and ✗indicates the UDA method; The bold result in the table represents the best result.

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 (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**

high-confidence and low-confidence domain. 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 samples in the target domain as n_t, the number of categories as K, so 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. This interval describes the range of samples selected for each category. We define the number of samples selected for the i-th category as k_i and the number of samples selected for the i-th category $k_i = \min(b_i, \max(k_i, a_i))$. This operation helps to ensure the balanced number of samples of each category in the constructed high-confidence domain.

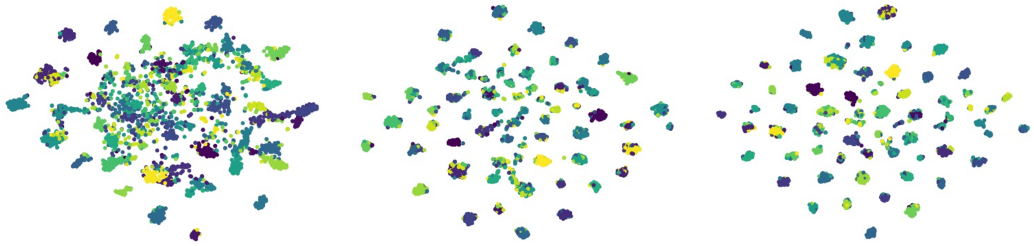
Experiments

Experimental Setup

Dataset We evaluate our method on three widely used UDA benchmark datasets: 1) VISDA (Peng et al. 2017) is a large-scale challenging dataset with 12 classes. 2) Office-Home (Venkateswara et al. 2017) is a medium-sized image classification dataset. There are four distinctive domains: Art, Clipart, Product, RealWorld. 3) Office31 (Saenko et al. 2010) is a small-sized image classification dataset. This dataset includes three different domains: Amazon (A), DSLR (D), and Webcam (W).

Experiment detail We first train a model using the labeled source domain samples and 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 absent. Following SSNLL (Chen et al. 2021), the data transform method for high-confidence samples adopts from (French, Mackiewicz, and Fisher 2018) and for different transform sample output to keep consistency. For office31 dataset, we calculate probability using the distance to the prototype of each class instead of the classifier, we generate the prototype following SHOT (Liang, Hu, and Feng 2020), besides, to balance the model obtained by Model Adaptation training and Model Self-training, the ensemble output of these two model will be used. More experiment detail will be shown in supplementary.

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



(a) Source Model (b) After Model Adaptation (c) After Model Self-Training

Figure 3: The qualitative results for different stage model on the Office-Home dataset Ar→Cl.

Network Hyper-parameters We set hyper-parameters for RCE loss $\beta = 0.1$ for Office dataset and $\beta = 0.01$ for VISDA dataset, $\gamma = 0.0025$ in training step B and C, $r = 0.4$ for VISDA and Office-home dataset and $r = 0.6$ for Office31 dataset. We adopt Stochastic Gradient Descent optimizer (SGD) with momentum 0.9 and weight decay 5×10^{-4} and 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 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 Office-Home and Office31 dataset, learning rate of the feature extractor is 3×10^{-3} and learning rate of the feature classifier is 1×10^{-2}. Besides, we exploit the entropy loss as (Saito et al. 2018; Long et al. 2016) to make the training procedure more stable in all experiment and we set hyper-parameters of this loss $\sigma = 0.05$.

Table 4: Ablation study results on **VISDA** dataset.

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

Experimental Results

We show the classification accuracy 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 (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 baseline source only model by 30.1% in terms of accuracy. Besides, our method achieves 87.5% accuracy, which outperforms the CPGA (Qiu et al. 2021) by a notable margin of 1.5%. This demonstrates that our method can effectively address the sample selection bias by reducing the distribution mismatch between high-confidence and low-confidence samples and eliminating the label noise in the high-confidence sample by applying the strong-weak paradigm. Furthermore, our method also improve the domain adaptation accuracy compared with traditional UDA methods (*i.e.*, 83.4% *v.s.* 87.5%). A similar observation on the results of 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 that each component of D-MCD contributes to the promotion of model performance on the target domain. Specifically, after removing model adaptation, 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 of the target domain. Besides, removing the matching distribution will also hurt the accuracy to 86.1%, showing that the sample selection bias is a main obstruct to the self-training strategy. Moreover, when employing the strong-weak paradigm, the accuracy on the target domain is improved from 83.9% to 87.5% as the strong-weak paradigm can effectively denoise the pseudo-label so that further promote the quality of the pseudo-label.

Qualitative Results

We visualize the output probability vector of the source domain model, the model after model adaptation, and the model after model self-training shown in Fig. 3. We first conduct model adaptation (Fig. 3), each category will present a more tight cluster but still is inevitably injected some label noise. After model self-training where we address the sample selection bias and reduce the label noise so that the cluster is more tight and clean.

Conclusion

In this paper, we address the SFUDA problem from the perspective of self-training and reveal that the self-training strategy for SFUDA usually suffers from sample selection bias and the label noise of the pseudo-labeled samples. We provide a rigorous theoretical analysis on how these two risks affect the model generalization ability on the target domain. Based on the theoretical analysis, we propose a novel Denoised Maximum Classifier Discrepancy (D-MCD), for the SFUDA problem. Specifically, we first minimize the distribution mismatch between high-confidence samples and the rest target domain samples to alleviate the sample selection bias. And then 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.

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 2021*, 4091–4101.

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.

Ghifary, M.; Kleijn, W. B.; and Zhang, M. 2014. Domain Adaptive Neural Networks for Object Recognition. In *PRICAI*, 898–904.

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.

Lao, Q.; Jiang, X.; and Havaei, M. 2021. Hypothesis Disparity Regularized Mutual Information Maximization. In *AAAI*, 8243–8251.

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 *The IEEE International Conference on Computer Vision (ICCV)*.

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 Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *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. .

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 Zhou, Z., ed., *IJCAI*, 2921–2927.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *ECCV*, 213–226. Springer.

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. *CoRR*, abs/2103.14357.

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

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. *CoRR*, abs/1905.08171.

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. 2021. Generalized Source-free Domain Adaptation. *CoRR*, abs/2108.01614.

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.