Hard Class Rectification for Domain Adaptation

Yunlong Zhang[a], Changxing Jing[a], Huangxing Lin [a], Chaoqi Chen[a], Yue Huang[a,*], Xinghao Ding[a] and Yang Zou[b]

[a]Fujian Key Laboratory of Sensing and Computing for SmartCity, School of Informatics, Xiamen University, Xiamen, Fujian, 361005, China
[b]Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213, U.S.A.

ARTICLE INFO

Keywords:
Unsupervised Domain Adaptation
Semi-Supervised Domain Adaptation
Pseudo-labeling
hard class problem

ABSTRACT

Domain adaptation (DA) aims to transfer knowledge from a label-rich and related domain (source domain) to a label-scare domain (target domain). Pseudo-labeling has recently been widely explored and used in DA. However, this line of research is still confined to the inaccuracy of pseudo labels. In this paper, we explore the imbalance issue of performance among classes in-depth and observe that the worse performances among all classes are likely to further deteriorate in the pseudo-labeling, which not only harms the overall transfer performance but also restricts the application of DA. In this paper, we propose a novel framework, called Hard Class Rectification Pseudo-labeling (HCRPL), to alleviate this problem from two aspects. First, we propose a simple yet effective scheme, named Adaptive Prediction Calibration (APC), to calibrate predictions of target samples. Then, we further consider the predictions of calibrated ones, especially for those belonging to the hard classes, which are vulnerable to perturbations. To prevent these samples to be misclassified easily, we introduce Temporal-Ensembling (TE) and Self-Ensembling (SE) to obtain consistent predictions. The proposed method is evaluated on both unsupervised domain adaptation (UDA) and semi-supervised domain adaptation (SSDA). Experimental results on several real-world cross-domain benchmarks, including ImageCLEF, Office-31, Office+Caltech, and Office-Home, substantiate the superiority of the proposed method.

1. Introduction

Over the last few years, Deep Neural Networks (DNNs) [29] achieved impressive performance in machine learning tasks, such as computer vision [22], speech recognition [1], medical analysis [74], industrial fault diagnosis [35, 36], and so on. Nevertheless, collecting and annotating large-scale training data in distinct domains for various applications is an expensive and labor-intensive process. Meanwhile, the application of DNNs is greatly limited because the learned network shows poor generalization ability when it encounters new environments. Domain adaptation (DA) [43] serves as an ideal solution for addressing this problem. It has raised widespread attentions [2, 16] in the machine learning community.

The majority of existing DA methods [21, 37, 15, 16, 57, 45, 62, 56, 54] were devoted to aligning source and target features by decreasing domain divergence, and these methods can be supported by the theoretical analysis of DA [2]. However, there are still two main limitations with these approaches: 1) the methods of global alignment of the source and target features cannot guarantee correct alignment of class-level representations, and 2) the global alignment methods cannot learn target-discriminative representations. Aligning the class conditional distributions of the source and target domains is an effective tool to tackle these limitations. However, directly pursuing the alignment of class conditional distributions is impossible due to the absence of target labels.

Pseudo-labeling [30] was first employed for semi-supervised learning tasks. Recently, it was also introduced into DA to solve the aforementioned limitations by alternatively selecting the target samples with high confident predictions as the pseudo-labeled target set (labeling phase) and training the model with the source domain and pseudo-labeled target set (training phase). Although pseudo-labeling is considered to be a promising paradigm, it is still limited due to inevitable false pseudo labels. Zhang et al. [71] demonstrated that false labels are easily fit by DNN, which harms its generalization. Retraining DNN with the false pseudo-labeled samples does not guarantee the generalization ability of the target domain. We further analyze the theory of DA [2] in Section 3 and demonstrate that the expected error on the target domain is determined by false pseudo labels ratio (i.e., (the number of incorrectly pseudo-labeled samples)/(the number of pseudo-labeled samples)). Therefore,

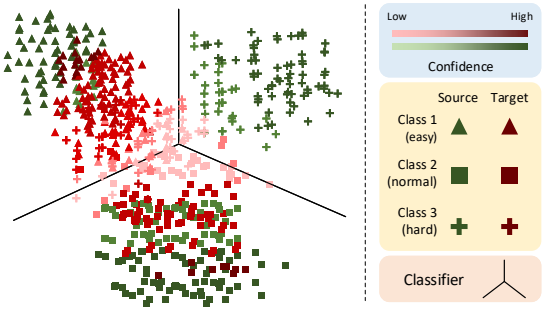


Figure 1: Hard class problem in existing pseudo-labeling based DA methods: Compared with class 1 and 2, class 3 has lower predictive class proportion (i.e., (the number of samples classified into a certain class)/(the number of target samples)). Meanwhile, for this class, target samples with higher confidence are mainly classified into class 1.

yhuang2010@xmu.edu.cn (Y. Huang)
ORCID(s):

reducing the false pseudo labels ratio is of crucial importance to pseudo-labeling methods. To achieve this, Zou et al. [76] enhanced pseudo-labeling from two aspects. 1) They introduced self-paced learning which generates pseudo labels from easy to hard to alleviate error accumulation of pseudo labels. 2) They utilized different confidence thresholds to select a target pseudo-labeled set for different classes. Zou et al. [75] then introduced confidence regularization to avoid overconfident labels. Saito et al. [51] adopted two classifiers with a multiview loss to label the target samples and used a fixed confidence threshold to pick up the reliable pseudo labels. Some works [66, 6, 11, 9] generate pseudo labels in the feature space and adopt different distances to measure confidence.

In this paper, we explore a problem that is unconsidering in the aforesaid methods. As shown in Figure 1, the classifier is trained on the source domain, and three classes deliver distinct performances. Class 1 belongs to easy classes and has a higher predictive class proportion (i.e., (the number of samples classified into a certain class)/(the number of target samples)). The target samples belonging to these classes are very likely to be classified correctly without extra manipulation. The classifier trained on the source domain can well generalize to the target domain for these classes. Class 2 belongs to normal classes and has a moderate predictive class proportion. Although some target samples belonging to it are misclassified, the predictions with higher confidence have higher accuracy. Therefore, the existing pseudo-labeling methods [76, 75, 51, 66, 6, 11, 9] progressively improve the performance on these classes by adding the samples with higher confident predictions into training. Class 3 belongs to hard classes and has a lower predictive class proportion. Meanwhile, for hard classes, even the target samples with higher confident predictions also are highly possible to be misclassified, adding these samples into training will misguide the classifier. Therefore, the pseudo-labeling methods cannot improve the performance of the hard classes and even deteriorate it, which is the main difference between the normal and hard classes.

To tackle the hard class problem, we present a simple yet effective scheme, named Adaptive Prediction Calibration (APC), and it calibrates the predictions of target samples to promote the hard classes, to maintain the normal ones, and to attenuate the easy ones. Furthermore, we consider the calibrated predictions are unstable and unreliable in the pseudo-labeling from the following aspects. First, DNNs are vulnerable to target samples since they are far away from the source domain [59] (i.e., distributional mismatch). Despite encountering a small perturbation (e.g., different augmentations, different classifiers), the predictions of target samples are changed drastically. Second, the APC will further magnify and deteriorate the unrobustness for hard classes since their predictions are enlarged proportionally (i.e., deviation magnification). To ensure the reliability of calibrated predictions, we propose two ensembling methods, Temporal-Ensembling (TE) and Self-Ensembling (SE).

The proposed schemes can be directly combined with the existing pseudo-labeling methods. In this paper, based on CBST [76], we propose a novel pseudo-labeling framework, which combines APC, SE, and TE to alleviate the hard class problem. The proposed framework is called Hard Class Rectification Pseudo-labeling (HCRPL).

The main contributions of this work can be summarized as follows:

i) We reveal the hard class problem in DA, which harms the performance of pseudo-labeling and restricts its applications.

ii) We propose a calibration method (i.e., APC) to alleviate the hard class problem by promoting the hard classes, maintaining the normal ones, and attenuating the easy ones. Furthermore, the predictions of target samples are vulnerable because of the distributional mismatch and deviation magnification. Hence, we introduce TE and SE to improve the reliability of predictions.

iii) We evaluate HCRPL on four public datasets under both UDA and SSDA settings. Extensive experimental results show that the proposed method achieves promising results in various DA scenarios.

2. Related Work

2.1. Unsupervised Domain Adaptation

Under the UDA setting, we are given a set of labeled source samples and a set of unlabeled target examples. According to Ben-David et al.'s [2] theoretical analysis of DA, the expected error for the target domain depends on three terms: expected error on the source domain, domain divergence, and shared error of ideal joint hypothesis, can divide UDA methods into two parts. In the first part, researchers assumed that the shared error of the ideal joint hypothesis was small and mainly focused on decreasing the domain divergence. One method of aligning distributions is through minimizing statistical divergences that measure the distance between two distributions. Representative divergences mainly include Maximum mean discrepancy (MMD) [21], Correlation alignment (CORAL) [56], Contrastive domain discrepancy (CDD) [25], and so on. Inspired by GAN [19], numerous methods [15, 16, 57, 45, 62, 56, 54, 7, 8] were proposed to align source and target domains by adversarial training. Although domain divergence was decreased, the shared error of the ideal joint hypothesis would be large if class conditional distributions are not aligned and separated. In the second part, researchers paid more attention to decreasing the shared error of the ideal joint hypothesis.

Among many schemes, pseudo-labeling is a promising paradigm for reducing the third term. Saito et al. [51] adopted two classifiers to label the target set and made a constraint for the weights of two classifiers to make them different from each other. Zou et al. [76] introduced self-paced learning which generates pseudo labels from easy to

hard to alleviate error accumulation of pseudo labels. Furthermore, they utilized different confidence thresholds to select predicted target samples for various classes. Zou et al. [75] introduced confidence regularization to prevent putting overconfident label belief in the wrong classes. On par with these methods that generate pseudo labels based on predictions, some methods generate pseudo labels in the feature space. Xie et al. [66] introduced feature centroids alignment after pseudo-labeling to DA. Chen et al. [6] proposed a progressive feature alignment that takes advantage of intra-class distribution variance in pseudo-labeling for UDA problems. Deng, Zheng, and Jiao [11] introduced similarity-preserving constraint that can be implemented by minimizing the triplet loss with labeled source features and pseudo-labeled target features. Li et al. [33] introduced label propagation to update pseudo labels and proposed landmark selection to re-weight the samples of the source and target domains. Wu et al. [64] determined pseudo labels by integrating the predictions of multiple classifiers. Chen et al. [10] improved the quality of pseudo labels by refining labels after each iteration. The proposed HCRPL is also based on pseudo-labeling and aims to improve the accuracy of pseudo labels by exploring the imbalance issue of performance among classes.

2.2. Semi-Supervised Domain Adaptation

Since most DA methods focus on the unsupervised setting, SSDA has not been well studied. Several recent work [46, 70, 34, 26, 50] show that SSDA can effectively boost the performance by adding merely few target labeled data (e.g. just one labeled image per class), suggesting that this setting may be more valuable in practical applications. For SSDA, the key to improving performance is learning target-discriminative representations [50]. In Saito et al.'s study [50], standard UDA methods [15, 38, 52] were shown to be empirically less effective in SSDA because they fail to learn discriminative class boundaries on the target domain. By optimizing a minimax loss on the conditional entropy of unlabeled data and a task loss, Saito et al. [50] reduced the distribution gap while learning discriminative features. Motiian et al. [41] exploited the Siamese architecture to learn an embedding subspace that is discriminative and where mapped visual domains are semantically aligned and yet maximally separated. Qin et al. [47] proposed a framework consisting of a generator and two classifiers, where one is a source-based classifier and the other is a target-based classifier. The target-based classifier attempts to cluster the target features to improve intra-class density and enlarge inter-class divergence; the source-based classifier is designed to scatter the source features to enhance the smoothness of the decision boundary. Yan et al.[68] proposed a semi-supervised entropic GromovWasserstein discrepancy approach to incorporate the supervision information when learning the optimal transport. The proposed HCRPL can promote target-discriminative representations of the easy, normal, and hard classes. Hence, it also contributes to SSDA.

3. Hard Class Problem

In this section, we first design experiments to confirm the existence of the hard class problem and then emphasize the significance of alleviating it from two practical aspects.

We set "Webcam" as the source domain and "Amazon" as the target domain and then adopt CBST to solve the domain shift problem between them. To investigate class-level performance, we choose precision, recall, and f1-score as metrics and report the results in Figure 6. Compared with training on source domain only, CBST enhances the performances on the majority of classes. We further observe the classes with lower precision, recall, or f1-score (e.g. 4-th and 28-th classes). These classes are so-called hard classes since low precision represents that massive predictions with high confidence are false, and low recall represents those correct predictions with high confidence are small. As a result, the hard class problem may further worsen the performance for these classes. For example, the precision, recall, and f1-score of the 28-th class decrease significantly.

Next, we emphasize two problems caused by the hard class problem. Firstly, it is noteworthy that the worst performance among all classes is more concerned rather than the average one in many applications. For instance, in the application of transfer the knowledge of anomaly detection from the source domain to target one, we expect to detect the anomalies of every type, but the hard class problem will lead to the poor performance of hard classes. Secondly, the hard class problem also will have a negative effect on overall transfer performance, and our explanation is based on Ben-David et al's [2] theoretical analysis of DA.

Theorem 3.1. *Let H be the hypothesis class. Given two different domains $\mathcal{S}$ and $\mathcal{T}$, we have*

$$\forall h \in \mathcal{H}, R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + C, \quad (1)$$

where the expected error on target samples $R_{\mathcal{T}}(h)$ are bounded by three terms: (1) the expected error on source domain, $R_{\mathcal{S}}(h)$; (2) $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$ is the domain divergence measured by a discrepancy distance between the source domain distribution $\mathcal{S}$ and the target domain distribution $\mathcal{T}$ w.r.t. *a hypothesis set $\mathcal{H}$; and (3) the shared error of ideal joint hypothesis C. $C = min_{h\in\mathcal{H}}[\epsilon_{\mathcal{S}}(h, f_{\mathcal{S}}) + \epsilon_{\mathcal{T}}(h, f_{\mathcal{T}})]$, and $\epsilon_{\mathcal{S}}(h)$ is the expected error of h on source domain.*

In this study, we focus on the third term that is the shared error of ideal joint hypothesis C. According to triangle inequality for classification error [2], that is, for any labeling functions f_1, f_2 and f_3, we have $\epsilon(f_1, f_2) \leq \epsilon(f_1, f_3) + \epsilon(f_2, f_3)$, we could have

$$\begin{aligned} C &= \min_{h\in\mathcal{H}} \epsilon_{\mathcal{S}}(h, f_{\mathcal{S}}) + \epsilon_{\mathcal{T}}(h, f_{\mathcal{T}}) \\ &\leq \min_{h\in\mathcal{H}} \epsilon_{\mathcal{S}}(h, f_{\mathcal{S}}) + \epsilon_{\mathcal{T}}(h, f_{\mathcal{T}_l}) + \epsilon_{\mathcal{T}}(f_{\mathcal{T}_l}, f_{\mathcal{T}}), \end{aligned} \quad (2)$$

$\epsilon_{\mathcal{S}}(h, f_{\mathcal{S}}) + \epsilon_{\mathcal{T}}(h, f_{\mathcal{T}_l})$ denotes the shared error of h^* on source domain $\mathcal{S}$ and pseudo-labeled set $\mathcal{D}_l$ and is minimized by the training model with source domain $\mathcal{S}$ and

pseudo-labeled set $\mathcal{D}_l$. $\epsilon_{\mathcal{T}}(f_{\mathcal{T}_l}, f_{\mathcal{T}})$ denotes false pseudo labels ratio. Overall, the expected error on the target domain is determined by the false pseudo labels ratio.

Next, we link the hard class problem with the false pseudo labels ratio from the following two aspects. Firstly, few target samples are classified into hard classes, and the model cannot learn to target-discriminative representations for the hard classes. Secondly, the hard class problem will cause error accumulation and increase the false pseudo labels ratio since pseudo-labeling may select false predictions with high confidence, and then these false predictions will misguide the classifier.

Algorithm 1 Overall workflow for HCRPL

Require: rounds Rs, epochs Es, the source domain $\mathcal{D}_s$, the target domain $\mathcal{D}_u$, pre-trained network parameter θ.
Ensure: trained network parameter θ.

```
Calculate the initial ensemble predictions Z = {z_i^u}_{i=1}^{m_u} of the target domain D_u based on the pre-trained model.
Let training set D_tr = D_s.
Let pseudo-labeled set D_l = ∅.
for r = 1 to Rs do
    for e = 1 to Es do
        Train network.                                  ▷ Training phase
        Update the ensemble predictions Z.              ▷ Predicting phase
    end for
    Select target samples with confident predictions and add them into pseudo-labeled set D_l.   ▷ Selecting phase
    Update training dataset D_tr = D_s ∪ D_l.
end for
```

4. Methods

4.1. Preliminary

This section describes HCRPL based on the UDA setting. Under this setting, a source domain $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{m_s}$ and a target domain $\mathcal{D}_u = \{(x_i^u)\}_{i=1}^{m_u}$ are given. We define pseudo-labeled set as $\mathcal{D}_l = \{(x_i^u, \hat{y}_i^u\}_{i=1}^{m_t}$. Specifically, y_i^s and $\hat{y}_i^u$ are one-hot vectors. Meanwhile, we assume that source and target domains contain same object classes, and we consider C classes. Under the SSDA setting, the union of source domain and labeled target set is regarded as the new source domain while the unlabeled target set is regarded as the new target domain.

The proposed HCRPL belongs to pseudo-labeling, but it is different from the standard pseudo-labeling where target samples are predicted after every epoch but not every round. To describe the proposed HCRPL more clearly, the labeling phase is divided into predicting and selecting phases. The overall training process is given in Algorithm 1 and Figure 2. It mainly includes three phases as follows:

1) Training phase: training network with training set $\mathcal{D}_{tr}$.

2) Predicting phase: generating ensemble predictions Z of target samples $\mathcal{D}_u$.

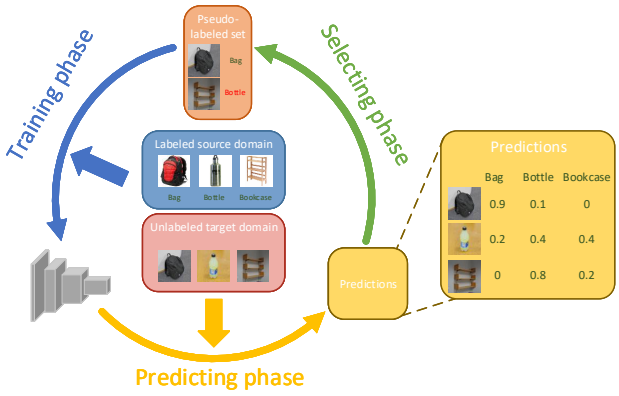


Figure 2: The overall framework of the proposed HCRPL, which is mainly composed of training, predicting, and selecting phases. The whole framework undergoes these three phases alternatively.

3) Selecting phase: selecting target samples with confident predictions as the pseudo-labeled set $\mathcal{D}_l$.

We define going through the process from training the network to updating $\mathcal{D}_{tr}$ once as one round. The proposed APC, TE, and SE are included in the predicting phase. The overall training process along with APC, TE, and SE are introduced below.

4.2. Adaptive Prediction Calibration

Adaptive prediction calibration is the core component of our approach. In the hard class problem, target samples are difficult to be classified into hard ones, which leads to lower predictive class proportions for hard classes. One way to alleviate this problem is magnifying the predictive proportions of the hard classes while keeping the normal ones and suppressing the easy ones. To control predictive class proportions in a reasonable interval, we try to eliminate the mismatch between predictive and true class proportions for the target domain. However, the latter is always unknown in practice. Here, we replace it with the class proportion of the source domain, which is reasonable due to the following two aspects. First, the label proportions of the source and target domains always are close, which is valid in many practical applications. Second, we demonstrate that the minor disagreement of label proportions slightly affects transfer performance by experiments, the details of which are shown in Section 5.6.1.

The detailed process of APC is shown in Figure 3 top. The target domain $\mathcal{D}_u$ is first fed into the trained model to obtain predictions $P = \{p_i^u\}_{i=1}^{m_u}$. Then, we define a ratio R of class distribution of the source domain $q(y)$ to predictive class distribution $p(y)$ as follows

$$R = q(y) \oslash p(y), \tag{3}$$

where $q(y) = \frac{1}{m_s}\sum_{i=1}^{m_s} y_i^s$, $p(y) = \frac{1}{m_u}\sum_{i=1}^{m_u} p_i^u$, $\oslash$ means element-wise division, and R is a C-dimensional vector with

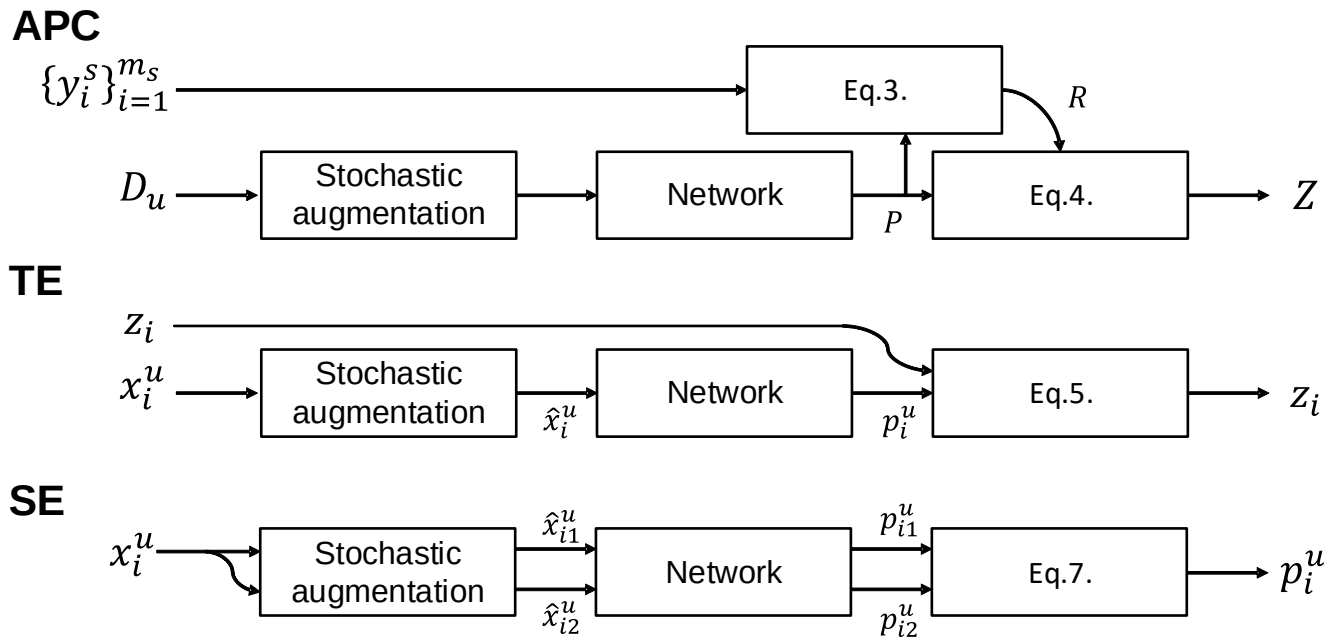


Figure 3: Structure of the training pass in HCRPL. Top: APC. Middle: TE. Bottom: SE. the details of Eq.3., Eq.4., Eq.5. and Eq.7. are show in Equal 3, 4, 5 and 7.

i-th dimension being the difficulty degree belonging to i-th class. Finally, we calibrate predictions P by

$$P \leftarrow \{\text{Normalization}(R \odot p_i^u)\}_{i=1}^{m_u}, \tag{4}$$

where $\text{Normalization}(x) = \frac{x}{\sum_i x_i}$ and $\odot$ means element-wise multiplication. Intuitively, we calibrate P by R. For a certain class c, if the predictive probability of class c is small, which means that class c is the hard class, the APC will increase the probabilities of classifying target samples into class c.

4.3. Temporal-Ensembling and Self-Ensembling

To ensure the reliability of predictions, we further introduce two ensembling methods, TE and SE. For TE, integrating predictions of multiple classifiers is considerable to obtain consistent predictions. Different from ATDA [51] that constructed two classifiers with multiview loss, we adopt the temporal-ensembling based method, which views the trained model after each epoch as a classifier. Hence, there is no need to construct multiple classifiers, which saves the number of parameters and avoids the multiview loss. As shown in Figure 3 middle, we evaluate the model on the target domain after every epoch and update ensemble predictions z_i by Exponential Moving Average (EMA). The EMA can be formulated as

$$z_i \leftarrow \alpha z_i + (1-\alpha) p_i^u. \tag{5}$$

The EMA can memorize all predictions and place a greater weight on the recent ones. α is the EMA momentum, and recent predictions will have a higher proportion with a lower α. If $\alpha = 0$, ensemble predictions z_i are equal to current ones p_i^u. Specifically, ensemble predictions z_i are used to select pseudo-labeled samples in the selecting phase.

For SE, we integrate the predictions of two different augmentations. As shown in Figure 3 bottom, we feed target samples into the trained model twice with different stochastic augmentations and obtain predictions p_{i1}^u and p_{i2}^u. Then, the average predictions $\frac{P_1+P_2}{2}$ are calculated. To obtain lower entropy predictions, we perform an additional step, named Sharpening [4, 3]. It is defined as:

$$\text{Sharpen}(p, T) = \text{Normalization}(p^{1/T}), \tag{6}$$

where T is named as the sharpening temperature. When T is smaller, predictions with lower entropy are obtained. Finally, we obtain predictions P by

$$P = \text{Sharpen}(\frac{P_1 + P_2}{2}, T). \tag{7}$$

TE and SE are somewhat similar to Π-model [28] and Mean Teacher [14, 61]. These methods took the predictive difference of different epochs and augmentations as a regularization term to constrain the model. Here, we consider that the distributional mismatch and the deviation magnification result in the vulnerability of predictions. Hence, it is adequately necessary to introduce TE and SE to stabilize the predictions.

4.4. Overall Training Process

The details of training, predicting, and selecting phases are described below.

In the training phase, the network is trained with the training dataset D_{tr}. In the first round, we view the source

Algorithm 2 Details of the prediction process

Require: ensemble predictions shadow value Z, prior class proportion $q(y)$, the target domain $\mathcal{D}_u$. network parameter θ, EMA momentum α.

Ensure: updated ensemble predictions shadow value Z.

1: Let $P_1 = \emptyset, P_2 = \emptyset$.
2: **for** $i = 1$ **to** m_u **do**
3: $\quad \hat{x}^u_{i1} = \text{Augment}\ (x^u_i)$
4: $\quad \hat{x}^u_{i2} = \text{Augment}\ (x^u_i)$
5: $\quad P_1 \leftarrow P_1 + p^u_{i1}$ ▷ $p^u_{i1} = p(y|\hat{x}^u_{i1}, \theta)$
6: $\quad P_2 \leftarrow P_2 + p^u_{i2}$ ▷ $p^u_{i2} = p(y|\hat{x}^u_{i2}, \theta)$
7: **end for**
8: $p(y) = \frac{1}{2m_u}(\sum_{i=1}^{m_u}(p^u_{i1} + p^u_{i2}))$
9: $R = q(y) \oslash p(y)$
10: $P_1 \leftarrow \{\text{Normalization}(R \otimes p^u_{i1})\}_{i=1}^{m_u}$
11: $P_2 \leftarrow \{\text{Normalization}(R \otimes p^u_{i2})\}_{i=1}^{m_u}$
12: $P = \text{Sharpen}(\frac{P_1+P_2}{2}, T)$
13: $Z \leftarrow \alpha Z + (1-\alpha)P$
14: **return** Z

domain $\mathcal{D}_s$ as training dataset D_{tr}. Subsequently, the training dataset is D_{tr} updated with the union of the pseudo-labeled target set $\mathcal{D}_l$ and source domain $\mathcal{D}_s$. The training objective is defined as

$$L = \frac{1}{m_s + m_t}(\sum_{i=1}^{m_s} \text{H}(y^s_i, p^s_i) + \sum_{i=1}^{m_t} \text{H}(\hat{y}^u_i, p^u_i)), \tag{8}$$

where $\text{H}(p, q)$ is a standard cross-entropy loss function. With the number of pseudo-labeled target samples increasing, the network learns more target-discriminative representations and gradually enhances the transfer performance on the target domain.

In the predicting phase, we aim to achieve accurate and robust predictions, especially for hard classes. We propose the APC, TE, and SE to adjust the predictions of target samples. The overall workflow pseudo-code for the predicting phase is given in Algorithm 2. The predicting phase can be split into five parts: Firstly, the target set is augmented twice and the corresponding predictions P_1, P_2 are obtained (Line 2-7). Secondly, the difficulty ratio R of prior class proportion $q(y)$ to predictive class distribution $p(y)$ is calculated (Line 8-9). Thirdly, the APC is applied to calibrate predictions P_1, P_2 (Line 10-11). Fourthly, average predictions $\frac{P_1+P_2}{2}$ are calculated (Line 12). Finally, ensemble predictions Z are updated by EMA (Line 13).

In the selecting phase, we select the predictions with higher confidence as pseudo labels. CBST [76] considers that different classes should have different confidence thresholds and dynamically adjusts thresholds to generate pseudo labels from easy to hard. The class-balanced self-training solver can be formulated as

$$\hat{y}^u_{ic} = \begin{cases} 1, \text{if } c = \underset{c}{\text{argmax}} \dfrac{z^u_{ic}}{\exp(-k_c)}, \\ \dfrac{z^u_{ic}}{\exp(-k_c)} > 1. \\ 0, \text{otherwise.} \end{cases} \tag{9}$$

Here, z^u_i means the ensemble predictions of i-th sample. z^u_{ic} means the c-th class probability for z^u_i. k_c for each class c is determined by a single portion parameter p which controls the number of selected samples. Practically, p gradually increases to select more pseudo-labeled samples. For a detailed algorithm, we recommend reading Algorithm 2 in Zou et al's paper [76].

5. Experiments

5.1. Datasets

5.1.1. ImageCLEF-DA

ImageCLEF-DA[1] is a benchmark dataset for ImageCLEF 2014 DA challenges. Three domains, including Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC (P), share 12 categories. Each domain contains 600 images and 50 images for each category.

5.1.2. Office-31

Office-31 [49] is a standard benchmark for DA tasks. The dataset contains 4110 images of 31 categories collected from three domains: Amazon (A), Webcam (W), and Dslr (D). Under the SSDA setting, we followed the settings used by Saito et al. [50] and then evaluated the proposed method on the task between W → A and D → A.

5.1.3. Office+Caltech

Office+Caltech [18] is a common benchmark that consists of four domains: Caltech, Amazon, Webcam, and Dslr. These four domains share images from ten categories. There are 2533 images in total.

5.1.4. Office-Home

Office-Home [63] is a more challenging DA dataset compared to Office-31. It consists of 15500 images from 65 categories and is organized into four domains: Art (Ar), Clipart (Cl), Product (Pr), and Real-world (Rw).

We show examples of four datasets in Figure 4. We can see different classes have different domain shifts. For example, in Office-Home, the examples of class 'Alarm-Clock' from different domains are similar to each other whereas the examples of class 'Backpack' vary a lot.

5.2. Baselines

5.2.1. Unsupervised Domain Adaptation

For Office+Caltech, we set AlexNet [27] as backbone and compare our approach with traditional and deep methods, such as SA [13], GFK [18], TCA [42], SCA [17], LPJT

[1] http://imageclef.org/2014/adaptation

Figure 4: Example images in ImageCLEF-DA, Office-31, Office+Caltech, and Office-Home

Table 1
ResNet50-based approaches on Office-Home under the UDA setting (%)

Method	Ar Cl	Ar Pr	Ar Rw	Cl Ar	Cl Pr	Cl Rw	Pr Ar	Pr Cl	Pr Rw	Rw Ar	Rw Cl	Rw Pr	Avg
ResNet50 [22]	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
RevGrad [15]	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
CBST [76]	51.4	74.1	78.9	56.3	72.2	73.4	54.4	41.6	78.8	66.0	48.3	81.0	64.7
CDAN+E [38]	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
AADA+CCN [69]	54.0	71.3	77.5	60.8	70.8	71.2	59.1	51.8	76.9	71.0	57.4	81.8	67.0
SAFN [67]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
SymNets [73]	47.7	72.9	78.5	64.2	71.3	74.2	**64.2**	48.8	79.5	**74.5**	52.6	82.7	67.6
ATM [32]	52.4	72.6	78.0	61.1	72.0	72.6	59.5	52.0	79.1	73.3	58.9	83.4	67.9
MDD [72]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	**53.6**	78.1	72.5	60.2	82.3	68.1
GSDA [23]	**61.3**	76.1	79.4	65.4	73.3	74.3	65.0	53.2	80.0	72.2	**60.6**	83.1	70.3
HCRPL (proposed)	59.5	**76.8**	**80.8**	**67.2**	**76.7**	**78.9**	63.2	53.1	**81.2**	72.3	57.2	**84.4**	**70.9**

[24], KJDIP-rbf [10], FSDA [58], MMD-CORAL [48], and GKE [65]. For the remaining datasets, we evaluated HCRPL with **ResNet-50** [22]. We compared the proposed HCRPL with the latest methods, including Reverse Gradient (**RevGrad**) [15], Joint Adaptation Networks (**JAN**) [39], Class-Balanced Self-Training (**CBST**) [76], Confidence Regularized Self-Training (**CRST**) [75], **CDAN** [38], **SAFN**[67], Domain-Symmetric Networks (**SymNets**) [73], Domain adversarial neural network (**DANN**) [16], Discriminative Adversarial Domain Adaptation (**DADA**) [60], **MDD** [72], Maximum Classifier Discrepancy (**MCD**) [53] and Cycle-consistent Conditional Adversarial Transfer Networks (**3CATN**) [31].

5.2.2. Semi-Supervised Domain Adaptation

In the SSDA experiments, we evaluated the proposed model with **AlexNet** [27] and **VGG16** [55]. The proposed model was compared with Domain adversarial neural network (**DANN**) [16], **CDAN** [38], **ENT** [20], Adversarial dropout regularization (**ADR**) [52], and Minimax Entropy (**MME**) [50].

5.3. Implementation Details

The proposed HCRPL was implemented with PyTorch[2]. For a fair comparison, our backbone network was the same as the competitive methods. For AlexNet, we replace the last full-connected layer with a randomly initialized bottleneck layer which consists of two fully-connected layers: $4096 \rightarrow C$. For VGG and ResNet, we replace the last full-connected layer with a randomly initialized C-way classifier layer. It was pre-trained on ImageNet and then fine-tuned using SGD with a weight decay of 5×10^{-4}, the momentum of 0.9, and the batch size of 32. Likewise, we used horizontal-flipping and random-cropping based data augmentation for all the training data. For the pseudo-labeling setting, we set the total number of pseudo-labeling rounds to be 30, each with 20 epochs. In the r-th round, we choose the top $p\%$ of the highest confidence target samples within each category, and $p = min(r * 5 + 10, 90)$. In the first round, the network was trained with labeled data (the source domain in UDA; the source domain and labeled target data in SSDA)

[2]https://pytorch.org/

Table 2
ResNet50-based approaches on Office-31 under the UDA setting (%)

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet50 [22]	68.4±0.2	96.7±0.2	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DANN [16]	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
CBST [76]	87.8±0.8	98.5±0.1	**100.0**±0.0	86.5±1.0	71.2±0.4	70.9±0.7	85.8
MCD [53]	89.6±0.2	98.5±0.1	**100.0**±0.0	91.3±0.2	69.6±0.3	70.8±0.3	86.6
CRST [75]	89.4±0.7	98.9±0.4	**100.0**±0.0	88.7±0.8	72.6±0.7	70.9±0.5	86.8
SAFN+ENT [67]	90.1±0.8	98.6±0.2	99.8±0.0	90.7±0.5	73.0±0.2	70.2±0.3	87.1
CDAN+E [38]	94.1±0.1	98.6±0.1	**100.0**±0.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
SymNets [73]	90.8±0.1	98.8±0.3	**100.0**±0.0	93.9±0.5	74.6±0.6	72.5±0.5	88.4
MDD [72]	94.5±0.3	98.4±0.1	**100.0**±0.0	93.5±0.2	74.6±0.3	72.2±0.1	88.9
3CATN [31]	95.3±0.3	**99.3**±0.5	**100.0**±0.0	94.1±0.3	73.1±0.2	71.5±0.6	88.9
DADA [60]	92.3±0.1	99.2±0.1	**100.0**±0.0	93.9±0.2	74.4±0.1	74.2±0.1	89.0
GSDA [23]	95.7	99.1	**100.0**	94.8	73.5	74.9	89.7
ATM [32]	95.7±0.2	**99.3**±0.1	**100.0** ± 0.0	**96.4** ± 0.2	74.1 ± 0.2	73.5 ± 0.3	89.8
HCRPL (proposed)	**95.9**±0.2	98.7±0.1	**100.0**±0.0	94.3±0.2	**75.0**±0.4	**75.4**±0.4	**89.9**

Table 3
Shallow and deep approaches on Office+Caltech under the UDA setting (%)

S → T	SA [13]	GFK [18]	TCA [42]	SCA [17]	LPJT [24]	FSDA [58]	KJDIP-rbf [10]	AlexNet [27]	MMD-CORAL [48]	GKE [65]	HCRPL
	Shallow							Deep			
A→ C	80.1	76.9	74.2	78.8	85.4	88.3	85.8	84.6	**89.1**	88.4	**89.1**
A→ D	78.3	79.6	78.3	85.4	-	87.9	87.9	88.5	96.6	**99.7**	95.8
A→ W	68.8	68.5	71.9	75.9	92.2	82.7	91.2	83.1	95.7	**97.6**	95.9
C→ A	89.5	88.4	89.3	89.5	92.1	92.8	92.4	91.8	93.6	93.5	**94.0**
C→ D	83.4	84.6	83.4	87.9	-	91.1	90.4	89.0	93.4	94.3	**98.3**
C→ W	75.9	80.7	80.0	85.4	92.7	88.8	89.5	83.1	95.2	98.3	**98.8**
D→ A	82.7	85.8	88.2	90.0	-	89.3	89.4	89.3	94.7	93.5	**94.8**
D→ C	75.7	74.1	73.5	78.1	-	80.0	78.5	80.9	84.7	83.8	**89.2**
D→ W	99.3	98.6	97.3	98.6	-	98.0	97.6	97.7	99.4	**99.7**	99.6
W→ A	77.8	75.3	80.0	86.1	92.3	87.6	92.1	83.8	**94.8**	94.4	94.6
W→ C	74.9	74.8	72.6	74.8	86.0	80.1	83.5	77.7	86.5	88.9	**88.9**
W→ D	**100.0**	**100.0**	**100.0**	**100.0**	-	99.4	96.8	**100.0**	**100.0**	**100.0**	**100.0**
Avg	82.2	82.4	82.4	85.9	-	88.8	89.6	87.5	93.6	94.3	**94.9**

with a learning rate of 5×10^{-5} or 1×10^{-4}. Furthermore, it was then retrained in the subsequent rounds with a learning rate of 1.5×10^{-5}. We set the EMA momentum α to 0.95. Under the SSDA setting, we added a few-shot module to AlexNet and VGG16 like Saito et al. [50] for better comparison with MME.

5.4. Results

5.4.1. Unsupervised domain adaptation

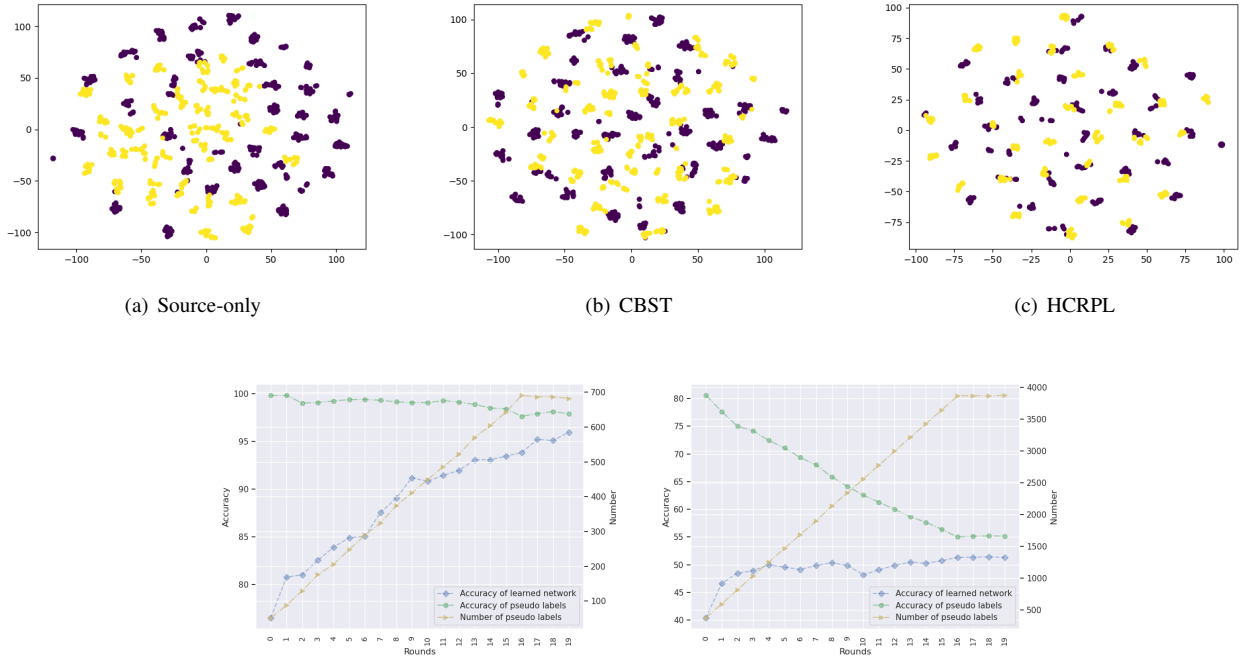
Transfer performances on Office-Home, Office-31, Office+Caltech, and ImageCLEF-DA datasets under the UDA setting are shown in Table 1, 2, 3, and 4, respectively. For Office-Home, the result is shown in Table 1, and HCRPL outperforms the best performance by 2.8% on average and achieves state-of-the-art performance on most tasks. It should be noted that the proposed framework has a larger improvement on the transfer tasks with larger domain shifts. For example, tasks Ar → Cl and Cl → Ar have poor generalization ability on the target domain, and HCRPL has an improvement over MDD by 5.4% on Ar → Cl and 7.2% on Cl → Ar. For Office-31, we report the averages and standard deviations of evaluation results over 3 runs. HCRPL outperforms all the other methods on $A \rightarrow W$, $A \rightarrow D$, $D \rightarrow A$, $D \rightarrow A$, and the average of all sub-tasks. For Office+Caltech, except for AlexNet based deep methods, our approach also compares with the shallow methods, which replace the image input with features extracted by Decaf [12] or VGG [55]. In general, deep methods are better than shallow ones, and our approach further outperforms some existing deep methods. For ImageCLEF-DA, HCRPL outperforms all the other methods on $I \rightarrow P$, $P \rightarrow I$, $C \rightarrow I$, $C \rightarrow P$, $P \rightarrow C$, and the average of all sub-tasks.

5.4.2. Semi-supervised domain adaptation

We show results on Office-31 and Office-Home under the SSDA setting. As shown in Table 5 and 9, the proposed HCRPL achieves the state-of-the-art performance on all settings (e.g., different networks, different labeled target sample size) and sub-tasks. Specifically, on Office-31 1-shot, HCRPL outperforms MME by 5.8% using AlexNet. As a reference, MME outperforms S+T by 6.3% under the same setting. Training with AlexNet is more challenging than VGG, but HCRPL improves more on AlexNet, which also proves the effectiveness of HCRPL in challenging scenarios. Similarly, under the SSDA setting, HCRPL has a

Table 4
Results on ImageCLEF-DA dataset under the UDA setting(%)

Method	I → P	P → I	I → C	C → I	C → P	P → C	Avg
ResNet50 [22]	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DANN [16]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
MCD [53]	77.3	89.2	92.7	88.2	71.0	92.3	85.1
JAN [39]	76.8	88.0	94.7	89.5	74.2	91.7	85.8
CBST [76]	77.8	91.7	96.2	91.1	75.0	93.9	87.6
CDAN+E [38]	77.7	90.7	**97.7**	91.3	74.2	94.3	87.7
SAFN [67]	78.0	91.7	96.2	91.1	77.0	94.7	88.1
AADA+CCN [69]	**79.2**	92.5	96.2	91.4	76.1	94.7	88.4
HCRPL	78.2	**92.9**	96.6	**92.3**	**77.5**	**95.8**	**88.9**



(a) Source-only (b) CBST (c) HCRPL

(d) Office-31 Amazon→Webcam (e) Office-Home Product→Clipart

Figure 5: (a)-(c): T-SNE visualization of features generated by Source-only, CBST, and HCRPL (purple: source, yellow: target). The result is obtained on Office-31 A → W under the UDA setting using ResNet-50. (d)-(e): Comparison of the actual accuracy of pseudo labels and learned network accuracy during training.

larger improvement on the tasks with larger domain shifts.

5.4.3. Comparisons with CBST

Tables 1, 2, 4, 5 and 9 also provide the results of CBST on different tasks. HCRPL not only outperforms on all tasks but also improves performance by a large margin compared to CBST, which implies that HCRPL promotes overall compared to CBST. Specifically, we discover that performance improvements are more visible on hard tasks (for example, A → D in Office-31), which illustrates that the hard class problem will further deteriorate the performance of pseudo-labeling in the case of large domain shift, and the proposed HCRPL could alleviate it.

5.5. Ablation Study

We conduct ablation studies under four different settings. The results are shown in Table 6. As shown, the APC module plays the most important role in HCRPL. The large performance degradation without APC indicates that calibrating the predictions of target samples is effective to improve transfer performance. Besides, SE and TE could further improve performance by improving the reliability of predictions. Meanwhile, it is found that the accuracy of pseudo labels varies similarly without SE or TE, which illustrates that they perform similar roles. Moreover, the performance of CBST is much lower than the proposed methods, which proves that the hard class problem deteriorates transfer performance dramatically and the proposed schemes are

Table 5
Results on the Office-31 datasets under the SSDA setting(%)

Network	Method	W → A		D → A	
		1-shot	3-shot	1-shot	3-shot
AlexNet	S+T	50.4	61.2	50.0	62.4
	DANN [16]	57.0	64.4	54.5	65.2
	ADR [52]	50.2	61.2	50.9	61.4
	CDAN [38]	50.4	60.3	48.5	61.4
	ENT [20]	50.7	64.0	50.0	66.2
	MME [50]	57.2	67.3	55.8	67.8
	CBST [76]	57.5	66.0	54.8	63.9
	HCRPL	**63.2**	**69.9**	**61.4**	**70.0**
VGG	S+T	57.4	62.9	68.7	73.3
	ENT [20]	51.6	64.8	70.6	75.3
	CDAN [38]	55.8	61.8	65.9	72.9
	ADR [52]	57.4	63.0	69.4	73.7
	DANN [16]	60.0	63.9	69.8	75.0
	MME [50]	62.7	67.6	73.4	77.0
	CBST [76]	71.4	76.6	70.8	76.2
	HCRPL	**74.6**	**77.2**	**74.0**	**77.8**

Table 6
Ablation study under four settings. U and S denote UDA and SSDA, respectively; and R and A denote ResNet50 and AlexNet, respectively. 1 means 1-shot. w/o means without.

Methods	Cl→Ar UR	A→W UR	Rw→Cl SA1	D→A SV1
CBST	56.5	87.0	39.7	62.4
HCRPL w/o APC	62.3	88.6	40.4	65.1
HCRPL w/o SE	66.6	92.5	45.3	68.8
HCRPL w/o TE	66.2	93.1	44.7	69.1
HCRPL (full)	**67.2**	**95.9**	**46.0**	**70.0**

effective.

5.6. Analysis

5.6.1. The effect of different prior class proportions.

We respectively set marginal class proportion of the source and target domains as prior class proportion and report the results in Table 7. The overall transfer accuracy of the former is only 0.4% lower than the latter, which supports that adopting the former to calibrate predictions is reasonable when the latter is unknown.

5.6.2. Pseudo-labeling Accuracy.

We report the accuracy of pseudo labels and learned network during training on Office-31 A→W and Office-Home Pr→Cl under the UDA setting in Figure 5(d) and 5(e), respectively. We found that (1) As training processes, test accuracy increases steadily, which illuminates the stability of our approach, and hence it can adapt to various scenarios better. (2) Test accuracy maintains a tight relationship with accuracy and the number of pseudo labels. In Office-31 A→W, the accuracy of pseudo labels keeps stable, and the number increases steadily. Meanwhile, the test accuracy can keep step with the number of pseudo labels. Office-Home Pr→Cl is a quite challenging task, and our approach also can extract positive information and improve test accuracy in the process of pseudo-labeling.

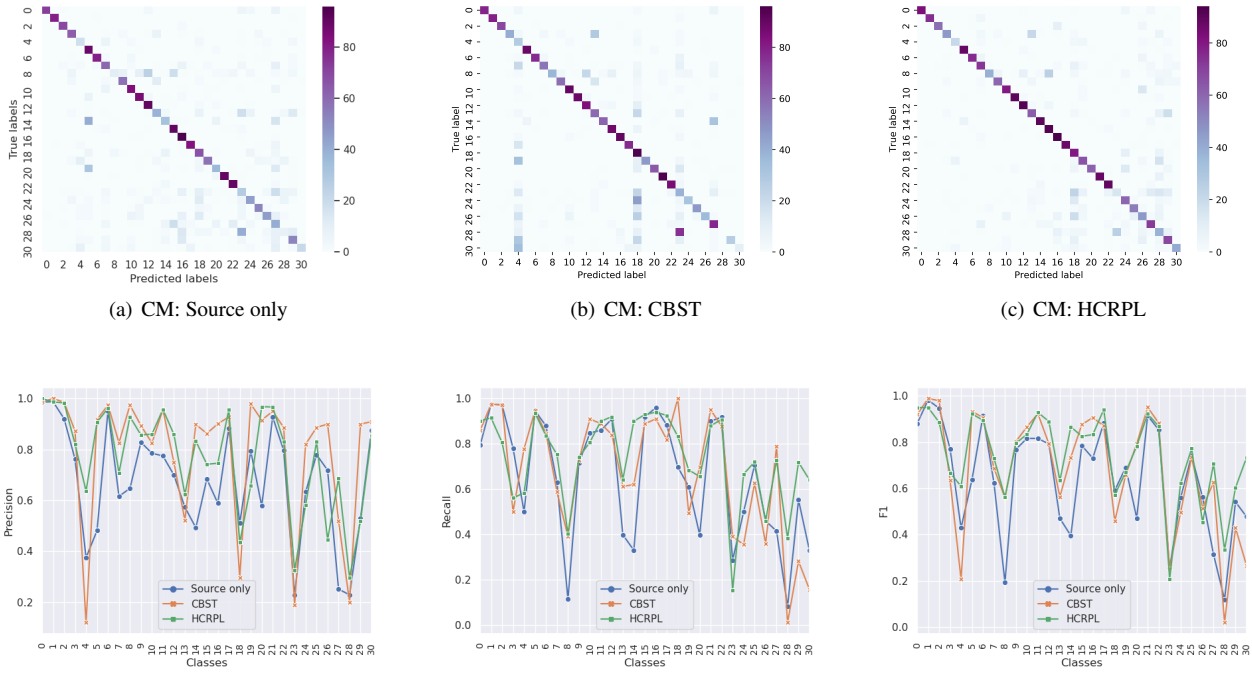
5.6.3. Exploring the hard classes.

We report confusion matrix, precision, recall, and f1-score of source only, CBST, and our approach in Figure 6. We first compare the results of Source only and CBST. Although precision, recall, and f1-score of majority classes are better to a certain extent, the worse performances among all classes may deteriorate further. Here, we focus on the performance of 28-th class in the confusion matrix. The majority of samples belonging to this class are misclassified into the 23-th class in the case of training on the source domain only. After pseudo-labeling, predictions more center on the 23-th class, which results from the misguidance of false predictions with high confidence. The results of precision, recall, and f1-score also confirm this conclusion. Furthermore, we found that the predictive class proportion of easy classes, such as 4-th and 18-th classes, are higher after applying pseudo-labeling (shown in Figures 6(a) and 6(b)), which results in the decline of precision for these classes, which also is the drawback of pseudo-labeling.

We further analyze the performance of our HCRPL from the aspect of hard classes. First, the HCRPL avoids predictions degenerate into few classes, which illustrates the HCRPL can control the predictive class proportion of each class in a reasonable interval and improve the precision of the easy classes. Second, we found the precision of majority classes, is better than CBST and source only. The recall is flat or slightly better than CBST and source only. Overall, class-level performance exceeds CBST and source only. Especially for hard classes, such as 28-th, and 30-th classes, the performance of HCRPL is superior apparently, which demonstrates that our approach can indeed alleviate the hard class problem.

5.6.4. Feature Visualization.

We train ResNet-50, CBST, and the proposed HCRPL on Office-31 A → W under the UDA setting and plot the learned features with t-SNE [40] in Figure 5 (a-c), respectively. The purple and yellow points represent the learned features of the source domain and target domains, respectively. As mentioned in section 1, pseudo-labeling is a promising paradigm for aligning and separating the class conditional distributions of various domains. Therefore, CBST and HCRPL could promote learning target-discriminative representations and aligning class conditional distributions. Furthermore, HCRPL could learn more target-discriminative representations due to improving the accuracy of pseudo labels.



(a) CM: Source only (b) CM: CBST (c) CM: HCRPL

(d) Precision (e) Recall (f) F1-score

Figure 6: (a)-(c) Confusion Matrix (CM) visualization for Source only, CBST, and HCRPL. (d)-(f): Precision, recall, and f1-score evaluated on three different models, Source-only, CBST, and HCRPL. The result is obtained on Office-31 W → A under the UDA setting using ResNet-50. To better visualize results, we arrange the categories in alphabetic order.

Table 7
Comparison of different prior class proportion on Office-31 under the UDA setting (%). S denotes the marginal class proportion of the source domain. T denotes the marginal class proportion of the target domain.

prior class proportion	A → W	D → W	W → D	A → D	D → A	W → A	Avg
S	95.9±0.2	98.7±0.1	**100.0**±0.0	94.3±0.2	75.0±0.4	75.4±0.4	89.9
T	**96.3**±0.1	**98.9**±0.1	**100.0**±0.0	**94.9**±0.2	**75.5**±0.3	**76.2**±0.3	**90.3**

5.6.5. Convergence.

The convergence of ResNet, CBST, and the proposed HCRPL can be demonstrated by the error rates in the target domain on Office-31 A → W under the UDA setting. As shown in Figure 7, the following observations can be made: 1) CBST and HCRPL are more stable than ResNet (baseline) due to alleviating overfitting to the source domain. 2) The target error decreases gradually learned by CBST and HCRPL because the pseudo-labeling methods progressively generate more pseudo labels and learn more of the target-discriminative representations as the training progresses. 3) The disparity of the target error between CBST and HCRPL increases gradually, which indicates that APC, SE, and TE effectively improve the accuracy of pseudo labels.

5.6.6. Hyperparameter Sensitivity

We then investigate the sensitivity of HCRPL for different choices of hyperparameters: EMA momentum α and sharpening temperature T. The results are reported in Table 8. For EMA momentum α, transfer accuracy fluctuates slightly when $\alpha \in [0.9, 0.97]$. However, when the value of α does not belong to this range, transfer accuracy degrades a lot. For sharpening temperature T, the smaller T achieves better performance than the larger one, which means the predictions with lower entropy are more accurate. Meanwhile, the minimal value of T also degrades performance.

Table 8
Comparison of different EMA momentum α and sharpening temperature T on A→W.

α	0.0	0.8	0.9	0.95	0.97	0.99
Acc (%)	92.5	93.1	95.2	**95.9**	94.5	93.1
T	0.2	0.3	0.5	0.7	0.8	1.0
Acc (%)	94.7	95.7	**95.9**	93.8	93.8	92.3

Table 9
Results on Office-Home dataset under the SSDA setting(%)

Network	Method	Ar Cl	Ar Pr	Ar Rw	Cl Ar	Cl Pr	Cl Rw	Pr Ar	Pr Cl	Pr Rw	Rw Ar	Rw Cl	Rw Pr	Avg
							One-shot							
AlexNet	S+T	37.5	63.1	44.8	54.3	31.7	31.5	48.8	31.1	53.3	48.5	33.9	50.8	44.1
	DANN [16]	42.5	64.2	45.1	56.4	36.6	32.7	43.5	34.4	51.9	51.0	33.8	49.4	45.1
	ADR [52]	37.8	63.5	45.4	53.5	32.5	32.2	49.5	31.8	53.4	49.7	34.2	50.4	44.5
	CDAN [38]	36.1	62.3	42.2	52.7	28.0	27.8	48.7	28.0	51.3	41.0	26.8	49.9	41.2
	ENT [20]	26.8	65.8	45.8	56.3	23.5	21.9	47.4	22.1	53.4	30.8	18.1	53.6	38.8
	MME [50]	42.0	69.6	48.3	58.7	37.8	34.9	52.5	36.4	57.0	54.1	39.5	59.1	49.2
	CBST [76]	39.7	69.1	46.0	59.3	31.6	33.7	54.6	32.5	57.3	53.2	36.0	57.4	47.5
	HCRPL	**46.0**	**73.3**	**48.8**	**63.8**	**38.7**	**38.4**	**58.1**	**37.4**	**59.7**	**61.0**	**40.1**	**62.2**	**52.3**
VGG	S+T	39.5	75.3	61.2	71.6	37.0	52.0	63.6	37.5	69.5	64.5	51.4	65.9	57.4
	DANN [16]	52.0	75.7	62.7	72.7	45.9	51.3	64.3	44.4	68.9	64.2	52.3	65.3	60.0
	ADR [52]	39.7	76.2	60.2	71.8	37.2	51.4	63.9	39.0	68.7	64.8	50.0	65.2	57.4
	CDAN [38]	43.3	75.7	60.9	69.6	37.4	44.5	67.7	39.8	64.8	58.7	41.6	66.2	55.8
	ENT [20]	23.7	77.5	64.0	74.6	21.3	44.6	66.0	22.4	70.6	62.1	25.1	67.7	51.6
	MME [50]	49.1	78.7	65.1	74.4	46.2	56.0	68.6	45.8	72.2	68.0	57.5	71.3	62.7
	CBST [76]	42.2	78.9	62.2	75.0	39.5	52.8	70.6	40.4	73.2	68.8	54.1	70.7	60.7
	HCRPL	**53.1**	**82.0**	**66.3**	**77.1**	**49.0**	**57.8**	**76.3**	**47.4**	**75.6**	**73.5**	**58.3**	**73.8**	**65.9**
							Three-shot							
AlexNet	S+T	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57.0	54.3	37.5	57.9	50.0
	DANN [16]	47.2	66.7	46.6	58.1	44.4	36.1	57.2	39.8	56.6	54.3	38.6	57.9	50.3
	ADR [52]	45.0	66.2	46.9	57.3	38.9	36.3	57.5	40.0	57.8	53.4	37.3	57.7	49.5
	CDAN [38]	41.8	69.9	43.2	53.6	35.8	32.0	56.3	34.5	53.5	49.3	27.9	56.2	46.2
	ENT [20]	44.9	70.4	47.1	60.3	41.2	34.6	60.7	37.8	60.5	58.0	31.8	63.4	50.9
	MME [50]	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
	CBST [76]	45.3	72.6	48.4	62.3	40.3	37.3	64.2	40.7	61.6	58.6	39.9	62.6	52.8
	HCRPL	**53.5**	**75.0**	**51.4**	**64.9**	**46.0**	**42.4**	**65.4**	**45.8**	**63.4**	**63.1**	**41.9**	**67.6**	**56.7**
VGG	S+T	49.6	78.6	63.6	72.7	47.2	55.9	69.4	47.5	73.4	69.7	56.2	70.4	62.9
	DANN [16]	56.1	77.9	63.7	73.6	52.4	56.3	69.5	50.0	72.3	68.7	56.4	69.8	63.9
	ADR [52]	49.0	78.1	62.8	73.6	47.8	55.8	69.9	49.3	73.3	69.3	56.3	71.4	63.0
	CDAN [38]	50.2	80.9	62.1	70.8	45.1	50.3	74.7	46.0	71.4	65.9	52.9	71.2	61.8
	ENT [20]	48.3	81.6	65.5	76.6	46.8	56.9	73.0	44.8	75.3	72.9	59.1	77.0	64.8
	MME [50]	56.9	82.9	65.7	76.7	53.6	59.2	75.7	54.9	75.3	72.9	61.1	76.3	67.6
	MME [50]	56.9	82.9	65.7	76.7	53.6	59.2	75.7	54.9	75.3	72.9	61.1	76.3	67.6
	CBST [76]	52.2	81.6	64.8	75.5	48.9	55.7	75.1	51.4	76.3	72.0	57.5	74.7	65.5
	HCRPL	**59.7**	**84.7**	**68.7**	**78.5**	**55.3**	**61.7**	**77.6**	**55.5**	**78.6**	**76.1**	**62.2**	**79.0**	**69.8**

6. Discussion and Conclusion

6.1. Strength

The major contribution of this paper is unraveling the hard class problem, which is always ignored in the existing pseudo-labeling methods yet is crucial in some practical scenarios. Compared with the existing pseudo-labeling methods, our approach improves not only overall performance but only the worst performance among all classes.

6.2. Weakness

The introduction of prior knowledge alleviates the hard class problem effectively. However, it also limits the application of our approach to a certain extent. Our basic assumption is that the source and target domains should have a similar label proportion, which is invalid in some applications, such as partial domain adaptation [5], open set domain adaptation [44]. In such variants of DA, applying APC obviously will miscalibrate predictions and result in massive false pseudo labels. Therefore, accurately inferring the marginal class distribution of the target domain should be further studied in future jobs. On the other hand, the proposed HCRPL can effectively improve the precision of hard classes but has little impact on the recall of them, which also is important in practical applications.

6.3. Conclusion

Pseudo-labeling is a promising paradigm for solving the DA problem. In this paper, we first revealed the hard class problem may occur in DA and is harmful to the applications of DA. To alleviate the hard class problem, we proposed APC to calibrate predictions according to the diffi-

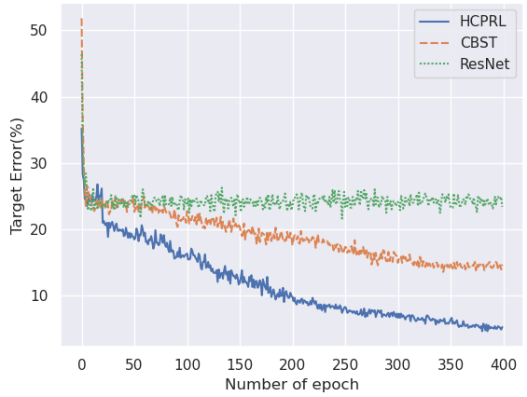


Figure 7: Test accuracy over iterations. The result is obtained on Office-31 A → W under the UDA setting using ResNet-50. ResNet means that we train the model without DA and only use the source domain as training data.

culty degree for each class. Furthermore, we introduced SE and TE to improve model robustness for target samples, especially for ones belonging to hard classes. In experiments, we demonstrated that HCRPL achieved good results and outperformed some state-of-the-art methods with considerable margins under the UDA setting. Meanwhile, HCRPL is suitable for the SSDA setting because it can learn target-discriminative representations. Experimental analysis shows that the proposed schemes can indeed alleviate the hard class problem and improve the accuracy of pseudo labels.

7. Acknowledgments

The work is supported in part by National Natural Science Foundation of China under Grants 81671766, 61971369, U19B2031 U1605252, 61671309, in part by Open Fund of Science and Technology on Automatic Target Recognition Laboratory 6142503190202, in part by Fundamental Research Funds for the Central Universities 20720180059, 20-720190116, 20720200003, in part by Tencent Open Fund.

References

[1] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Zhu, Z., 2015. Deep speech 2: End-to-end speech recognition in english and mandarin. Computer Science .

[2] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W., 2010. A theory of learning from different domains. Machine Learning 79, 151–175.

[3] Berthelot, D., Carlini, N., Cubuk, E.D., Kurakin, A., Sohn, K., Zhang, H., Raffel, C., 2019a. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv preprint arXiv:1911.09785 .

[4] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A., 2019b. Mixmatch: A holistic approach to semi-supervised learning, in: Advances in Neural Information Processing Systems, pp. 5050–5060.

[5] Cao, Z., Ma, L., Long, M., Wang, J., 2018. Partial adversarial domain adaptation, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 135–150.

[6] Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., Huang, J., 2019a. Progressive feature alignment for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 627–636.

[7] Chen, C., Zheng, Z., Ding, X., Huang, Y., Dou, Q., 2020a. Harmonizing transferability and discriminability for adapting object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8869–8878.

[8] Chen, C., Zheng, Z., Huang, Y., Ding, X., Yu, Y., 2021. I3net: Implicit instance-invariant network for adapting one-stage object detectors, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[9] Chen, D.D., Wang, Y., Yi, J., Chen, Z., Zhou, Z.H., 2019b. Joint semantic domain alignment and target classifier learning for unsupervised domain adaptation. arXiv preprint arXiv:1906.04053 .

[10] Chen, S., Harandi, M., Jin, X., Yang, X., 2020b. Domain adaptation by joint distribution invariant projections. IEEE Transactions on Image Processing 29, 8264–8277.

[11] Deng, W., Zheng, L., Jiao, J., 2018. Domain alignment with triplets. CoRR abs/1812.00893.

[12] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T., 2014. Decaf: A deep convolutional activation feature for generic visual recognition, in: ICML.

[13] Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T., 2013. Unsupervised visual domain adaptation using subspace alignment, in: 2013 IEEE International Conference on Computer Vision, pp. 2960–2967. doi:10.1109/ICCV.2013.368.

[14] French, G., Mackiewicz, M., Fisher, M.H., 2018. Self-ensembling for visual domain adaptation, in: 6th International Conference on Learning Representations, OpenReview.net.

[15] Ganin, Y., Lempitsky, V.S., 2015. Unsupervised domain adaptation by backpropagation, in: Bach, F.R., Blei, D.M. (Eds.), Proceedings of the 32nd International Conference on Machine Learning, JMLR.org. pp. 1180–1189.

[16] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S., 2016. Domain-adversarial training of neural networks. J. Mach. Learn. Res. 17, 59:1–59:35.

[17] Ghifary, M., Balduzzi, D., Kleijn, W., Zhang, M., 2017. Scatter component analysis: A unified framework for domain adaptation and domain generalization. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 1414–1430.

[18] Gong, B., Shi, Y., Sha, F., Grauman, K., 2012. Geodesic flow kernel for unsupervised domain adaptation, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE. pp. 2066–2073.

[19] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in neural information processing systems, pp. 2672–2680.

[20] Grandvalet, Y., Bengio, Y., 2005. Semi-supervised learning by entropy minimization, in: Advances in neural information processing systems, pp. 529–536.

[21] Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.J., 2012. A kernel two-sample test. J. Mach. Learn. Res. 13, 723–773.

[22] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[23] Hu, L., Kan, M., Shan, S., Chen, X., 2020. Unsupervised domain adaptation with hierarchical gradient synchronization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4043–4052.

[24] Jing-jing, L., Mengmeng, J., Ke, L., Lei, Z., Tao, S., 2019. Locality preserving joint transfer for domain adaptation. arXiv: Computer Vision and Pattern Recognition .

[25] Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G., 2019. Contrastive adaptation network for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recog-

nition, pp. 4893–4902.
[26] Kim, T.K., Kim, C., 2020. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation, in: ECCV.
[27] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Bartlett, P.L., Pereira, F.C.N., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, pp. 1106–1114.
[28] Laine, S., Aila, T., 2017. Temporal ensembling for semi-supervised learning, in: 5th International Conference on Learning Representations, ICLR, OpenReview.net.
[29] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436–444.
[30] Lee, D.H., 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on challenges in representation learning, ICML, p. 2.
[31] Li, J., Chen, E., Ding, Z., Zhu, L., Lu, K., Huang, Z., 2019a. Cycle-consistent conditional adversarial transfer networks, in: Proceedings of the 27th ACM International Conference on Multimedia, pp. 747–755.
[32] Li, J., Chen, E., Ding, Z., Zhu, L., Lu, K., Shen, H.T., 2020a. Maximum density divergence for domain adaptation. IEEE transactions on pattern analysis and machine intelligence .
[33] Li, J., Jing, M., Lu, K., Zhu, L., Shen, H.T., 2019b. Locality preserving joint transfer for domain adaptation. IEEE Transactions on Image Processing 28, 6103–6115.
[34] Li, L., Zhang, Z., 2019. Semi-supervised domain adaptation by covariance matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 41, 2724–2739.
[35] Li, T., Zhao, Z., Sun, C., Cheng, L., Chen, X., Yan, R., Gao, R., 2019c. Waveletkernelnet: An interpretable deep neural network for industrial intelligent diagnosis. ArXiv abs/1911.07925.
[36] Li, T., Zhao, Z., Sun, C., Yan, R., Chen, X., 2020b. Multi-receptive field graph convolutional networks for machine fault diagnosis. IEEE Transactions on Industrial Electronics , 1–1.
[37] Long, M., Cao, Y., Wang, J., Jordan, M.I., 2015. Learning transferable features with deep adaptation networks, in: Bach, F.R., Blei, D.M. (Eds.), Proceedings of the 32nd International Conference on Machine Learning, JMLR.org. pp. 97–105.
[38] Long, M., Cao, Z., Wang, J., Jordan, M.I., 2018. Conditional adversarial domain adaptation, in: Advances in Neural Information Processing Systems, pp. 1640–1650.
[39] Long, M., Zhu, H., Wang, J., Jordan, M.I., 2017. Deep transfer learning with joint adaptation networks, in: ICML.
[40] Maaten, L.v.d., Hinton, G., 2008. Visualizing data using t-sne. Journal of machine learning research 9, 2579–2605.
[41] Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G., 2017. Unified deep supervised domain adaptation and generalization, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 5715–5725.
[42] Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q., 2010. Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks 22, 199–210.
[43] Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345–1359.
[44] Panareda Busto, P., Gall, J., 2017. Open set domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 754–763.
[45] Purushotham, S., Carvalho, W., Nilanon, T., Liu, Y., 2017. Variational recurrent adversarial deep domain adaptation, in: 5th International Conference on Learning Representations, OpenReview.net.
[46] Qin, C., Wang, L., Ma, Q., Yin, Y., Wang, H., Fu, Y., 2020a. Opposite structure learning for semi-supervised domain adaptation. ArXiv abs/2002.02545.
[47] Qin, C., Wang, L., Ma, Q., Yin, Y., Wang, H., Fu, Y., 2020b. Opposite structure learning for semi-supervised domain adaptation. arXiv preprint arXiv:2002.02545 .
[48] Rahman, M., Fookes, C., Baktash, M., Sridharan, S., 2020. On minimum discrepancy estimation for deep domain adaptation. ArXiv abs/1901.00282.
[49] Saenko, K., Kulis, B., Fritz, M., Darrell, T., 2010. Adapting visual category models to new domains, in: ECCV, Springer.
[50] Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K., 2019. Semi-supervised domain adaptation via minimax entropy. CoRR abs/1904.06487.
[51] Saito, K., Ushiku, Y., Harada, T., 2017. Asymmetric tri-training for unsupervised domain adaptation, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org. pp. 2988–2997.
[52] Saito, K., Ushiku, Y., Harada, T., Saenko, K., 2018a. Adversarial dropout regularization, in: 6th International Conference on Learning Representationss, OpenReview.net.
[53] Saito, K., Watanabe, K., Ushiku, Y., Harada, T., 2018b. Maximum classifier discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732.
[54] Shao, R., Lan, X., Yuen, P.C., 2018. Feature constrained by pixel: Hierarchical adversarial deep domain adaptation, in: Proceedings of the 26th ACM international conference on Multimedia, pp. 220–228.
[55] Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition, in: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations.
[56] Sun, B., Feng, J., Saenko, K., 2016. Return of frustratingly easy domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence.
[57] Sun, B., Saenko, K., 2016. Deep CORAL: correlation alignment for deep domain adaptation, in: Hua, G., Jégou, H. (Eds.), Computer Vision - ECCV 2016 Workshops, pp. 443–450.
[58] Sun, F., Wu, H., Luo, Z., Gu, W., Yan, Y., Du, Q., 2019. Informative feature selection for domain adaptation. IEEE Access 7, 142551–142563.
[59] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R., 2014. Intriguing properties of neural networks, in: Bengio, Y., LeCun, Y. (Eds.), 2nd International Conference on Learning Representations.
[60] Tang, H., Jia, K., 2019. Discriminative adversarial domain adaptation. arXiv preprint arXiv:1911.12036 .
[61] Tarvainen, A., Valpola, H., 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: 5th International Conference on Learning Representations, OpenReview.net.
[62] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T., 2014. Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 .
[63] Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S., 2017. Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5018–5027.
[64] Wu, H., Yan, Y., Lin, G., Yang, M., Ng, M., Wu, Q., 2020a. Iterative refinement for multi-source visual domain adaptation. IEEE Transactions on Knowledge and Data Engineering , 1–1.
[65] Wu, H., Yan, Y., Ye, Y., Ng, M., Wu, Q., 2020b. Geometric knowledge embedding for unsupervised domain adaptation. Knowl. Based Syst. 191, 105155.
[66] Xie, S., Zheng, Z., Chen, L., Chen, C., 2018. Learning semantic representations for unsupervised domain adaptation, in: Dy, J.G., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, PMLR. pp. 5419–5428.
[67] Xu, R., Li, G., Yang, J., Lin, L., 2019. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1426–1435.
[68] Yan, Y., Li, W., Wu, H., Min, H., Tan, M., Wu, Q., 2018. Semi-supervised optimal transport for heterogeneous domain adaptation, in: IJCAI.
[69] Yang, J., Zou, H., Zhou, Y., Zeng, Z., Xie, L., 2020a. Mind the

discriminability: Asymmetric adversarial domain adaptation, in: European Conference on Computer Vision, Springer. pp. 589–606.

[70] Yang, L., Wang, Y., Gao, M., Shrivastava, A., Weinberger, K.Q., Chao, W.L., Lim, S.N., 2020b. Deep co-training with task decomposition for semi-supervised domain adaptation. arXiv: Computer Vision and Pattern Recognition .

[71] Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., a. Understanding deep learning requires rethinking generalization, in: 5th International Conference on Learning Representations.

[72] Zhang, Y., Liu, T., Long, M., Jordan, M.I., b. Bridging theory and algorithm for domain adaptation, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning.

[73] Zhang, Y., Tang, H., Jia, K., Tan, M., 2019. Domain-symmetric networks for adversarial domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5031–5040.

[74] Zhang, Y., Wei, Y., Wu, Q., Zhao, P., Niu, S., Huang, J., Tan, M., 2020. Collaborative unsupervised domain adaptation for medical image diagnosis. IEEE Transactions on Image Processing 29, 7834–7844.

[75] Zou, Y., Yu, Z., Liu, X., Kumar, B., Wang, J., 2019. Confidence regularized self-training, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 5982–5991.

[76] Zou, Y., Yu, Z., Vijaya Kumar, B., Wang, J., 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training, in: Proceedings of the European conference on computer vision (ECCV), pp. 289–305.