Seek Common Ground While Reserving Differences: A Model-Agnostic Module for Noisy Domain Adaptation

Yukun Zuo, Hantao Yao, *Member, IEEE*, Liansheng Zhuang, *Member, IEEE*, and Changsheng Xu, *Fellow, IEEE*

***Abstract*—Noisy domain adaptation aims to solve the problem that the source dataset contains noisy labels in domain adaptation. Previous methods handle noisy labels by selecting the small-loss samples with inconsistent predictions between two models and discarding the consistent samples, resulting in many noises contained in the selected samples. By jointly considering the consistent and inconsistent samples, we propose a model-agnostic module, named Seek Common Ground While Reserving Differences (SCGWRD), to reduce the impact of noisy samples. The proposed SCGWRD module consists of Seek Common Ground (SCG) component and Reserve Differences (RD) component by utilizing the outputs of two symmetrical domain adaptation models. As the common samples with consistent predictions between two models are more likely to be clean samples, the SCG component applies the small-loss strategy to select the reliable samples with consistent predictions. Unlike SCG, the RD component maintains the divergences between two models with mutual learning and reduces the effect of noisy data using the samples with different predictions and small losses. Evaluations on three benchmarks demonstrate the effectiveness and robustness of the proposed SCGWRD module for noisy domain adaptation.**

***Index Terms*—Noisy domain adaptation, Seek common ground component, Reserve differences component.**

I. Introduction

UNSUPERVISED domain adaptation, which aims to transfer knowledge learned from a label-rich source domain to a label-scarce target domain, has attracted more and more attention. A lot of methods [1]–[8] have been proposed to reduce the domain gap between the source domain and target domain. However, they all assume the datasets are clean with accurate annotations, leading to that they are not suitable for domain adaptation in real-world scenarios containing many noisy labels. Therefore, solving unsupervised domain adaptation under noisy environments is the key to improve its generalization.

A novel task named *Noisy Domain Adaptation* [9], [10] is introduced to address unsupervised domain adaptation under noisy environments, which assumes that the source domain contains label noise.[1] As label noise has adverse effects for domain alignment, noisy domain adaptation is a more challenging task than standard domain adaptation. The core of noisy domain adaptation is to filter out noisy labels and use clean images for domain alignment. The small-loss approaches [11]–[14] assumes that the samples whose supervised loss is lower than a threshold can be treated as clean samples. For example, a promising approach [12] utilizes the small-loss strategy to select clean samples based on the outputs of two symmetrical models. However, two symmetrical models are easy to converge into a consensus. Furthermore, some methods [13], [14] utilize the small-loss samples with different predictions to constrain two models to be diverged. However, the samples with inconsistent predictions always contain many noisy samples, *i.e.,* 75% of the samples selected by Co-teaching+ [13] are noisy, which deteriorate domain alignment. The reason is that these methods discard the samples with consistent predictions and only consider inconsistent ones, as shown in Fig. 1. The samples with consistent predictions are more likely to be clean samples, which complement the inconsistent samples. Therefore, jointly considering consistent and inconsistent small-loss samples can make the selected samples contain few noises and boost domain alignment for noisy domain adaptation.

To address the above issues, we propose a model-agnostic module named Seek Common Ground While Reserving Difference (SCGWRD), which consists of Seek Common Ground (SCG) component and Reserve Differences(RD) component.

Manuscript received February 24, 2021; revised June 4, 2021; accepted July 4, 2021. Date of publication July 16, 2021; date of current version March 4, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102205, in part by the National Natural Science Foundation of China under Grants 61902399, 61721004, U1836220, U1705262, 61832002, 61720106006, U20B2070, 61976199, and 62036012, in part by Beijing Natural Science Foundation under Grant L201001, and in part by the Key Research Program of Frontier Sciences, CAS under Grant QYZDJ-SSW-JSC039. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yazhou Yao. *(Corresponding author: Changsheng Xu.)*

Yukun Zuo and Liansheng Zhuang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: zykpy@mail.ustc.edu.cn; lszhuang@ustc.edu.cn).

Hantao Yao is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: hantao.yao@nlpr.ia.ac.cn).

Changsheng Xu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: csxu@nlpr.ia.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3097495.

Digital Object Identifier 10.1109/TMM.2021.3097495

[1]In this work, we only focus on label noise. The image noise, which refers to low-quality pixels of images, is the other type of noise in noisy domain adaptation.

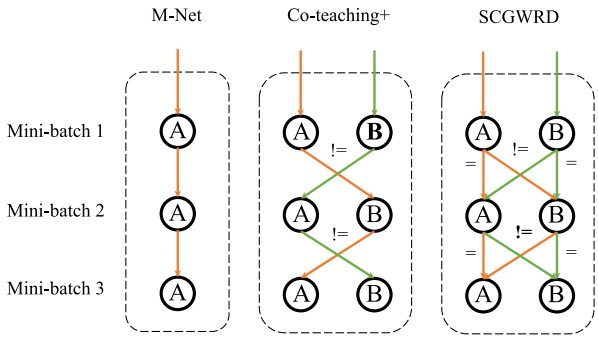


Fig. 1. Comparison of MentorNet (M-Net), Co-teaching+ and SCGWRD. Orange arrows and green arrows represent the error flows from model A and B, respectively. M-Net only maintains one model A. Co-teaching only uses the different samples (!=) which have inconsistent predictions between two models. The SCGWRD jointly considers the different samples (!=) and common samples (=) between two models.

The SCG component assumes the samples with consistent predictions between two symmetrical models are more likely to be clean samples. However, these samples still contain some noises. Therefore, the *small-loss strategy* is applied to remove the noisy samples. For the SCG component, two symmetrical domain adaptation (DA) models are easily to converge into a consensus if only using the consistent samples. Furthermore, the RD component is proposed to avoid two models being converged by picking the inconsistent samples with small losses. Once the RD component enhances the divergence between two symmetrical models, it can help the SCG component select reliable samples. By combining these two components, our proposed module can effectively pick the clean samples and reduce the effects of the noisy samples.

Concretely, given noisy source samples, we first utilize two symmetrical domain adaptation models to extract the corresponding predictions of each sample. Since the proposed module only relies on the predictions of domain adaptation model, it is a model-agnostic module that can be plugged and played with any domain adaptation model. Based on the predictions, the whole samples can be classified into two groups: common samples and difference samples, which denote the samples having consistent and inconsistent predictions between two models, respectively. In the SCG component, each model selects the reliable samples from the common samples with the small-loss strategy to back propagate and update its parameters for self-training. Besides the common samples, each model picks small-loss samples from different samples of its peer model to back propagate and update its parameters in the RD component. Moreover, the pseudo-labels obtained for unlabeled target samples can be regarded as noisy labels, and the proposed SCGWRD is also applied for the unlabeled target samples.

The evaluations on three benchmarks demonstrate the superiority of the proposed module,*e.g.,* obtaining the performance of 86.8%, 59.3%, and 84.4% under 40% label corruption in Office-31, Office-Home, and Bing-Caltech datasets, respectively. The contributions can be summarized as follows:

- We propose a SCGWRD module to discover the reliable samples by considering the Seek Common Ground component and Reserving Differences component for noisy domain adaptation,
- By treating the pseudo-labels of target samples as noisy labels, the proposed SCGWRD module can also be applied to enhance the target representation learning, which provides a new perspective for standard domain adaptation.
- The proposed module is a model-agnostic module, which can be plugged and played with existing domain adaptation models, *e.g.,* obtains the improvement of 2.9%, 14.4%, and 2.5% under 40% label corruption in Office-31 for GVB [15], MCD [16], and CAN [17], respectively.

II. Related Work

Domain adaptation and noisy labels are two related research areas to noisy domain adaptation. We thus give a brief description of these related research areas.

Domain adaptation. Domain adaptation aims to transfer knowledge from a labeled source domain to an unlabeled target domain. Recently, many methods [3], [15], [18]–[23] have been proposed, which can be classified into three categories. Firstly, divergence-based methods [3], [5], [17], [18], [24] aim to minimize a divergence that measures the distribution distance between two domains to explicitly reduce domain discrepancy. Secondly, adversarial methods [2], [15], [19], [25]–[27] use the minimax games between a discriminator and a generator to learn domain-invariant features to achieve domain transfer. Thirdly, ensemble-based methods [20], [21], [28] use ensemble predictions to perform self-ensembling learning or use ensemble to measure the confidence of pseudo-labels in the target domain. Although the above methods have achieved good performance, they perform poorly under noisy environments, which is a more realistic scenario known as noisy domain adaptation. Compared with the standard domain adaptation, noisy domain adaptation is a more challenge problem.

Noisy labels. Recently, selecting small-loss samples is a widely used strategy to solve the noisy label problem [11]–[13]. Based on the fact that the samples with small supervised loss are more likely to be clean ones, small-loss samples can be used for training. For example, MentorNet [11] pre-trains a teacher model for selecting small-loss samples to train the student network. Nonetheless, MentorNet suffers from the accumulated error caused by noisy labels. Unlike MentorNet, Co-teaching [12] adopts two symmetrical models and uses the small-loss samples of each model to update its peer model. To maintain two models diverged, Co-teaching+ [13] selects the small-loss samples with inconsistent predictions between two models, and updates the parameters of each model with the small-loss samples from its peer model. However, the samples with different predictions contain many noises, which are harmful to representation learning. Moreover, the common samples with consistent predictions are more likely to be clean samples. Therefore, our work jointly considers the common and different samples to obtain more reliable samples for noisy domain adaptation.

Noisy domain adaptation. The critical of noisy domain adaptation is to discover the noisy samples and use reliable samples for domain alignment. For example, TCL [9] proposes the transferable curriculum learning approach to find noiseless and transferable source samples. RDA [10] proposes an offline curriculum learning to select clean source samples, and uses proxy distribution based on margin discrepancy to reduce the impact of feature noise. Because the above methods rely on adversarial learning, which is vulnerable to noisy labels, they are not robust for noisy domain adaptation. Different from them, we propose a model-agnostic module that only relies on the outputs of domain adaptation models and can be incorporated into any existing domain adaptation methods.

III. The Proposed Approach

Domain adaptation (DA) aims to transfer knowledge from a labeled source domain to an unlabeled target domain. Formally, we define the source and target datasets as $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and $\mathcal{D}_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$, where x, y, and n denotes the images, labels, and the number of images. Note that the source and target datasets have the same categories. Given the source dataset $\mathcal{D}_s$ along with the target images $\mathcal{X}_t = \{x_i^t\}_{i=1}^{n_t}$, existing DA methods [15], [18]–[20] aim to predict the label y^t for each target image x^t. However, these methods all assume that the whole labels in the source domain are clean, which is strict in real-world scenarios. The labels of source images always contain many noises due to errors in manual annotation, label polysemy, or the bias of a crowd-sourcing system [9].

To address the noisy label problem, a novel task named *Noisy Domain Adaptation* is introduced by assuming that the source domain contains label noise. Therefore, the source dataset in noisy domain adaptation can be redefined as $\hat{\mathcal{D}}_s = \{(x_i^s, \hat{y}_i^s)\}_{i=1}^{n_s}$, where $\hat{y}^s$ represents the source labels with noise. As $\hat{y}^s$ is unavailable, $\hat{y}^s$ is obtained by corrupting the source clean label y^s with a label transition matrix T, where $T_{ij} = p(\hat{y} = j | y = i)$ represents the probability of being flipped into a noise label j when the true label is i. Similar to previous methods [9], [10], we assume that the noise contained in the source labels is uniform noise, which denotes that the true labels are flipped into other labels with the same probability. Therefore, the label transition matrix T is defined as:

$$T_{ij} = \begin{cases} = 1 - \beta & i = j, \\ = \dfrac{\beta}{M-1} & i \neq j, \end{cases} \tag{1}$$

where $\beta \in [0, 1]$ is a parameter denoting the noise rate, and M is the number of categories.

The key of noisy domain adaptation is how to filter out noisy data and use reliable samples to align two domains. In this work, we propose a model-agnostic module named Seek Common Ground While Reserving Differences (SCGWRD), which consists of two components: Seek Common Ground (SCG) component and Reserve Differences (RD) component. Combined existing domain adaptation models with the SCGWRD module, the framework of noisy domain adaptation is presented in Fig. 2. As shown in Fig. 2, the Domain Alignment module, which can be any existing DA model, is used to obtain the predictions and align two domains. Given noisy source samples and unlabeled target samples, we firstly apply two symmetrical DA models $\mathbf{f}_1$ and $\mathbf{f}_2$ to generate the corresponding predictions. Based on the predictions, all samples can be classified into two groups: common samples and different samples, which denote the samples having consistent and inconsistent predictions between two models, respectively. In the SCG component, each model picks small-loss data from the common samples to back propagate itself and updates its parameters for self-training. In the RD component, the reliable samples are selected from the different samples using the small-loss strategy in each model to back propagate and update its peer model, maintaining the differences between two models and reducing the error from noisy labels by peer model mutually. In the following, we give detailed descriptions of the Domain Alignment module, Seek Common Ground component, and Reserve Differences component.

A. Domain Alignment Module

Domain Alignment module, a critical component for noisy domain adaptation, is used to align the source and target domains. Furthermore, the Domain Alignment module also needs to help filter out noisy samples. As our work aims to discover reliable samples, we adopt existing DA models as the Domain Alignment module and focus on how to filter out noisy samples, *e.g.*, CAN [17], GVB [15], and MCD [16] have been used for evaluations. Given the source and target samples, the constraint of the Domain Alignment module $\mathcal{L}_{da}$ is the objective function for the DA model,

$$\mathcal{L}_{da} = \mathcal{L}_{sup} + \mathcal{L}_{align}, \tag{2}$$

where $\mathcal{L}_{sup}$ denotes the supervised loss in the source domain, and $\mathcal{L}_{align}$ is the loss for aligning the distributions between two domains.

As the source labels contain many noises, using $\mathcal{L}_{sup}$ for the whole samples can seriously deteriorate the performance of domain adaptation. Therefore, filtering out noisy samples and using the selected clean samples for $\mathcal{L}_{sup}$ are reasonable for noisy domain adaptation. However, using one model is hard to select clean samples or reliable samples. Inspired by Co-teaching [12], we use two symmetrical DA models for sample selection, where each model is initialized independently, leading to that all the samples can be projected into two independent feature spaces. Consequently, the samples obtaining consistent prediction from two feature spaces have high confidence to be clean data.

Given noise source sample x_i^s, the predictions of two models are denoted as $\mathbf{f}_1(x_i^s)$ and $\mathbf{f}_2(x_i^s)$, respectively. Once obtaining the predictions, the corresponding pseudo-labels, which can be used to justify whether two predictions are consistent, can be obtained with Eq. (3),

$$\begin{aligned} \bar{y}_i^{s_1} &= \arg\max \mathbf{f}_1(x_i^s), \\ \bar{y}_i^{s_2} &= \arg\max \mathbf{f}_2(x_i^s). \end{aligned} \tag{3}$$

Similarly, the pseudo-labels for target samples can be obtained and denoted as $\bar{y}_i^{t_1}$ and $\bar{y}_i^{t_2}$. With the obtained pseudo-labels,

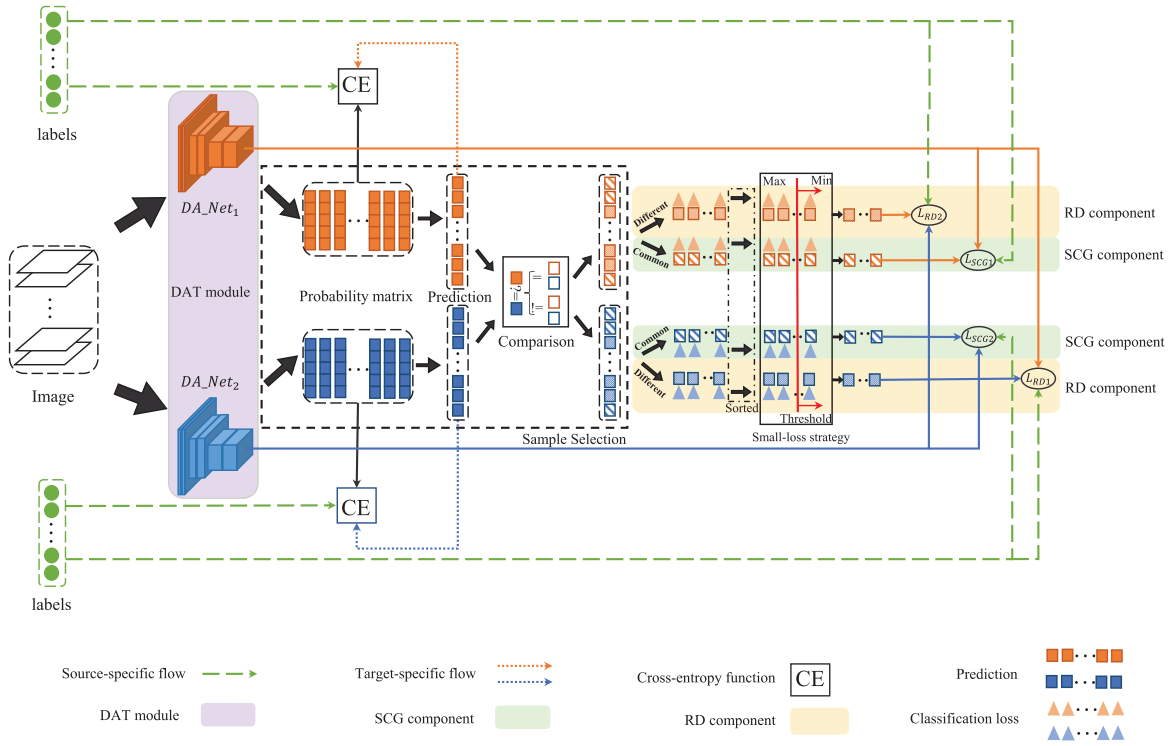


Fig. 2. An overview of our Seek Common Ground While Reserve Differences (SCGWRD) module combined with existing DA methods. Domain Alignment (DAT) module aims to align two domains and generate the prediction for each sample. Sample Selection obtains the common samples and different samples by comparing the predictions between two model. SCGWRD module consists of Seek Common Ground (SCG) component and Reserve Differences (RD) component. In the SCG component, each model selects small-loss data from the common samples to teach itself. In the RD component, each model utilizes small-loss data from the different samples to update the parameters of its peer model.

the source dataset and target dataset for noisy domain adaptation can be redefined as $\mathcal{D}_s = \{(x_i^s, \hat{y}_i^s, \bar{y}_i^{s_1}, \bar{y}_i^{s_2})\}_{i=1}^{n_s}$ and $\mathcal{D}_t = \{(x_i^t, \bar{y}_i^{t_1}, \bar{y}_i^{t_2})\}_{i=1}^{n_t}$, which are used to select reliable samples.

B. Seek Common Ground Component

Since the common samples are more likely to be clean data, they are essential for aligning the source and target domains. For the source and target domain, the corresponding common samples are defined as *source common samples* and *target common samples*. As the source and target samples have the same strategies of seeking the common samples, we treat the source common samples as an example to introduce the Seek Common Ground component.

Given source dataset $\mathcal{D}_s = \{(x_i^s, \hat{y}_i^s, \bar{y}_i^{s_1}, \bar{y}_i^{s_2})\}_{i=1}^{n_s}$, we first select the samples which have the consistent predictions between two DA models. The selection is conducted by considering whether the two pseudo-labels $\bar{y}_i^{s_1}$ and $\bar{y}_i^{s_2}$ are the same,

$$\mathcal{A}^s = \{x_i^s | \bar{y}_i^{s_1} = \bar{y}_i^{s_2}, i \in \mathbb{N}_+^{n_s}\}, \tag{4}$$

where $\mathbb{N}_+^{n_s} = \{1, 2, 3, \ldots, n_s\}$ and $\mathcal{A}^s$ denotes the selected source common samples.

However, $\mathcal{A}^s$ still contains many noises. Based on the fact that the neural networks tend to remember clean data first and then those of noisy data [29], the small-loss strategy is thus proposed to remove noisy samples. Specifically, clean data usually have a smaller loss than noisy data. Therefore, the small-loss strategy treats the samples with small classification loss as reliable samples. With the given corrupted source labels, the classification loss for each sample x_i^s in two models defined as $l_i^{s_1} = \mathcal{L}(\mathbf{f}_1(x_i^s), \hat{y}_i^s)$ and $l_i^{s_2} = \mathcal{L}(\mathbf{f}_2(x_i^s), \hat{y}_i^s)$, where $\mathcal{L}$ represents the cross-entropy loss, and $\hat{y}_i^s$ represents the given label. In each mini-batch data, each model selects $\tau\%$ of small-loss samples as reliable samples, *i.e.,* $\mathcal{R}^{s_1}$ and $\mathcal{R}^{s_2}$,

$$\begin{aligned} \mathcal{R}^{s_1} &= \{x_i^s | l_i^{s_1} \leqslant T(\mathbf{L}_{\mathcal{A}^s}^{s_1}, \tau), x_i^s \in \mathcal{A}^s\}, \\ \mathcal{R}^{s_2} &= \{x_i^s | l_i^{s_2} \leqslant T(\mathbf{L}_{\mathcal{A}^s}^{s_2}, \tau), x_i^s \in \mathcal{A}^s\}, \end{aligned} \tag{5}$$

where $T(\mathbf{L}_{\mathcal{A}^s}^{s_1}, \tau)$ is a function to determine the threshold that can select $\tau\%$ of small losses from the whole loss set $\mathbf{L}_{\mathcal{A}^s}^{s_1}$ of the common samples. τ is a parameter for the small-loss strategy.

Once obtaining reliable samples for the source domain, each model back propagates these samples and update its parameters for self-training. For noisy source sample, the given label $\hat{y}^s$ is used to compute the classification loss and update the corresponding model:

$$\begin{aligned} L_{SCG}^{s_1} &= -\mathop{\mathbb{E}}_{x \in \mathcal{R}^{s_1}} [\mathcal{L}(\mathbf{f}_1(x), \hat{y}^s)], \\ L_{SCG}^{s_2} &= -\mathop{\mathbb{E}}_{x \in \mathcal{R}^{s_2}} [\mathcal{L}(\mathbf{f}_2(x), \hat{y}^s)], \end{aligned} \tag{6}$$

where $L_{SCG}^{s_1}$ and $L_{SCG}^{s_2}$ denote the loss for models $\mathbf{f}_1$ and $\mathbf{f}_2$ in the Domain Alignment module, respectively.

Similarly, the target reliable samples can be generated and denoted as $\mathcal{R}^{t_1}$ and $\mathcal{R}^{t_2}$. For unlabeled target samples, the pseudo-labels $\bar{y}_i^{t1}$ and $\bar{y}_i^{t2}$ are used to select the common samples and perform the small-loss strategy due to the true label are unavailable. Therefore, the supervised loss of the target reliable samples is:

$$L_{SCG}^{t_1} = -\mathop{\mathbb{E}}_{x\in\mathcal{R}^{t_1}}[\mathcal{L}(\mathbf{f}_1(x), \bar{y}^{t_1})],$$
$$L_{SCG}^{t_2} = -\mathop{\mathbb{E}}_{x\in\mathcal{R}^{t_2}}[\mathcal{L}(\mathbf{f}_2(x), \bar{y}^{t_2})], \tag{7}$$

where $L_{SCG}^{t_1}$ and $L_{SCG}^{t_2}$ denote the loss for the models $\mathbf{f}_1$ and $\mathbf{f}_2$ in the Domain Alignment module, respectively.

Finally, the total loss of the SCG component is:

$$L_{SCG} = L_{SCG}^{s_1} + L_{SCG}^{s_2} + L_{SCG}^{t_2} + L_{SCG}^{t_2}. \tag{8}$$

C. Reserve Differences Component

Although the Seek Common Ground component can obtain reliable samples for domain alignment, there still exist many problems. Firstly, two DA models are easy to converge into a consensus since they only select the common samples to update, which deteriorates the efficacy of the SCG component. Secondly, the SCG component only uses the samples with consistent prediction between two domain adaptation models, and ignores the inconsistent samples. Inspired by the Co-teaching+ [13], using the mutual learning between two models can enlarge their divergence, whose key is to let the two models supervise and learn from each other. For example, the pseudo-label $\bar{y}^{f1}$ generated from model $\mathbf{f}_1$ is used to optimize its peer model $\mathbf{f}_2$, vice versa. Therefore, considering the inconsistent samples can enlarge the discrepancy between two symmetrical models in the Domain Alignment module. To address the above issue, the Reserve Differences component is proposed to utilize the samples with inconsistent predictions for maintaining two models diverged and reducing the impact of noisy samples. For the source and target domains, the corresponding inconsistent samples are defined as *source different samples* and *target different samples*. Since the way of obtaining the source different samples and target different samples is similar, we treat the *different source samples* as an example to describe how to obtain and use the inconsistent samples.

Given source dataset $\mathcal{D}_s = \{(x_i^s, \hat{y}_i^s, \bar{y}_i^{s_1}, \bar{y}_i^{s_2})\}_{i=1}^{n_s}$, we first select the inconsistent source samples by considering whether the pseudo-labels $\bar{y}_i^{s_1}$ and $\bar{y}_i^{s_2}$ are different:

$$\mathcal{B}^s = \{x_i^s | \bar{y}_i^{s_1} \neq \bar{y}_i^{s_2}, i \in \mathbb{N}_+^{n_s}\}, \tag{9}$$

where $\mathbb{N}_+^{n_s} = \{1, 2, 3, \ldots, n_s\}$ and $\mathcal{B}^s$ denotes the selected source different samples.

However, some different samples would have adverse effects on domain alignment by using mutual learning. Similar to the SCG component, we adopt the small-loss strategy to select $\tau\%$ of small-loss samples as the different valid samples in each mini-batch data. The selected different valid samples for source domain are denoted as $\mathcal{V}^{s_1}$ and $\mathcal{V}^{s_2}$,

$$\mathcal{V}^{s_1} = \{x_i^s | l_i^{s_1} \leqslant T(\mathbf{L}_{\mathcal{B}^s}^{s_1}, \tau), x_i^s \in \mathcal{B}^s\},$$
$$\mathcal{V}^{s_2} = \{x_i^s | l_i^{s_2} \leqslant T(\mathbf{L}_{\mathcal{B}^s}^{s_2}, \tau), x_i^s \in \mathcal{B}^s\}, \tag{10}$$

where $T(\mathbf{L}_{\mathcal{B}^s}^{s_1}, \tau)$ is a function to determine the threshold that can select $\tau\%$ of small losses from the whole loss set $\mathbf{L}_{\mathcal{B}^s}^{s_1}$ of the different samples. $l_i^{s_1} = \mathcal{L}(\mathbf{f}_1(x_i^s), \hat{y}_i)$ and $l_i^{s_2} = \mathcal{L}(\mathbf{f}_2(x_i^s), \hat{y}_i)$ are the classification loss for sample x_i^s.

Algorithm 1: Seek Common Ground While Reserving Differences.

Input: Two DA models f_1, f_2 with weights θ_1, θ_2, learning rate λ, epoch E_{max} and E_k, noise source set D^s, unlabeled target set D^t, iteration η_{max}, noise rate β, selected proportion $\tau_s(e)$, $\tau_1(e)$, $\tau_2(e)$, hyper-parameters ρ_1, ρ_2;
Output: θ_1 and θ_2;

```
for e = 1, 2, 3, ..., E_max do
  for η = 1, 2, 3, ..., η_max do
    Fetch mini-batch D_η^s from D^s, D_η^t from D^t;
    // Deal with noisy source samples
    Obtain source common samples A^s with consistent predictions by Eq. (4);
    Obtain source different samples B^s inconsistent predictions by Eq. (9);
    Get small-loss source common samples R^{s1} and R^{s2} of each model by Eq. (5) according to τ_s(e);
    Get small-loss source different samples V^{s1} and V^{s1} of each model by Eq. (10) according to τ_s(e);
    Calculate Δθ_1^s = λ∇(L_SCG^{s1} + L_RD^{s1}), Δθ_2^s = λ∇(L_SCG^{s2} + L_RD^{s2}) by Eq. (6) and Eq. (11);
    // Deal with unlabeled target samples similarily
    Obtain target common samples A^t with consistent predictions;
    Obtain target different samples B^t with inconsistent predictions;
    Get small-loss target common samples R^{t1} and R^{t2} of each model according to τ_1(e);
    Get small-loss target different samples V^{t1} and V^{t1} of each model according to τ_2(e);
    Calculate Δθ_1^t = λ∇(L_SCG^{t1} + L_RD^{t1}), Δθ_2^t = λ∇(L_SCG^{t2} + L_RD^{t2}) by Eq. (7) and Eq. (12);
    // Domain Alignment module
    Calculate Δθ_1^a = λ∇L_{align_1}, Δθ_2^a = λ∇L_{align_2} in DA models f_1, f_2;
    //Update θ_1 and θ_2
    Update θ_1 = θ_1 - Δθ_1^s - Δθ_1^t - Δθ_1^a, θ_2 = θ_2 - Δθ_2^s - Δθ_2^t - Δθ_2^a;
  end for
  Update τ_s(e) = 1 − min{e/E_k (β + 0.1), β + 0.1},
  τ_1(e) = min{e/E_k ρ_1, ρ_1} and
  τ_2(e) = min{e/E_k ρ_2, ρ_2};
end for
return θ_1, θ_2.
```

After obtaining $\mathcal{V}^{s_1}$ and $\mathcal{V}^{s_2}$, the given corrupted label $\hat{y}^s$ is used to update its peer model:

$$L_{RD}^{s_1} = - \mathop{\mathbb{E}}_{x \in \mathcal{V}^{s_2}} [\mathcal{L}(\mathbf{f}_1(x), \hat{y}^s)],$$
$$L_{RD}^{s_2} = - \mathop{\mathbb{E}}_{x \in \mathcal{V}^{s_1}} [\mathcal{L}(\mathbf{f}_2(x), \hat{y}^s)], \tag{11}$$

where $L_{RD}^{s_1}$ and $L_{RD}^{s_2}$ denote the loss for models $\mathbf{f}_1$ and $\mathbf{f}_2$ in the Domain Alignment module, respectively.

From Eq. (11), we observe that the valid samples $\mathcal{V}^{s_2}$ generated from the model $\mathbf{f}_2$ is used to update the model $\mathbf{f}_1$, vice versa. The advantage of the above mutual learning is that it can maintain the divergence of two models and reduce the error from noisy labels.

For unlabeled target samples, the different valid samples $\mathcal{V}^{t_1}$ and $\mathcal{V}^{t_2}$ are generated similarly. Since there is no label for target sample, the obtained pseudo-label $\bar{y}^t$ is used to update its peer model:

$$L_{RD}^{t_1} = - \mathop{\mathbb{E}}_{x \in \mathcal{V}^{t_2}} [\mathcal{L}(\mathbf{f}_1(x), \bar{y}_i^{t_2})],$$
$$L_{RD}^{t_2} = - \mathop{\mathbb{E}}_{x \in \mathcal{V}^{t_1}} [\mathcal{L}(\mathbf{f}_2(x), \bar{y}_i^{t_1})], \tag{12}$$

where $L_{RD}^{t_1}$ and $L_{RD}^{t_2}$ denote the target loss for the two models $\mathbf{f}_1$ and $\mathbf{f}_2$ in the Domain Alignment module, respectively.

Finally, the total loss in the RD component is:

$$L_{RD} = L_{RD}^{s_1} + L_{RD}^{s_2} + L_{RD}^{t_1} + L_{RD}^{t_2}. \tag{13}$$

D. Overall Objective

The final model is a combination of the Domain Alignment modules, Seek Common Ground component and Reserve Differences component. Therefore, the overall objective function is:

$$\min_{\mathbf{f}_1, \mathbf{f}_2} L = L_{align} + L_{SCG} + L_{RD}. \tag{14}$$

For the pseudo-code description of the algorithm details, please refer to Algorithm 1.

IV. Experiments

A. Datasets

Office-31 [35] is a well-known dataset for domain adaptation, which contains 31 classes and consists of three domains: **A**mazon, **W**ebcam and **D**SLR, where Amazon domain contains 2817 images collected from amazon.com, Webcam includes 795 images obtained from web camera, and DSLR comprises 498 images shot by SLR camera. Six transfer tasks can be obtained through the permutation of the three domains.

Office-Home [36] is a challenging dataset containing around 15 500 images from 65 different categories. Office-Home consists of four domains: **A**rt, **C**lipart, **P**roduct, and **R**eal-world. The images in Art are paintings, sketches, and artistic depictions. Clipart consists of clipart images. The product is composed of images without background, and Real-World comprises regular images captured with a camera. There are 12 different transfer tasks by permutating four domains.

Bing-Caltech [37] is a real noisy dataset consisting of Bing and Caltech datasets. The images in Bing dataset contain rich noises because they are collected by retrieving the category labels with Bing search engine.

For Office-31 and Office-Home, the uniform label corruption is used to generate noisy labels based on the given clean labels, where each label is flipped into other labels uniformly with the noise rate β by Eq. (1). For Bing-Caltech, we treat Bing dataset as the noisy source domain and Caltech dataset as the clean target domain. The experiments in Bing-Caltech represents the performance in real-world noisy domain adaptation.

B. Implementation Details

We implement the proposed method in the Pytorch platform using Nvidia Titan V100 GPU. For fair comparison, we use the ResNet-50 [30] pre-trained on ImageNet [38] as backbone network. We utilize the task-specific FC layer to replace the last FC layer and finetune the model with labeled source samples and unlabeled target samples. Furthermore, domain-specific batch normalization layers are adopted in the network. For optimization, we use mini-batch SGD with momentum of 0.9, and using the learning rate policy introduced in CAN [17], *i.e.*, the learning rate ψ_p is adjusted by $\psi_p = \psi_0(1 + ap)^{-b}$, where ψ_0 is the initial learning rate, p is the training progress changing from 0 to 1, $a = 10$, and $b = 0.75$. The initial learning rates are set as $1\mathrm{e}^{-3}$ and $1\mathrm{e}^{-2}$ for backbone network and task-specific FC layer, respectively. We utilize all labeled source samples and unlabeled target samples during training. During reference, we ensemble the predictions of the two models as final outputs.

τ is set for selecting the appropriate proportion of small-loss samples in the SCG component and RD component. Regarding the diverse circumstances of the source domain and the target domain, there are different settings of τ for noise source samples and unlabeled target samples. For noise source samples, the noise labels are all used as supervision signals for the supervised loss during training. According to the recent work about memorization effects [39] of deep neural networks, the network would remember the clean samples first, and then overfit on these noise samples gradually. Therefore, τ changes dynamically with epoch e and is defined as $\tau(e)$. For the noise source samples, $\tau_s(e)$ is set to be relatively large to select more samples at the beginning of training. By increasing the epoch number e, $\tau_s(e)$ will decrease linearly. Inspired by Co-teaching+ [13], we set $\tau_s(e) = 1 - \min\{\frac{e}{E_k}(\beta + 0.1), \beta + 0.1\}$ for selecting clean data strictly for the SCG and RD components.

Since there are no labels provided for supervised loss and the domain gap existed between the source and target domains, most of the predicted labels of unlabeled target samples are incorrect at first. Then, the accuracy of predictions increases gradually. Therefore, $\tau_1(e)$ and $\tau_2(e)$ which respectively denote the selection proportion of unlabeled target samples in the SCG component and the RD component both start from 0 at the beginning of training, and increase linearly with epoch e. Since the unlabeled target common samples with the consistent predictions between two models are more likely to be predicted correctly, we set $\tau_1(e) = \min\{\frac{e}{E_k}\rho_1, \rho_1\}$ and $\rho_1 = 0.5$ in the

TABLE I
ACCURACY(%) OF **OFFICE-31** UNDER 40% LABEL CORRUPTION. "+" INDICATES COMBINATION OF OUR MODULE WITH DOMAIN ADAPTATION MODEL

Method	Office-31 40% Label Corruption						
	A→W	W→A	A→D	D→A	W→D	D→W	Avg.
ResNet [30]	47.2	33.0	47.1	31.0	68.0	58.8	47.5
SPL [31]	72.6	50.0	75.3	38.9	83.3	64.6	64.1
MentorNet [11]	74.4	54.2	75.0	43.2	85.9	70.6	67.2
DAN [3]	63.2	39.0	58.0	36.7	71.6	61.6	55.0
RTN [32]	64.6	56.2	76.1	49.0	82.7	71.7	66.7
DANN [33]	61.2	46.2	57.4	42.4	74.5	62.0	57.3
ADDA [25]	61.5	49.2	61.2	45.5	74.7	65.1	59.5
MDD [34]	74.7	55.1	76.7	54.3	89.2	81.6	71.9
TCL [9]	82.0	65.7	83.3	60.5	90.8	77.2	76.6
RDA [10]	89.7	67.2	92.0	65.5	96.0	92.7	83.6
GVB [15]	49.6	36.2	51.4	37.0	51.2	49.8	45.8
MCD [16]	73.0	48.0	76.5	56.7	90.5	82.5	71.2
CAN [17]	86.7	71.5	90.0	73.7	93.4	90.6	84.3
GSR(GVB+SCGWRD)	53.8	36.3	54.6	37.6	57.8	51.8	$48.7^{\uparrow 2.9}$
MSR(MCD+SCGWRD)	79.9	62.8	85.0	66.3	**100.0**	**97.0**	$85.6^{\uparrow 14.4}$
CSR(CAN+SCGWRD)	**91.6**	**71.7**	**93.0**	**74.8**	95.2	94.7	$\mathbf{86.8}^{\uparrow 2.5}$

TABLE II
ACCURACY(%) OF **OFFICE-HOME** UNDER 40% LABEL CORRUPTION

Method	Office-Home 40% Label Corruption												
	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
ResNet [30]	19.8	37.8	46.5	22.3	32.1	30.5	20.5	13.3	37.0	31.8	19.8	50.1	30.1
DANN [33]	25.3	40.4	51.9	36.5	43.2	48.3	34.7	25.8	54.6	46.2	34.3	61.3	41.9
MDD [34]	**42.2**	59.9	**66.9**	47.2	59.0	59.8	40.6	34.5	60.9	55.2	42.9	73.3	53.5
TCL [9]	21.1	35.6	61.4	16.1	44.6	36.4	24.6	30.4	**68.7**	59.9	25.7	68.6	44.1
RDA [10]	40.3	56.9	64.3	46.9	57.1	59.7	41.2	32.6	59.7	51.1	42.0	71.0	51.9
CAN [17]	27.7	53.1	59.5	33.1	55.8	53.2	31.3	30.3	56.6	38.4	33.6	65.6	44.9
CSR	40.2	**62.4**	65.3	**55.4**	**67.5**	**63.8**	**52.3**	**47.2**	**68.7**	**64.3**	**51.0**	**73.5**	**59.3**

SCG component. Different from the SCG component, we set $\tau_2(e) = \min\{\frac{e}{E_k}\rho_2, \rho_2\}$ and $\rho_2 = 0.1$ in the RD component because the unlabeled different samples with inconsistent predictions are less reliable. ρ_1 and ρ_2 are hyper-parameters and will be analyzed in ablation studies.

C. Comparison With Existing Methods

To demonstrate the effective of SCGWRD, we incorporate the proposed module into existing DA methods, *e.g.,* Gradually Vanishing Bridge (GVB) [15], Maximum Classifier Discrepancy (MCD) [16] and Contrastive Adaptation network (CAN) [17], and make comparison with existing methods [9]–[11], [15], [16], [25], [30]–[34]. The Transferable Curriculum Learning (TCL) [9] and Robust Domain adaptation (RDA) [10] are state-of-the-art noisy domain adaptation methods. For the GVB, MCD, and CAN, we reimplement them with the released code.

Office-31: The evaluation on Office-31 with noise rate 40% is summarized in Table I. From Table I, we can observe that combining the proposed SCGWRD and CAN (CSR) achieves the best performance among all the methods. Especially for the noisy domain adaptation methods TCL and RDA, CSR obtains noticeable improvements, *e.g.,* obtaining 10.2% and 3.2% improvements for TCL and RDA, respectively. We also observe that the baseline CAN has obtained a higher performance than TCL and RDA. The reason is that the CAN is a statistical-based method, which is robust to noise corruption. Moreover, from Table I we can see that incorporating the proposed SCGWRD into existing methods can boost their performance, *e.g.,* improving the average performance from 45.8%, 71.2%, and 84.3% to 48.7%, 85.6%, and 86.8% for GVB, MCD, and CAN, respectively. The improvement shows the effectiveness of the SCGWRD for noisy domain adaptation.

Office-Home: Table II summarizes the related results for Office-Home under 40% label corruption. By comparing Table I and Table II, we observe that the Office-Home is a more challenging dataset than Office-31, *e.g.,* the existing state-of-the-art performances are 53.5% and 84.8% for Office-Home and Office-31, respectively. The CSR model that incorporates our SCGWRD module into CAN achieves the best performance among all the methods, *e.g.,* obtaining the performance of 59.3% *vs* 53.5% for MDD [34]. Furthermore, adding the proposed SCGWRD module obtains the improvement of 14.4% over CAN [17].

Bing-Caltech: Different from the above comparison, we further conduct comparison on the **real-world** noisy datasets, *e.g.,* Bing-Caltech. The comparison of Bing → Caltech is shown in Fig. 3, from which we can see that CSR performs better than other methods, *e.g.,* outperforms the state-of-the-art method RDA by 2.7%. The result in Bing-Caltech proves the effectiveness of our SCGWRD module in real-world noisy domain adaptation.

Based on the above comparison of three different datasets, we can conclude that SCGWRD is a useful module for noisy domain adaptation.

D. Ablation Studies

By taking the existing CAN [17] as the baseline, we give some ablation studies to show the effectiveness and rationality

TABLE III
EFFECT OF THE SEEK COMMON GROUND (SCG) COMPONENT AND RESERVE DIFFERENCE (RD) COMPONENT. DAM REPRESENTS DOMAIN ALIGNMENT MODULE. ACCURACY(%) OF OFFICE-31 UNDER 40% LABEL CORRUPTION ARE REPORTED

Method	Components			Office-31 40% Label Corruption						
	DAM	SCG	RD	A→W	W→A	A→D	D→A	W→D	D→W	Avg
CAN	√			86.7	71.5	90.0	73.7	93.4	90.6	84.3
COM	√	√		85.8	67.4	88.3	66.0	95.0	77.6	80.0
CRD	√		√	**92.7**	67.9	92.2	67.6	94.9	93.5	84.8
CSR	√	√	√	91.6	**71.7**	**93.0**	**74.8**	**95.2**	**94.7**	**86.8**

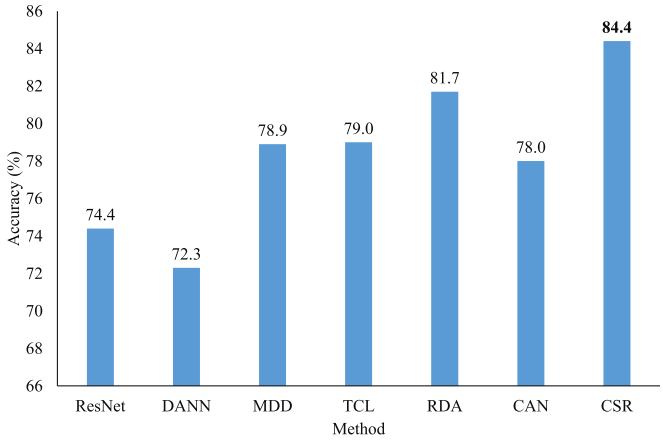


Fig. 3. Accuracy(%) of **Bing-Caltech** with Real-world Noises.

TABLE IV
EFFECT OF SCGWRD ON DIFFERENT DOMAINS. '-S' AND '-T' DENOTE THE MODELS FOR THE SOURCE AND TARGET DOMAINS, RESPECTIVELY. ACCURACY(%) OF OFFICE-31 UNDER 40% LABEL CORRUPTION ARE REPORTED

Method	Office-31 40% Label Corruption						
	A→W	W→A	A→D	D→A	W→D	D→W	Avg
CAN	86.7	71.5	90.0	73.7	93.4	90.6	84.3
CSR-s	91.1	71.2	**93.0**	73.6	95.0	94.5	86.4
CSR-t	90.6	71.2	91.7	74.0	94.5	91.1	85.5
CSR	**91.6**	**71.7**	**93.0**	**74.8**	**95.2**	**94.7**	**86.8**

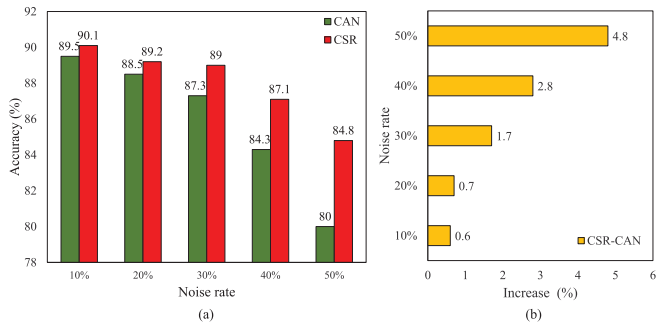


Fig. 4. (a) Represents the mean accuracy of CAN and CSR under different noise rate in Office-31. (b) Depicts the accuracy gap between CSR and CAN under different noise rates.

of the proposed SCGWRD module on Office-31 under 40% label corruption.

Effect of SCG and RD components. To demonstrate the rationality of the Seek Common Ground (SCG) component and Reserve Differences (RD) component, we analyze two components separately and summarize the results in Table III. As shown in Table III, the COM model that incorporates the Seek Common Ground component into CAN obtains a lower performance than CAN, *e.g.,* 84.3% *vs* 80.0%. Although using the SCG component can discover reliable samples, the selected samples still contain many noises. Merely considering the common samples leads to that the adverse effects of these noises are amplified due to self-training. Unlike the SCG component, we observe that using Reserve Differences component obtains a higher performance than CAN. CRD combines the RD component with CAN and improves the mean performance from 84.3% to 84.8%. The reason is that considering the different samples can maintain divergences between two models, and use the small-loss samples of each model to train its peer model can reduce the error from noisy labels by peer models mutually. As the RD component can enlarge the divergences between two models by using the different samples, and the SCG component can discover the reliable samples, jointly considering these two components can select more reliable samples by discarding the noisy samples. By combining the SCG and RD components, the final CSR model achieves the highest performance, *e.g.,* obtaining the mean performance of 86.8%.

Effect on different domains. Besides the noisy source images, the pseudo-labels for unlabeled target samples can be treated as noisy labels. Therefore, we conduct experiments to show that the proposed SCGWRD module can be effectively applied for noisy source samples and unlabeled target samples, and summarize the results in Table IV. CSR-s and CSR-t indicate that the SCGWRD module is merely applied on noisy source labeled samples and unlabeled target samples, respectively. As shown in Table IV, CSR-s and CSR-t both obtain a higher performance than the baseline CAN, which demonstrates the effectiveness of our proposed SCGWRD for the source and target domains. After jointly optimizing the source and target images, the CSR model obtains the best performance of 86.8%. Therefore, the proposed SCGWRD module is effectively for noisy source samples and unlabeled target samples in noisy domain adaptation.

Noise rate β. For noisy domain adaptation, the noise rate is a critical parameter. We thus give a detailed analysis of the noise rate β, and show the results in Fig. 4. As shown in Fig. 4, increasing the noise rate would degrade the performance. However, CSR that combines our SCGWRD module with CAN has a lower drop rate than CAN, proving that the proposed module is robust to label noise. We also consider a boisterous environment by setting the noise rate to 80%. Under this setting, the accuracy of CAN is 23.5% while CSR achieves the performance of 37.1%.

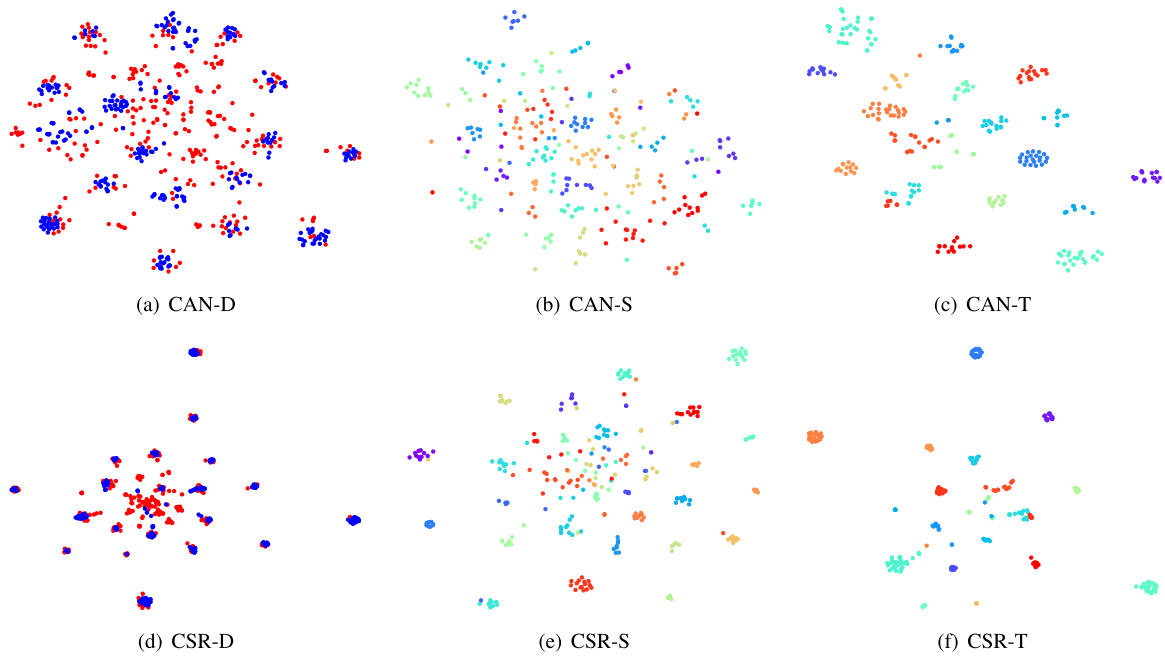


Fig. 5. The feature visualization of A → D task in Office-31 about CAN and CSR under 40% Label Corruption. (a) and (d) represent the source features and target features in CAN and CSR, respectively. Red dots indicate source features and blue dots indicate target features. (b) and (e) denote the source features of each class in CAN and CSR, respectively. (c) and (f) depict the target features of each class in CAN and CSR, respectively. In (b), (c), (e) and (f), different colors represent the features of different classes.

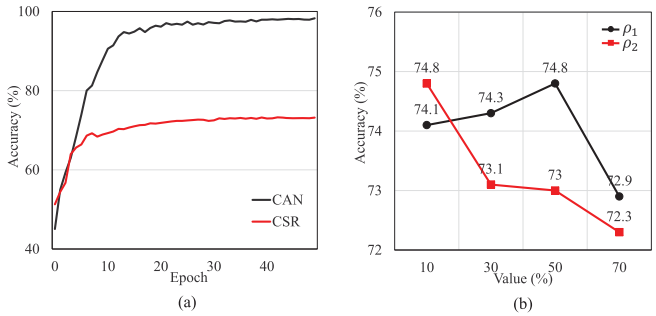


Fig. 6. (a) denotes the variation of training accuracy for D → W. (b) shows the effect of hyper-parameters ρ_1 and ρ_2 for D → A.

Overfitting to noisy data. The critical of noisy domain adaptation is how to avoid the overfitting to noisy source labels. We thus analyze the classification accuracy of noisy source data to show that our SCGWRD module can avoid the overfitting of noisy source samples. As shown in Fig. 6(a), CAN overfits the noise label during the training stage, *e.g.,* the training accuracy reaches to 100% for the source corrupted labels, which means that the noisy samples have been "corrected" classified. Different from CAN, the training accuracy finally reaches to 70% for CSR. As the training samples contain 40% noise samples, our module can avoid overfitting to noisy data, and use clean samples to train the Domain Alignment modules.

Hyper-parameters analysis. The threshold for selecting samples is critical for the small-loss strategy. We finally give some analysis of the hyper-parameters ρ_1 and ρ_2, which determine the proportion of selected common and different target samples in each mini-batch data, respectively. We evaluate the effect of ρ_1 and ρ_2 in D → A task of Office-31, and summarize the related results in Fig. 6(b). As shown in Fig. 6(b), a higher $\rho_1 = 50\%$ is used to select sufficient and reliable enough common samples. Moreover, a lower $\rho_2 = 10\%$ is applied to discover the different samples with higher confidence. We observe that the accuracy would be decreased when setting the higher or lower value of ρ_1. The reason is that ρ_1 is related to the noise rate β and seeking more or less common target samples is negative for transfer learning. Different from ρ_1, the higher ρ_2 the lower performance. The reason is that using a higher ρ_2 would select many noisy samples in the RD component.

Feature visualization. We further illustrate the effectiveness of our proposed module by visualizing the features in A → D transfer task of Office-31 under 40% label corruption. We utilize T-SNE to visualize the visual descriptions obtained by the last FC layer in the source domain and target domain, as shown in Fig. 5. Fig. 5 (a) and (d) show the source features and target features in CAN and CSR, respectively. From Fig. 5 (a) and (d), we can observe that the source features and target features are better aligned in CSR. Fig. 5 (b) and (e) denote the source features of each class in CAN and CSR, respectively, from which we can see that CSR obtains more discriminative and correct source features. Fig. 5 (c) and (f) depict the target features of each class in CAN and CSR, respectively. As shown in Fig. 5 (c) and (f), the CSR obtains more discriminative target features than CAN. To sum up, the CSR can effectively eliminate the influence of label corruption, and gets more discriminative and alignment features.

V. Conclusion

The key of noisy domain adaptation is to discover the noisy samples and reduce the negative effect caused by these samples. Based on the predictions of existing domain adaptation methods, we propose a model-agnostic module, named Seek Common Ground While Reserving Differences (SCGWRD), to discover the common samples and different samples for domain alignment. As SCGWRD merely relies on the outputs of domain adaptation methods, it can be incorporated into any existing domain adaptation method. The evaluations of three benchmarks demonstrate the effectiveness and generalization of SCGWRD. Although SCGWRD can effectively discover the noisy samples, it is a complex module because it relies on two domain adaptation models. In the future, we will explore how to use a single model to discover noise samples effectively.

References

[1] S. Ben-David *et al.*, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, 2010.

[2] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[3] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[4] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3722–3731.

[5] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 801–814, Apr. 2019.

[6] H. Song, X. Wu, W. Yu, and Y. Jia, "Extracting key segments of videos for event detection by learning from web sources," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1088–1100, May 2018.

[7] X. Ma, T. Zhang, and C. Xu, "Deep multi-modality adversarial networks for unsupervised domain adaptation," *IEEE Trans. Multimedia*, vol. 21, no. 9, pp. 2419–2431, Sep. 2019.

[8] Y. Zuo, H. Yao, and C. Xu, "Attention-based multi-source domain adaptation," *IEEE Trans. Image Process.*, vol. 30, pp. 3793–3803, 2021.

[9] Y. Shu, Z. Cao, M. Long, and J. Wang, "Transferable curriculum for weakly-supervised domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4951–4958.

[10] Z. Han, X.-J. Gui, C. Cui, and Y. Yin, "Towards accurate and robust domain adaptation under noisy environments," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, C. Bessiere, Ed, 7 2020, pp. 2269–2276.

[11] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2304–2313.

[12] B. Han *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8527–8537.

[13] X. Yu *et al.*, "How does disagreement help generalization against label corruption?" in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7164–7173.

[14] E. Malach and S. Shalev-Shwartz, "Decoupling" when to update" from" how to update," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 960–970.

[15] S. Cui *et al.*, "Gradually vanishing bridge for adversarial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12455–12464.

[16] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3723–3732.

[17] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4893–4902.

[18] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.

[19] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4058–4065.

[20] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," in *Proc. Int. Conf. Learn. Representations*, no. 6, 2018.

[21] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2017.

[22] Y. Zuo, H. Yao, and C. Xu, "Category-level adversarial self-ensembling for domain adaptation," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2020, pp. 1–6.

[23] Y. Zuo, H. Yao, L. Zhuang, and C. Xu, "Margin-based adversarial joint alignment domain adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, 2021, doi: 10.1109/TCSVT.2021.3081729.

[24] H. Yan *et al.*, "Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation," *IEEE Trans. Multimedia*, vol. 22, no. 9, pp. 2420–2433, Sep. 2020.

[25] E. Tzeng *et al.*, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.

[26] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5419–5428.

[27] T. Shermin, G. Lu, S. W. Teng, M. Murshed, and F. Sohel, "Adversarial network with multiple classifiers for open set domain adaptation," *IEEE Trans. Multimedia*, pp. 1–1, 2020.

[28] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, doi: 10.1109/TMM.2020.3016126.

[29] D. Arpit *et al.*, "A closer look at memorization in deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 233–242.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[31] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1189–1197.

[32] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[33] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[34] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7404–7413.

[35] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

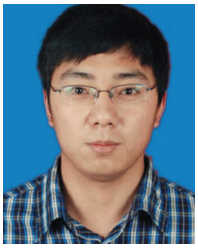
[36] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5018–5027.

[37] A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 181–189.

[38] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[39] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.

Yukun Zuo received the B.S. degree, in 2018 in information security from the University of Science and Technology of China, Hefei, China, where he is currently working toward the Ph.D. degree. His research interests include computer vision and machine learning.



Hantao Yao (Member, IEEE) received the B.S. degree from XiDian University, Xi'an, China, in 2012 and the Ph.D. degree with the Institute of Computing Technology, University of Chinese Academy of Sciences, Beijing, China, in 2018. After graduation from 2018 to 2020, he was a Postdoctoral with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently an Assistant Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include zero-shot learning, person tracking and detection, and person re-identification. He was the recipient of National Postdoctoral Programme for Innovative Talents.

Liansheng Zhuang (Member, IEEE) received the bachelor's and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2001 and 2006, respectively. In 2011, he was nominated to join the STARTRACKER Project of Microsoft Research of Asia (MSRA), and he was a Vendor Researcher with the Visual Computing Group, Microsoft Research, Beijing, China. From 2012 to 2013, he was a Visiting Research Scientist with the Department of EECS, University of California at Berkeley, Berkeley, CA, USA. He is currently an Associate Professor with the School of Information Science and Technology, USTC. His main research interests include computer vision, and machine learning. He is a member of ACM and CCF.

Changsheng Xu (Fellow, IEEE) is currently a Professor with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He has hold 50 granted/pending patents and published over 400 refereed research papers in his research fields, which include multimedia content analysis, pattern recognition, and computer vision. He was the Editor-in-Chief, an Associate Editor, the Guest Editor, the General Chair, the Program Chair, the Area or Track Chair, and a TPC Member for more than 20 IEEE and ACM prestigious multimedia journals, conferences and workshops, including the IEEE TRANSACTION ON MULTIMEDIA, *ACM Transaction on Multimedia Computing, Communications and Applications,* and ACM Multimedia conference. He is IAPR Fellow and ACM Distinguished Scientist.