Class Discriminative Adversarial Learning for Unsupervised Domain Adaptation

Lihua Zhou
School of CSE, University of Electronic Science and Technology of China, Chengdu, China
lihua.zhou@std.uestc.edu.cn

Mao Ye*
School of CSE, University of Electronic Science and Technology of China, Chengdu, China
cvlab.uestc@gmail.com

Xiatian Zhu*
Surrey Institute for People-Centred Artificial Intelligence, CVSSP, University of Surrey, Guildford, UK
xiatian.zhu@surrey.ac.uk

Shuaifeng Li
School of CSE, University of Electronic Science and Technology of China, Chengdu, China

Yiguang Liu
School of Computer Science, Sichuan University, Chengdu, China
liuyg@scu.edu.cn

ABSTRACT

As a state-of-the-art family of Unsupervised Domain Adaptation (UDA), bi-classifier adversarial learning methods are formulated in an adversarial (minimax) learning framework with a single feature extractor and two classifiers. Model training alternates between two steps: (I) constraining the learning of the two classifiers to maximize the prediction discrepancy of unlabeled target domain data, and (II) constraining the learning of the feature extractor to minimize this discrepancy. Despite being an elegant formulation, this approach has a fundamental limitation: Maximizing and minimizing the classifier discrepancy is not class discriminative for the target domain, finally leading to a suboptimal adapted model. To solve this problem, we propose a novel **Class Discriminative Adversarial Learning** (CDAL) method characterized by discovering class discrimination knowledge and leveraging this knowledge to discriminatively regulate the classifier discrepancy constraints on-the-fly. This is realized by introducing an evaluation criterion for judging each classifier's capability and each target domain sample's feature reorientation via objective loss reformulation. Extensive experiments on three standard benchmarks show that our CDAL method yields new state-of-the-art performance. Our code is made available at https://github.com/buerzlh/CDAL.

CCS CONCEPTS

• **Computing methodologies** → **Transfer learning**; **Neural networks**.

KEYWORDS

Unsupervised Domain Adaptation, Bi-classifier Adversarial Learning, Class Discriminative Adversarial Learning

* Mao Ye and Xiatian Zhu are the corresponding authors.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548143

ACM Reference Format:
Lihua Zhou, Mao Ye*, Xiatian Zhu*, Shuaifeng Li, and Yiguang Liu. 2022. Class Discriminative Adversarial Learning for Unsupervised Domain Adaptation. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548143

1 INTRODUCTION

Supervised deep learning excel in various computer vision tasks [9, 15, 32, 46] with reliance on big data annotation. Nonetheless, labeling such training data for every single domain is prohibitively expensive or impossible for many scenarios. Unsupervised domain adaptation (UDA) [30] is a viable solution by transferring the knowledge from a labeled source domain to an unlabeled target domain.

Among existing UDA methods, a representative family of top-performing methods, bi-classifier adversarial learning, are formulated in an adversarial learning framework [4, 16, 18, 33], wherein the feature extractor and two classifiers form a minimax game alternating between two steps: **(1)** Optimizing the classifiers subject to maximizing the discrepancy of their predictions, with an aim to identify the target samples out of the support of source distribution; **(2)** Optimizing the feature extractor subject to minimizing this classifier discrepancy, with an aim to align the distributions across domains. While being an elegant formulation, it suffers from a fundamental limitation. Concretely, as shown in Fig.1(a), target samples out of the support of source distribution can be summarized in two types: *Easy samples* with the consistent predicted labels by two classifiers; *Ambiguous (hard) samples* with inconsistent predicted labels by two classifiers, and misclassified by at least one classifier. We observe that traditional bi-classifier adversarial learning methods tend to yield more ambiguous samples from maximizing the prediction discrepancy because it forces the predictions of two classifiers to be inconsistent. After minimizing this discrepancy, easy samples can be often matched to the correct category labels, while ambiguous samples could be dominated by misclassified classifier and matched to the wrong class [4]. Together with that the learning constraints of both steps are *not* class discriminative, the performance of such domain adapted classifiers would be limited.

For minimizing the performance drop of the adapted classifiers with bi-classifier adversarial learning, properly handling the ambiguous target samples is a key. To that end, we propose a novel **Class Discriminative Adversarial Learning** (CDAL) method. It

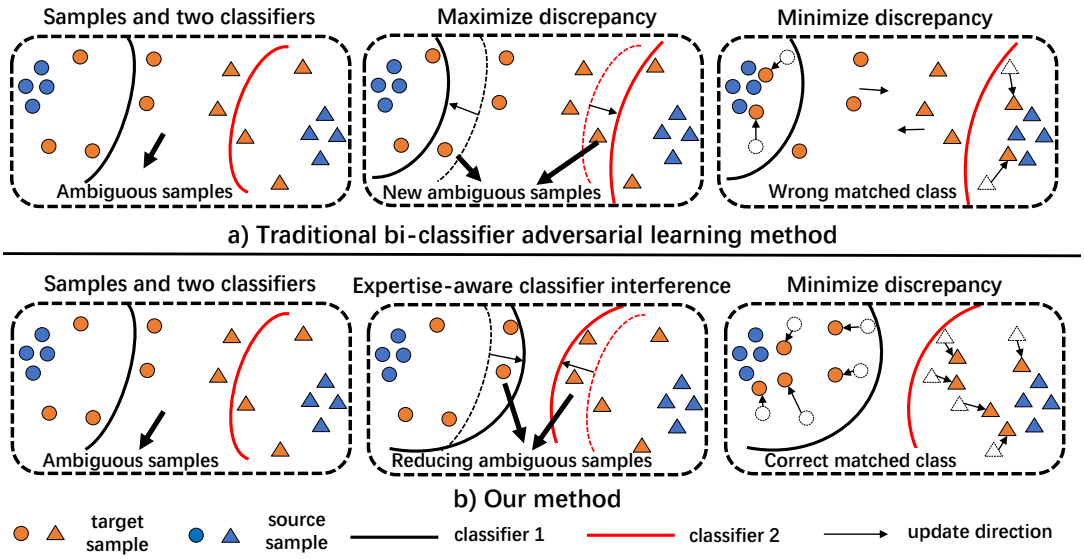

Figure 1: Illustration of (a) the traditional bi-classifier adversarial learning method *vs.* (b) our class discriminative adversarial learning (CDAL) method. Traditional bi-classifier adversarial method maximizes the prediction discrepancy when optimizing two classifiers, which yields more ambiguous target samples. Besides, ambiguous target samples may be matched to wrong classes when minimizing the prediction discrepancy in optimizing the feature extractor. To address this problem, we propose a novel CDAL framework characterized with an *Expertise-aware Classifier Interference* (ECI) strategy for more discriminative classifier optimization. It can reason away ambiguous target samples gradually, leading to superior domain alignment.

is based on an *Expertise-aware Classifier Interference* (ECI) strategy that enables the classifiers to be mutually beneficial and become increasingly discriminative during training. Concretely, with ECI we evaluate the per-sample discrimination ability of the two classifiers. Subsequently, for the classifier making a better prediction, we maximize the prediction discrepancy against the other to find the target samples out of the support of source distribution; Meanwhile, the other classifier is optimized instead by minimizing the prediction discrepancy so that it can be corrected and aligned with the better-performing classifier for more consistent and accurate predicting. As a consequence, the two classifiers improve their discriminative power progressively, resulting in less ambiguous samples. To further suppress ambiguous target samples, we design a complementary representation regularization for enhancing the learning of feature extractor with conditional distribution alignment.

Our contributions can be summarized as follows: (1) We propose a novel *Class Discriminative Adversarial Learning* (CDAL) framework for UDA. This solves the fundamental limitations of existing alternatives in relating the predictions of two classifiers without considering their discriminating capability. (2) Compared with traditional bi-classifier adversarial learning methods, CDAL can more effectively improve the discrimination ability of both adapted classifiers whilst reasoning away ambiguous target samples during training. (3) Extensive experiments show that CDAL outperforms state-of-the-art methods by a clear margin on three standard datasets.

2 RELATED WORK

Unsupervised Domain Adaptation (UDA) aims to transfer knowledge learned from a labeled source domain to an unlabeled target domain. The existing research routes of unsupervised domain adaptation methods can be roughly divided into six routes. The first route is *statistic moment matching*, which mitigates the gap between two domains by minimizing some defined statistical discrepancy metrics, such as DAN [23], CORAL [36, 48], CMD [44], CAN [12] and ETD [17] and so on. The second route applies *adversarial learning framework*, which introduces a domain discriminator for domain classification, then forces feature extractor confusing domain discriminator to learn domain invariant features. The represented methods are DANN [6], CDAN [24], and ADDA [39] and so on. The third route is based on *adversarial generation framework*, which combines the domain discriminator with a generator, and generates fake data and aligns the distribution between the two domains based on pixel-level. The methods with high attention are CoGAN [21], SimGAN [35] and CycleGAN [47]. The fourth route uses *self-training strategy*, which implicitly minimizes the discrepancy between two domains by incorporating auxiliary self-supervised learning tasks into the original task, such as MTAE [7], DRCN [8], and ssUDA [37]. The fifth route applies *ensemble learning* by learning multiple models together. The typical approaches are based on the Mean Teacher framework [1, 3, 5, 38], consisting of a student network and a teacher network.

The final strategy is called *bi-classifier adversarial learning*, which plays a minimax game with a single feature extractor and two distinct classifiers during domain adaptation. This type of methods maximize the prediction discrepancy of unlabeled target domain samples when optimizing the two classifiers and minimize this discrepancy when optimizing the feature extractor, so as to align the distribution between two domains [4, 16, 18, 33]. Specifically, MCD [33] uses $\mathcal{L}_1$ norm to calculate the prediction discrepancy,

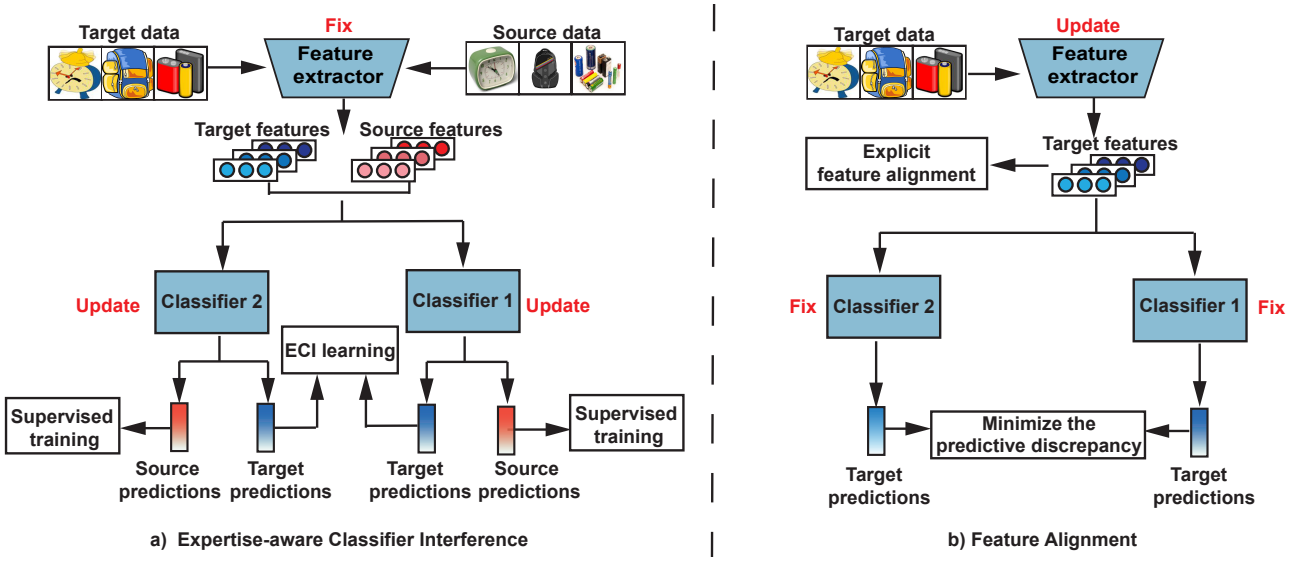

Figure 2: An illustration of our *Class Discriminative Adversarial Learning* (CDAL) method. In the first step, the model (including the feature extractor and two classifiers) is trained by labeled source samples. (a) In the second step, the feature extractor is fixed while the two classifiers are updated by the proposed *Expertise-aware Classifier Interference* (ECI) strategy. Note, the supervised training supervisory on source domain is applied to preserve the classification ability. (b) In the third step, the feature extractor is then optimized by minimizing the discrepancy between the two fixed classifiers. Feature alignment is also applied across domains.

while SWD [16] proposes slide wasserstein distance and BCDM [18] proposes classifier determinacy disparity distance. On the basis of MCD, CGDM [4] proposes gradient discrepancy minimization, which forces the feature extractor to further align the gradient discrepancy between two domains. However, neither of these methods considers the classification ability of two classifiers, leading to a suboptimal adapted model. By making two classifiers learn each other's discriminating capability, CDAL gradually makes two classifiers more class discriminative, which effectively solves the challenges caused by ambiguous target samples.

3 PRELIMINARY

Suppose that the source domain $D_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$ consists of n_s labeled samples, the target domain $D_t = \{(\boldsymbol{x}_i^t)\}_{i=1}^{n_t}$ consists of n_t unlabeled samples. The source domain and the target domain have the same label space $\{1, 2, \ldots, K\}$, but with different data distributions. Our goal is to transfer source domain knowledge to the target domain so that the adapted model can correctly classify the target samples.

Bi-classifier adversarial learning methods [16, 18, 33] usually have three steps. In the first step, the feature extractor g and two classifiers h_1 and h_2 are trained based on the labeled source samples, which makes the model fit the distribution of the source domain. The objective function is defined as

$$\min_{g,h_1,h_2} \mathcal{L}_{cls}(D_s) = \frac{1}{2}\sum_{j=1}^{2} \mathbb{E}_{\boldsymbol{x}_i^s \in D_s} \mathcal{L}^{ce}(\boldsymbol{p}_{i,j}^s, \boldsymbol{y}_i^s), \tag{1}$$

where $\mathcal{L}^{ce}(\cdot,\cdot)$ represents the cross entropy function. $\boldsymbol{p}_{i,j}^s$ means the prediction of the classifier h_j respect to the source sample $\boldsymbol{x}_i^s$.

In the second step, the feature extractor g is frozen, and two classifiers h_1 and h_2 are updated by maximizing the prediction discrepancy for unlabeled target samples to find the target samples out of the support of source distribution while minimizing the cross entropy for labeled source samples as follows,

$$\min_{h_1,h_2} \mathcal{L}_{cls}(D_s) - \mathcal{L}_{dis}(D_t), \tag{2}$$

where

$$\mathcal{L}_{dis}(\mathcal{X}) = \mathbb{E}_{\boldsymbol{x}\in\mathcal{X}} dist(\boldsymbol{p}_1, \boldsymbol{p}_2). \tag{3}$$

Here, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are predictions of two classifiers respect to the sample $\boldsymbol{x}$, respectively. The function $dist(\boldsymbol{p}_1, \boldsymbol{p}_2)$ measures the discrepancy between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ where the traditional $\mathcal{L}_1$ norm [4, 33], slide wasserstein distance [16], classifier determinacy disparity [18] can be used.

In the third step, the feature extractor g is updated to align the distribution between two domains by minimizing the prediction discrepancy of unlabeled target samples with the fixed two classifiers as

$$\min_{g} \mathcal{L}_{dis}(D_t). \tag{4}$$

In summary, traditional bi-classifiers adversarial learning methods [16, 18, 33] align distribution between two domains by playing a minimax game between feature extractor and two classifiers.

4 METHOD

Overview. We adopt the existing bi-classifier adversarial learning paradigm with three steps. In the first step, our method is consistent with traditional bi-classifier adversarial learning methods, using labeled source samples to train the feature extractor g and two classifiers h_1 and h_2. Furthermore, we introduce a memory to store m source features for each category. Through the memory,

the source cluster center of each category can be approximately obtained efficiently. As shown in Figure 2, in the second step, an *Expertise-aware Classifier Interference* (ECI) strategy is employed. In reference to the pseudo labels estimated based on the memory at the beginning of each epoch, the target samples are divided into two subsets, each of which represents those better predicted by the classifiers h_1 and h_2, respectively. One classifier maximizes the prediction discrepancy against the other classifier on the subset where it yields better prediction and minimizes the prediction discrepancy on another subset where it yields worse prediction. In the final step, along with the original prediction discrepancy minimization, the feature representations of the target domain training samples are aligned to source cluster centers, which are calculated based on memory, to perform conditional distribution alignment.

4.1 Expertise-aware classifier interference

4.1.1 Pseudo-labeling target samples. The pseudo-labeling is based on clustering. To obtain the cluster center of each category of the source domain in real time, a memory is constructed to save part source features. At the beginning of training, we sweep the source domain and randomly select m source sample features for each category and store them in memory. Since the source domain data is labeled, any source samples can be quickly fitted by the model and their features are highly discriminative. Therefore, to ensure the real time nature of memory, in each iteration, it is updated by replacing the original stored features with the source features in the current batch for each category based on the first in first out principle.

Then, the memory is used to calculate source cluster centers which are regarded as initial cluster centers of K-means clustering algorithm [14]. The specific steps at the beginning of each epoch are as follows. (1) Each source cluster center is approximately calculated by the labeled source samples in memory as $\boldsymbol{c}_k^s = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{f}_i^{s,k}$, $k \in \{1, 2, \ldots, K\}$, where $\boldsymbol{f}_i^{s,k}$ is the i-th sample feature respect to k-th category in the memory. (2) K-means clustering algorithm [14] is used to cluster the target samples. The initial cluster centers are initialized as the source cluster centers, i.e., $\boldsymbol{c}_k^t = \boldsymbol{c}_k^s$. (3) Each target sample calculates its distance to each cluster center to obtain a pseudo label, that is, $\hat{\boldsymbol{y}}_i^t = \arg\min_k cos(g(\boldsymbol{x}_i^t), \boldsymbol{c}_k^t)$, where the $cos(\cdot, \cdot)$ is the cosine distance function; then, the cluster center is updated according to the current pseudo labels of the target domain, i.e., $\boldsymbol{c}_k^t = \frac{1}{n_{tk}}\sum_{i=1}^{n_t} \mathbb{1}_{\hat{\boldsymbol{y}}_i^t=k} g(\boldsymbol{x}_i^t)$, $j \in \{1, 2, \ldots, K\}$, where $n_{tk} = \sum_{i=1}^{n_t} \mathbb{1}_{\hat{\boldsymbol{y}}_i^t=k}$. We repeat the above procedure (3) and (4) until the algorithm converges. After this step, all target domain samples can get their pseudo labels. For convenience, we express these pseudo labels using one hot vector as $\{\tilde{\boldsymbol{y}}_i^t\}_{i=1}^{n_t}$.

4.1.2 Expertise-aware classifier interference (ECI). Our ECI strategy aims to improve the second step of traditional bi-classifier adversarial learning methods. In this step, the two classifiers are optimized while the feature extractor is fixed. As mentioned before, the traditional methods maximize the prediction discrepancy, making the predictions inconsistent and yielding more ambiguous target samples. ECI is designed for solving this problem.

Specifically, ECI first needs to know which target samples can be better classified by h_1 or h_2. So the first step is dividing the target samples into two subsets which correspond to the h_1 and h_2 respectively. At each iteration, given a batch of target samples $\{\boldsymbol{x}_i^t\}_{i=1}^{B^t}$, their predictions from two classifiers are $\{\boldsymbol{p}_{i,1}^t\}_{i=1}^{B^t}$, $\{\boldsymbol{p}_{i,2}^t\}_{i=1}^{B^t}$, respectively. Since the target domain has no label information, pseudo label, which is described in Section 4.1.1, is used to approximate the current label information. The Kullback-Leibler (KL) divergence [10], $KL(P|Q) = \sum p_i \log(\frac{p_i}{q_i})$, can be used to calculate the similarity between the classification prediction and the one-hot pseudo label vector for each sample. If the KL divergence is small, it means the classification prediction is close to the pseudo label. So for a specific target sample, if the KL divergence corresponding to a classifier is smaller, without losing generality, we can consider this classifier performs better on this sample. Based on this point, the target samples are split into two parts as:

$$B_1^t = \{x_i^t | KL(\tilde{\boldsymbol{y}}_i^t, \boldsymbol{p}_{i,1}^t) < KL(\tilde{\boldsymbol{y}}_i^t, \boldsymbol{p}_{i,2}^t)\},$$
$$B_2^t = \{x_i^t | KL(\tilde{\boldsymbol{y}}_i^t, \boldsymbol{p}_{i,1}^t) > KL(\tilde{\boldsymbol{y}}_i^t, \boldsymbol{p}_{i,2}^t)\},$$

where the sets B_1^t and B_2^t correspond to the classifiers h_1 and h_2 respectively where their performances are better.

As traditional bi-classifier adversarial learning methods, the target domain samples out of source distribution are found by maximizing the prediction discrepancy, while our ECI strategy further reduces the ambiguous samples as much as possible. In order to achieve the above two optimizations at the same time, the importance of using the ECI strategy to divide the samples is reflected. Specifically, for updating the classifier h_1, it can perform better than h_2 on B_1^t set according to the ECI strategy, so it is further required to maximize the prediction discrepancy with h_2 to detect the target samples excluded by the support of the source domain [33]; while h_1 perform worse than h_2 on B_2^t set, which may predict incorrectly and yield the ambiguous target samples on B_2^t set, so it is needed to minimize the prediction discrepancy to correct its prediction. The loss function of h_1 is as follows:

$$\min_{h_1} \mathcal{L}_{dis1}(D_t) = -\mathcal{L}_{dis}(B_1^t) + \mathcal{L}_{dis}(B_2^t), \tag{5}$$

where the first term pushes the prediction of classifier h_1 far away from that of h_2 on the B_1^t set while the second term let it close to prediction of h_2 on the B_2^t set, the definition of $\mathcal{L}_{dis}(\cdot)$ refers to Eq. (3). On the other hand, for updating the classifier h_2, the objective function is defined as:

$$\min_{h_2} \mathcal{L}_{dis2}(D_t) = \mathcal{L}_{dis}(B_1^t) - \mathcal{L}_{dis}(B_2^t), \tag{6}$$

where the second term pushes the prediction of classifier h_2 far away from that of h_1 on the B_2^t set while the first term enforces it close to the prediction of h_1 on the B_1^t set.

Furthermore, we need the updated classifiers also work in the source domain. By combing above objective functions, the overall loss of our ECI strategy is formulated as,

$$\min_{h_1} \mathbb{E}_{x_i^s \in D_s} \mathcal{L}^{ce}(h_1(g(\boldsymbol{x}_i^s)), \boldsymbol{y}_i^s) + \mathcal{L}_{dis1}(D_t), \tag{7}$$

$$\min_{h_2} \mathbb{E}_{x_i^s \in D_s} \mathcal{L}^{ce}(h_2(g(\boldsymbol{x}_i^s)), \boldsymbol{y}_i^s) + \mathcal{L}_{dis2}(D_t). \tag{8}$$

Remark. In addition to using KL divergence to divide the target domain samples into two subsets, we also consider using the self-entropy as the evaluation criterion, which is shown in the Model Analysis (Section 5.2).

4.1.3 Analysis. Here we interpret how our ECI strategy can make the two classifiers more class discriminative compared with the traditional counterpart. For easier explanation, we consider a two-class classification task, i.e., the label space $\mathcal{Y} = \{1, 2\}$. Given a target domain sample x, let the predictions of the two classifiers be $P_1 = [p_{1,1}, p_{1,2}]$ and $P_2 = [p_{2,1}, p_{2,2}]$, respectively; The pseudo label $\tilde{\boldsymbol{y}}$ of x is $[1, 0]$ (i.e., of class 1) and the function $dist(\cdot, \cdot)$ in Eq. (3) is $\mathcal{L}_1$ normalized; Without loss of generality, we assume the classifier h_1 performs better, satisfying the following relations: $p_{1,1} > p_{2,1}$, $p_{1,2} < p_{2,2}$, $KL(\tilde{\boldsymbol{y}}, P_1) < KL(\tilde{\boldsymbol{y}}, P_2)$, and $x \in B_1^t$.

With the ECI strategy, the loss functions derived on this sample x for updating the two classifiers h_1 and h_2 can be written as:

$$\min_{h_1} \mathcal{L}_{dis1} = -|P_1 - P_2| = -p_{1,1} + p_{2,1} + p_{1,2} - p_{2,2}, \tag{9}$$

$$\min_{h_2} \mathcal{L}_{dis2} = |P_1 - P_2| = p_{1,1} - p_{2,1} - p_{1,2} + p_{2,2}, \tag{10}$$

where Eq. (9) focuses on optimizing h_1, with P_2 regarded as the objective constant. Similarly, Eq. (10) is for updating h_2. The updating formulation is expressed as:

$$\hat{P}_1 = P_1 - \theta \frac{\partial \mathcal{L}_{dis1}}{\partial P_1} = [p_{1,1} + \theta, p_{1,2} - \theta], \tag{11}$$

$$\hat{P}_2 = P_2 - \theta \frac{\partial \mathcal{L}_{dis2}}{\partial P_2} = [p_{2,1} + \theta, p_{2,2} - \theta], \tag{12}$$

where $\theta > 0$ is the learning rate. By the KL formula, we have $KL(\tilde{\boldsymbol{y}}, \hat{P}_i) < KL(\tilde{\boldsymbol{y}}, P_i)$ with $i \in \{1, 2\}$. This means the two classifiers *both* become more discriminative after the updating as above.

In comparison, the traditional bi-classifier adversarial learning methods maximize the prediction discrepancy between the two classifiers. This is the same as our ECI strategy when optimizing the classifier h_1. However, for updating the classifier h_2, they still maximize the prediction discrepancy as follows:

$$\min_{h_2} \mathcal{L}_{dis2} = -|P_1 - P_2| = -p_{1,1} + p_{2,1} + p_{1,2} - p_{2,2}. \tag{13}$$

The updated P_2 then becomes:

$$\hat{P}_2 = P_2 - \theta \frac{\partial \mathcal{L}_{dis2}}{\partial P_2} = [p_{2,1} - \theta, p_{2,2} + \theta]. \tag{14}$$

In this case, $KL(\tilde{\boldsymbol{y}}, P_2) < KL(\tilde{\boldsymbol{y}}, \hat{P}_2)$. This means the classifier h_2 degrades on the target sample x, i.e., classifying it to class 2 (a wrong prediction). However, the updated classifier h_1 can instead correctly classify x to class 1, making x as an ambiguous sample with inconsistent predictions by the two classifiers. In short, given a target sample, the traditional bi-classifier adversarial learning methods improve the better-performing classifier whilst degrading the worse-performing one. On the contrary, our ECI improves *both classifiers* as elaborated above.

4.2 Representation alignment

Traditional bi-classifier adversarial learning methods typically try to minimize the prediction discrepancy of target samples to optimize feature extractor g whilst aligning the features across domains, as formulated in Eq. (4). Whilst easy target samples are often aligned to correct classes, this strategy is ineffective in tackling the ambiguous target samples which may be detected by misclassified classifier and performed an inaccurate class-wise distribution alignment [4].

Algorithm 1 Class Discriminative Adversarial Learning

Input: Source domain $D_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$, target domain $D_t = \{(\boldsymbol{x}_i^t)\}_{i=1}^{n_t}$, the epoch number T, the mini-batch number M.
Output: An adapted model.
Procedure:

```
for t = 1:T do
    Update pseudo labels of target domain {y_i^t}_{i=1}^{n_t} ;
    for m = 1:M do
        Forward a mini-batch through the model;
        Step 1: Train g, h_1, h_2 based on labeled source samples (Eq. (1));
        Step 2: Train h_1, h_2 based on both labeled source samples and pseudo-labeled target samples (Eqs.(7) and (8)) ;
        Step 3: Update memory based on source samples and train g based on pseudo-labeled target samples (Eq. (16)) ;
    end for
end for
return Adapted model.
```

To address this problem, a representation regularization is further imposed for conditional distribution alignment between two domains. Formally, it forces the features of target samples being close to the source distribution, i.e., pushed away from ambiguous regions. This regularization is designed as follows:

$$\mathcal{L}_{clu}(D) = \frac{1}{K} \sum_{k=1}^{K} \left\| \frac{1}{n_{tj}} \sum_{i=1}^{n_t} \mathbb{1}_{\hat{y}_i^t = j} \phi(g(\boldsymbol{x}_i^t)) - \phi(\boldsymbol{c}_k^s) \right\|^2 \tag{15}$$

where $\boldsymbol{c}_k^s$ is the k-th class center calculated based on the up-to-date memory (Section 4.1.1), $D = \{D_s, D_t\}$, and $n_{tj} = \sum_{i=1}^{n_t} \mathbb{1}_{\hat{y}_i^t = j}$. ϕ is a Gaussian kernel function often used in UDA. Conceptually, $\mathcal{L}_{clu}(D)$ aligns the per-category representation centers across source and target domains by encouraging the semantic correspondence between pseudo labels and genuine labels.

Combining the traditional alignment constraint (Eq. (4)) and proposed regularization, our representation alignment objective is formed as:

$$\min_{g} \mathcal{L}_{dis}(D_t) + \alpha \mathcal{L}_{clu}(D). \tag{16}$$

where $\alpha > 0$ is a trade-off hyperparameter.

In training, we repeat the above process until the model converges. At test time, the mean prediction results from two adapted classifiers are used as the test results. Our algorithm is summarized in Algorithm 1.

5 EXPERIMENTS

Datasets. In our experiments we use three standard UDA datasets. ***ImageCLEF***[25][1] is a popular dataset, it contains three domains: Caltech-256 (C), ImageNet ILSVRC2012 (I) and PASCALVOC2012 (P). There are 600 images in each domain and 50 for each category. ***Office-Home***[40][2] is a more challenging dataset with 15,588 images from 65 classes in four domains: Artistic images (A), Clip-Art images (C), Product images (P) and RealWorld images (R). ***Visda-17***[31][3] is

[1] https://www.imageclef.org/2014/adaptation
[2] https://www.hemanthdv.org/officeHomeDataset.html
[3] http://ai.bu.edu/visda-2017/

Table 1: Comparison with the state-of-the-art methods on *ImageCLEF* dataset. Metric: classification accuracy (%); Backbone: ResNet50.

Method	Venue	I→P	P→I	I→C	C→I	C→P	P→C	*avg*
ResNet-50 [9]	CVPR16	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DANN [6]	CVPR16	75.0	86.0	96.2	87.0	74.3	91.5	85.0
CDAN+E[24]	NIPS18	77.7	90.7	97.7	91.3	74.2	94.3	87.7
A^2LP[45]	ECCV20	79.6	92.7	96.7	92.5	78.9	96.0	89.4
ETD[17]	CVPR20	**81.0**	91.7	**97.9**	**93.3**	79.5	95.0	89.7
CKB+MMD[27]	CVPR21	80.7	92.2	96.5	92.2	79.9	96.7	89.7
CGDM [4]	CVPR21	78.7	93.3	97.5	92.7	79.2	95.7	89.5
BCDM [18]	AAAI21	79.5	93.2	96.8	91.3	78.9	95.8	89.3
MCD [33]	CVPR18	77.3	89.2	92.7	88.2	71.0	92.3	85.1
MCD+ECI	Ours	79.3 ±0.2	92.5 ±0.1	96.3 ±0.1	90.5±0.2	78.0±0.2	94.8±0.1	88.6
SWD [16]	CVPR19	78.3	90.3	93.2	89.7	73.3	93.8	86.4
SWD+ECI	Ours	79.8±0.1	92.7±0.2	96.8±0.0	92.9±0.1	77.3±0.2	96.5±0.1	89.3
CDAL	Ours	80.4±0.1	**93.7±0.1**	97.8±0.0	**93.3±0.1**	**80.2±0.3**	**97.5±0.2**	**90.5**

Table 2: Comparisons with the state-of-the-art methods on *Office-Home* dataset. Metric: classification accuracy (%); Backbone: ResNet50.

Method	Venue	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	*avg*
ResNet-50[9]	CVPR16	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [6]	CVPR16	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN+E [24]	NIPS18	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
ALDA [2]	AAAI20	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
MDD+IA [11]	ICML20	56.2	**77.9**	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
MetaAlign [41]	CVPR21	59.3	76.0	80.2	65.7	74.7	75.1	65.7	**56.5**	81.6	74.1	**61.1**	**85.2**	71.3
CKB+MMD[27]	CVPR21	54.2	74.1	77.5	64.6	72.2	71.0	64.5	53.4	78.7	72.6	58.4	82.8	68.7
TSA [19]	CVPR21	53.6	75.1	78.3	64.4	73.7	72.5	62.3	49.4	77.5	72.2	58.8	82.1	68.3
TCM[43]	ICCV21	58.6	74.4	79.6	64.5	74.0	75.1	64.6	56.2	80.9	**74.6**	60.7	84.7	70.7
SCDA[20]	ICCV21	57.5	76.9	**80.3**	65.7	74.9	74.5	65.5	53.6	79.8	74.5	59.6	83.7	70.5
MCD[33]	CVPR18	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
MCD+ECI	Ours	57.4	74.0	78.6	62.3	73.7	75.0	64.4	54.5	81.1	73.3	60.3	83.7	69.9
SWD[16]	CVPR19	51.3	70.3	75.0	56.2	69.4	71.6	59.8	53.8	80.2	71.1	59.2	83.4	66.8
SWD+ECI	Ours	58.0	75.4	79.0	64.1	73.3	74.9	64.6	54.1	81.1	72.5	60.5	83.8	70.1
CDAL	Ours	**59.5**	77.8	80.0	**67.0**	**77.1**	**76.6**	**66.6**	56.2	**81.8**	74.3	60.6	84.6	**71.8**
		±0.3	±0.1	±0.1	**±0.2**	**±0.2**	**±0.0**	**±0.2**	±0.1	**±0.0**	±0.1	±0.0	±0.1	

a challenging benchmark for domain adaptation which focuses on the 12-class synthesis-to-real object recognition task. The source domain contains 152,397 synthetic images and the target domain has 55,388 real object images.

Implementation details. Our experiment is performed on the Pytorch platform. Each experiment was run 5 times to enhance the robustness of the results. For fair comparison, all models use the same feature extractor. Specifically, Resnet50 is used as the backbone on the ImageCLEF and Office-home datasets, and Resnet101 is used on the Visda-17 dataset. In addition, two classifiers are linear networks with one layer. The SGD optimizer is chosen to update the model and CosineAnnealingLR [26] is used to update the learning rate of the optimizer. For the hyperparameter α, it set as 0.2 for Eq. (16) in all experiments. For hyperparameter m, which is the memory size of the feature stored for each category, it set as 6/10/10 for Office-Home/ImageCLEF/Visda17. Our method uses MCD [33] as baseline for convenience.

Competitors. To prove the effectiveness of our method, we compare our method with the following state-of-the-art methods: the methods based on *statistic moment matching* are CKB+MMD [27], ETD [17], TSA [19] and TCM [43]; the methods based on *adversarial learning framework* are DANN [6], CDAN+E [24], MDD+IA [11], ALDA [2], MetaAlign [41], DWL [42], CLS [22] and SCDA [20]. In addition, MCD [33], SWD [16], BCDM [18] and CGDM [4], which belong to *the category of bi-classifier adversarial learning* and are related to our method. 'MCD+ECI' and 'SWD+ECI' represent the corresponding methods where ECI strategy is applied.

5.1 Result analysis

Results on ImageCLEF. The comparisons between our method and other state-of-the-art UDA methods are shown in Table 1. Our method CDAL significantly outperforms other methods and obtains the best overall performance. The reason is our method requires the model to be more class discriminative when performing distribution alignment. Compared with CGDM, which applies gradient discrepancy alignment to solve the problem of ambiguous target sample, CDAL yields 1.0% improvement, which proves that our solution can handle ambiguous target samples effectively. Compared with previous top performing ETD and CKB+MMD, both of them belong to *statistic moment matching*, CDAL also yields 0.8% improvement, which proves the effectiveness of our method.

Table 3: Comparison with the state-of-the-art methods on *Visda-17* dataset. Metric: per-class classification accuracy (%); Backbone: ResNet101.

Method	Venue	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	*avg*
ResNet-101 [9]	CVPR16	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DANN [6]	JMLR16	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
CDAN [24]	NIPS18	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
ALDA [2]	AAAI20	93.8	74.1	82.4	69.4	90.6	87.2	89.0	67.6	93.4	76.1	87.7	22.2	77.8
BCDM [18]	AAAI21	95.1	**87.6**	81.2	73.2	92.7	95.4	86.9	**82.5**	95.1	84.8	88.1	39.5	83.4
DWL [42]	CVPR21	90.7	80.2	**86.1**	67.6	92.4	81.5	86.8	78.0	90.6	57.1	85.6	28.7	77.1
CGDM [4]	CVPR21	93.4	82.7	73.2	68.4	92.9	94.5	88.7	82.1	93.4	82.5	86.8	49.2	82.3
CLS [22]	ICCV21	92.6	84.5	73.7	**72.7**	88.5	83.3	89.1	77.6	89.5	89.2	85.8	**72.7**	81.6
MCD[33]	CVPR18	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
MCD+ECI	Ours	93.4	77.2	76.9	51.2	89.9	92.1	83.4	74.8	84.7	72.3	85.8	55.2	78.1
SWD[16]	CVPR19	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
SWD+ECI	Ours	93.6	78.6	76.7	51.1	90.3	93.2	83.4	75.7	85.2	75.6	85.8	57.0	78.9
CDAL	Ours	**97.5**	84.9	81.0	70.5	**97.1**	**97.3**	**90.6**	80.9	**96.2**	**94.9**	**88.2**	48.7	**85.7**

Results on Office-Home. The results are shown in Table 2. Our method achieves the best accuracy on 6 out of 12 tasks in total and overall performance. Compared with TCM and MetaAlign, our method leads the overall performance by 1.1% and 0.5% respectively. Especially on tasks C→P and C→R, the performance of our method has been improved by 2.2% and 1.5%, respectively, compared with the state-of-the-art results.

Results on Visda-17. In this dataset, the per-class classification accuracy is reported which is shown in Table 3. And our method yields a huge improvement compared with other method. Compared with the previous best performing method BCDM, which is also based on *bi-classifier adversarial learning*, our method improves the accuracy by 2.3%. From Table 1-3, all tasks have proved that MCD+ECI is better than MCD and SWD+ECI is better than SWD, which proves effectiveness of our ECI strategy.

5.2 Model analysis

Table 4: Stability analysis of ECI strategy respect to different criteria.

Method	I→P	P→I	I→C	C→I	C→P	P→C
MCD	77.3	89.2	92.7	88.2	71.0	92.3
MCD+ECI(SE)	78.0	91.8	96.8	91.7	77.8	95.0
MCD+ECI(KL)	79.3	92.5	96.3	90.5	78.0	94.8

ECI strategy based on self-entropy. In Section 4.1.2, the pseudo labels are used to divide the target samples into two parts based on KL divergence. Self-entropy is also popular for many methods to judge the quality of prediction [13, 24]. Here, we further use self-entropy to analyze the stability of our ECI strategy. The two parts are defined as

$$B_1^t = \{x_i^t | SE(\boldsymbol{p}_1^t) < SE(\boldsymbol{p}_2^t)\}, \tag{17}$$

$$B_2^t = \{x_i^t | SE(\boldsymbol{p}_1^t) > SE(\boldsymbol{p}_2^t)\} \tag{18}$$

where $SE(P) = -\sum p_i \log p_i$ represents self-entropy[34]. The results are shown in Table 4, where MCD+ECI(SE) represents the method of dividing samples based on self-entropy and the second step of MCD is modified using our ECI strategy. As shown in Table 4, the performance of MCD is also improved. It shows the stability of our ECI strategy to the classification quality criterion.

Table 5: Comparison of number of ambiguous target samples.

Method	I→P	P→I	I→C	C→I	C→P	P→C
MCD	10	11	10	17	31	5
MCD+ECI	6	7	4	6	15	2
CDAL	5	4	3	2	6	2

The performance of reducing ambiguous target samples. The central point of our method is reducing the ambiguous target domain samples, such that the features can be discriminatively aligned. Therefore, we count the number of ambiguous samples for the trained models using different strategies, where MCD is original method [33], MCD+ECI means the ECI strategy is applied based on MCD, and CDAL is our complete algorithm where feature regularization is applied. The result is shown in Table 5, which shows that our proposed method can effectively reduce the number of ambiguous samples by both of ECI strategy and feature regularization.

Table 6: The ablation study of second step.

Method	I→P	P→I	I→C	C→I	C→P	P→C
PSE	74.0	89.7	93.2	88.6	66.8	93.0
MCD	77.3	89.2	92.7	88.2	71.0	92.3
MCD+ECI	**79.3**	**92.5**	**96.3**	**90.5**	**78.0**	**94.8**

The ablation study of second step. In this experiment, we modify the second step of the original MCD by using the pseudo label, which is described in Section 4.1.1, and the ECI strategy, respectively. The results are shown in Table 6. For PSE, we directly use pseudo labels to train two classifiers for target domain samples in the second step as baseline. From the result in Table 6, PSE can slightly outperform MCD in some simple tasks, such as P→I, I→C, C→I and P→C, but it has a huge lag compared to MCD on some difficult tasks, such as I→P and C→P. This is because the domain discrepancy is often large on difficult tasks, resulting in relatively large noise of pseudo labels, which affects the performance of the model. In addition, compared with MCD+ECI and PSE, MCD+ECI leads on

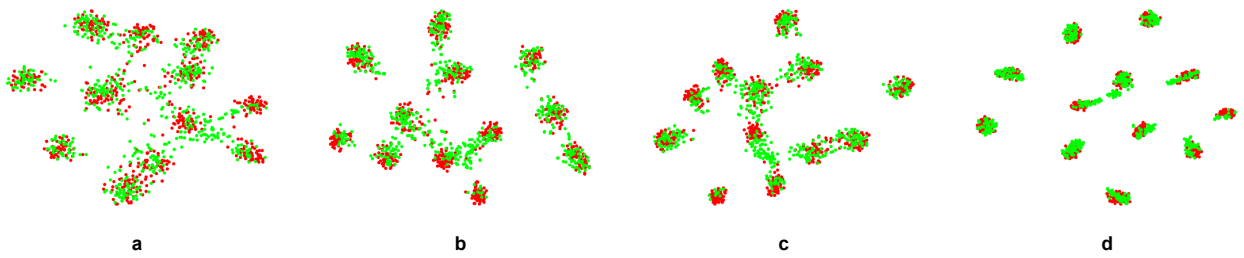

Figure 3: Feature visualizations by T-SNE. The learned features by different combinations of strategies are shown on the task I(red)→P(green) of *ImageCLEF* dataset, (a) MCD, (b) MCD+ECI, (c) MCD+CLU, (d) CDAL.

all tasks. In fact, both are essentially class discriminative learning, except that PSE uses hard pseudo labels while our method uses the prediction of another classifier as a soft label for learning, which has better generalization ability [29].

Table 7: The ablation study of full algorithm.

Method	I→P	P→I	I→C	C→I	C→P	P→C
MCD	77.3	89.2	92.7	88.2	71.0	92.3
MCD+ECI	79.3	92.5	96.3	90.5	78.0	94.8
MCD+CLU	79.2	93.1	97.2	91.8	76.4	96.2
CDAL	**80.4**	**93.7**	**97.8**	**93.3**	**80.2**	**97.5**

The ablation study of full algorithm. To analyze the role of each part of our method, we conduct an ablation study on ImageCLEF dataset, which uses the original MCD as the baseline, and the experimental results are shown in Table 7. From Table 7, we can get the following conclusions: (1) Due to no class discriminative information used in the original MCD method, the ambiguous target samples may match to wrong classes, which makes a suboptimal adapted model. (2) By using ECI strategy, MCD+ECI can well solve the problem of ambiguous target samples, so as to improve the performance of model. In addition, MCD+CLU uses the representation regularization loss $\mathcal{L}_{clu}$ to explicitly perform feature distribution alignment between two domains, which can directly give the ambiguous samples an optimization direction. (3) CDAL uses the above two strategies to optimize the model jointly, and finally achieves the best results.

Visual analysis. To give an intuitive understanding of our method, the features of the transfer task I→P on ImageCLEF dataset are visualized by t-SNE [28] in Fig. 3. It presents consistent observations :(a) MCD, (b) MCD+ECI, (c) MCD+CLU, (d) CDAL. It can be observed that the features in Fig. 3(a) are very scattered. Compared with Fig. 3(a), the features in Fig. 3(b) are more concentrated due to the influence of ECI strategy. Compared with Fig. 3(a), the features in Fig. 3(c) are also more concentrated due to the influence of explicit alignment loss $\mathcal{L}_{clu}$. Finally, the features learned by our method, which are the original MCD equipped with the proposed ECI strategy and explicit alignment loss, are shown in Fig. 3(d). Compared with the features learned by other variants (Fig. 3(a)-Fig. 3(c)), the features in Fig. 3(d) are the most concentrated.

Parameter Analysis. To verify the robustness of our method, all transfer tasks on ImageCLEF are carried to analyze the sensitivity

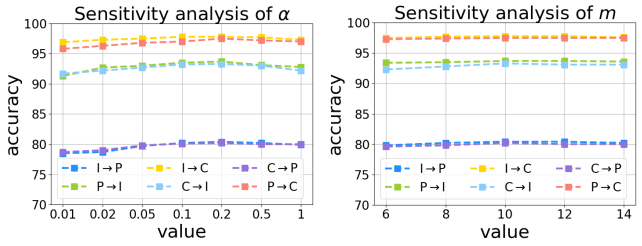

Figure 4: Sensitivity analysis of (l) loss weight α and (r) size of memory m.

of parameter α and m. For the parameter α, which is a balance hyperparameter in Eq. (16), the experimental result is shown in the left of Fig. 4, which turns it from 0.01 to 1.0. When α changes from 0.01 to 0.2, the accuracy of our method shows an upward trend. When α changes from 0.2 to 1.0, the accuracy of our method can drop a bit. On the whole, the accuracy of our method does not change drastically, which proves the robustness of our method. For the parameter m, which is size of our memory during the training, the experimental result is shown in the right of Fig. 4, which takes a value every 2 from 6 to 14. Overall, the change of m has little effect on the accuracy, that is, the model is very robust to m.

6 CONCLUSIONS

In this paper, we investigated the problem of ambiguous target samples in the bi-classifier adversarial learning which is always ignored by previous approaches. A Class Discriminative Adversarial Learning (CDAL) method is proposed which employs an ECI strategy and a representation regularization based on traditional bi-classifier adversarial learning. The ECI strategy boost the two classifiers learning each other to reduce ambiguous target domain samples. Instead of only minimizing discrepancy to align the distribution between two domains, the proposed representation regularization does explicit feature alignment which forces target samples close to source distribution and far away ambiguous region. These two strategies can reduce the ambiguous target samples in adaptation process. The results on the public datasets prove the reliability of our method.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2018YFE0203900) and Sichuan Science and Technology Program (2020YFG0476).

REFERENCES

[1] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. 2019. Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11457–11466.

[2] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. 2020. Adversarial-learned loss for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3521–3528.

[3] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. 2020. Unbiased Mean Teacher for Cross Domain Object Detection. *arXiv preprint arXiv:2003.00707* (2020).

[4] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. 2021. Cross-Domain Gradient Discrepancy Minimization for Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3937–3946.

[5] Geoffrey French, Michal Mackiewicz, and Mark Fisher. 2018. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*.

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[7] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. 2015. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE International Conference on Computer Vision*. 2551–2559.

[8] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. 2016. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*. Springer, 597–613.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[10] John R Hershey and Peder A Olsen. 2007. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, Vol. 4. IEEE, IV–317.

[11] Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. 2020. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *International Conference on Machine Learning*. PMLR, 4816–4827.

[12] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. 2019. Contrastive Adaptation Network for Unsupervised Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4893–4902.

[13] Youngeun Kim, Sungeun Hong, Donghyeon Cho, Hyoungseob Park, and Priyadarshini Panda. 2020. Domain Adaptation without Source Data. *arXiv preprint arXiv:2007.01524* (2020).

[14] K Krishna and M Narasimha Murty. 1999. Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 29, 3 (1999), 433–439.

[15] Sajja Tulasi Krishna and Hemantha Kumar Kalluri. 2019. Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)* 7, 5S4 (2019), 427–432.

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

[17] Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. 2020. Enhanced transport distance for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13936–13944.

[18] Shuang Li, Fangrui Lv, Binhui Xie, Chi Harold Liu, Jian Liang, and Chen Qin. 2021. Bi-Classifier Determinacy Maximization for Unsupervised Domain Adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 8455–8464.

[19] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. 2021. Transferable Semantic Augmentation for Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11516–11525.

[20] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. 2021. Semantic Concentration for Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9102–9111.

[21] Ming-Yu Liu and Oncel Tuzel. 2016. Coupled generative adversarial networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 469–477.

[22] Xiaofeng Liu, Zhenhua Guo, Site Li, Fangxu Xing, Jane You, C-C Jay Kuo, Georges El Fakhri, and Jonghye Woo. 2021. Adversarial unsupervised domain adaptation with conditional and label shift: Infer, Align and Iterate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10367–10376.

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*. PMLR, 97–105.

[24] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional Adversarial Domain Adaptation. In *NeurIPS*.

[25] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*. PMLR, 2208–2217.

[26] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[27] You-Wei Luo and Chuan-Xian Ren. 2021. Conditional Bures Metric for Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13989–13998.

[28] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[29] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems* 32.

[30] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2009), 1345–1359.

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

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

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

[34] Claude E. Shannon. 2001. A Mathematical Theory of Communication. *The Bell System Technical Journal* 5, 3 (2001), 3–55.

[35] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. 2017. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2107–2116.

[36] Baochen Sun, Jiashi Feng, and Kate Saenko. 2017. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*. Springer, 153–171.

[37] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. 2019. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825* (2019).

[38] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1195–1204.

[39] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7167–7176.

[40] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5018–5027.

[41] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. 2021. MetaAlign: Coordinating Domain Alignment and Classification for Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16643–16653.

[42] Ni Xiao and Lei Zhang. 2021. Dynamic Weighted Learning for Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15242–15251.

[43] Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. 2021. Transporting Causal Mechanisms for Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8599–8608.

[44] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. 2017. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811* (2017).

[45] Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. 2020. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *European Conference on Computer Vision*. Springer, 781–797.

[46] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. 2017. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing* 14, 2 (2017), 119–135.

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2223–2232.

[48] Junbao Zhuo, Shuhui Wang, Weigang Zhang, and Qingming Huang. 2017. Deep unsupervised convolutional domain adaptation. In *Proceedings of the 25th ACM International Conference on Multimedia*. 261–269.