Unsupervised Class-imbalanced Domain Adaptation with Pairwise Adversarial Training and Semantic Alignment

Weili Shi, Ronghang Zhu, and Sheng Li, *Senior Member, IEEE*

***Abstract*—Unsupervised domain adaptation (UDA) has become an appealing approach for knowledge transfer from a labeled source domain to an unlabeled target domain. However, when the classes in source and target domains are imbalanced, most existing UDA methods experience significant performance drop, as the decision boundary usually favors the majority classes. Some recent class-imbalanced domain adaptation (CDA) methods aim to tackle the challenge of biased label distribution by exploiting pseudo-labeled target samples during the training process. However, these methods suffer from the issues with unreliable pseudo labels and error accumulation during training. In this paper, we propose a pairwise adversarial training approach for class-imbalanced domain adaptation. Unlike conventional adversarial training in which the adversarial samples are obtained from the ℓ_p ball of the original samples, we generate adversarial samples from the interpolated line of the aligned pairwise samples from source and target domains. The pairwise adversarial training (PAT) is a novel data-augmentation method which can be integrated into existing unsupervised domain adaptation (UDA) models to tackle the CDA problem. Inspired by the noise injection, we also extend the pairwise adversarial training to noisy pairwise adversarial training (nPAT), in which the random noise is injected into the generation of the adversarial samples. In our study, we evaluate our proposed methods as well as the baselines on three major benchmark datasets, namely Office-Home, DomainNet and Office-31. For Office-Home and Office-31, we sample the data according to the Reversely-unbalanced Source and Unbalanced Target (RS-UT) protocol so that the class distribution can be imbalanced. The extensive experimental results show that UDA models integrated with our proposed nPAT can achieve prominent improvements on most tasks compared to the baseline methods as well as the state-of-the-art CDA methods. The average accuracy of our nPAT can achieve 66.56% and 80.22% on Office-Home and DomainNet, respectively, which are higher than that of the second-best methods. Besides, Experiments also show that our method is robust to the unreliability of the pseudo labels.**

***Index Terms*—Imbalanced domain adaptation; adversarial training; semantic alignment**

I. Introduction

Domain adaptation (DA) aims to alleviate the domain gap between source and target domains. Recent years have witnessed the significant progress of DA based on deep neural networks [1]–[10]. Most of existing DA methods are belong to unsupervised domain adaptation (UDA) which targets to achieve knowledge transfer from a labeled source domain to an unlabelled target domain. These methods usually assume that only covariate shift occurs in the source and target domains, while the label distributions in two domains are identical. However, this assumption may not hold in real-world applications. For instance, in wild-life pictures, the commonly seen animals such as rabbit and deer appear more frequently than the rare animal such as panda and crocodile. Public datasets such as DomainNet [11] and and MSCOCO [12] exhibit imbalanced class distribution. Figure 1 illustrates the imbalanced label distributions in the *Real* domain and *Sketch* domain from the DomainNet dataset [11].

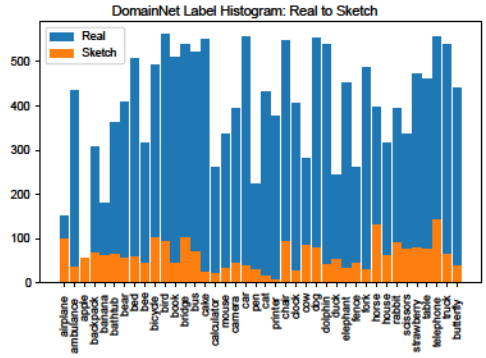


Fig. 1: Illustration of label distribution shift in DomainNet.

To address the issue of imbalanced label distributions in domain adaptation, some recent studies [13]–[15] try to jointly model the conditional feature distribution shift and label distribution shift (LDS). This problem is referred to as Class-imbalanced Domain Adaptation (CDA). Let x and y denote the samples and labels, respectively. p and q separately represent the probability distributions of source and target domains. The common assumptions in UDA involve the covariate shift (i.e., $p(x) \neq q(x)$) and identical label distribution (i.e., $p(y) = q(y)$). In CDA, however, apart from the covariate shift, both the conditional feature shift and label shift exist, i.e., $p(x \mid y) \neq q(x \mid y)$, $p(y) \neq q(y)$. Obviously, CDA is more challenging than UDA.

Recent studies [14] have demonstrated that the mainstream UDA methods suffer significant performance drop, as the classifier favors the majority classes. Only a few CDA approaches have been proposed by far. According to the previous work [14], the negative effect of label shift is reduced by exploiting the pseudo labelled target samples via self-

Manuscript received in January 2024.

W. Shi and S. Li are with the School of Data Science, University of Virginia, Charlottesville, VA, 22903. E-mail: {rhs2rr, shengli}@virginia.edu.

R. Zhu is with the School of Computing, University of Georgia, Athens, GA, 30602. E-mail: ronghangzhu@uga.edu.

training. The previous work [15] used an implicit sampling method based on pseudo labels to align the joint distribution between features and labels. The previous work [14], [15] that aims to solve the class-imbalanced domain adaptation commonly adopted the pseudo labels for the data from target domain. However, one critical problem of these methods is that the pseudo labels are likely to suffer from ill-calibrated probabilities [16], especially in the early beginning of the training process. And thus the unreliable pseudo labels cause error accumulation during the training process, which largely degrades the model performance. In our proposed method, the pseudo labeling technique is also adopted. However, since the target data with pseudo labels is not directly utilized in our training procedure, the problem of the error accumulation caused by the pseudo labels can be alleviated considerably.

Augmenting training data has been proven as an effective strategy to tackle the issue of biased label distributions in class-imbalance learning [17], [18]. In addition to the traditional data augmentation techniques, adversarial training is also capable of generating semantically meaningful synthetic samples that help enhance the robustness of models. However, these approaches only consider a single domain, and they cannot be directly applied to solve the CDA problem. In the previous study, Shi et al. [19] proposed a pairwise adversarial training (PAT) approach that augments training data for class-imbalanced domain adaptation. Unlike conventional adversarial training in which the adversarial samples are obtained from the ℓ_p ball of the original samples, the semantic adversarial samples are generated by convex combinations of the aligned pair-wise samples from source and target domains to alleviate the gap between the two domains. Recently, Xu et. al. [20] proposed the Noisy Feature Mixup (NFM) [20] in which the synthetic data is generated by the noise-perturbed convex combinations of pairs of data points and suggested that NFM [20] can achieve a better regularization effect than that of mixup [21]. Inspired by the NFM [20], in our study we propose the noisy pairwise training (nPAT) in which the random noise is injected in the generation of the interpolated adversarial sample to tackle the class-imbalanced domain adaptation problem. Due to the noise injection during the data augmentation process, the new synthetic data show a better regularization effect than that of PAT [19] on the biased decision boundary in CDA problem and extensive experimental results suggest that our proposed nPAT can achieve better performance than that of PAT [19] on most tasks. At last, a class-imbalanced semantic centroid alignment strategy is designed to explicitly align the source and target domains in the feature space.

This paper is a substantial extension of our previous work [19]. In our work, we propose a novel noisy pairwise adversarial training (nPAT) approach to generate the noisy interpolated adversarial samples to alleviate the gap between the imbalanced source domain and target domain. In addition, The experiments on multiple benchmark datasets are performed and the experimental results reveal the superiority of our nPAT over the previous PAT [19] method. Overall, the main contributions of this paper are four-fold. (1) We propose a novel pairwise adversarial training approach that generates adversarial samples from pairs of samples across the source and target domains, and further exploits these samples to augment training data. When dealing with imbalanced training data, the PAT can be integrated into existing domain adaptation models to improve the performance. (2) A new optimization algorithm is proposed to solve the pairwise adversarial training problem. (3) We extend the pairwise adversarial training by injecting the noise in the generation of the interpolated adversarial samples. (4) We conduct extensive evaluations on benchmark datasets, and results show that our approach obtains competing performance compared with the original models and state-of-the-art CDA methods. Besides, experiments also show that our approach is robust to the unreliability of the pseudo labels.

The paper is organized as follows. In Section II we talk about the related work. In Section III we thoroughly discuss our PAT and nPAT methods, the relevant algorithm and the corresponding theoretical explanation. The experimental results and the ablation study are illustrated in Section IV.

TABLE I: Primary notations used in the paper

Notation	Definition
$\mathcal{D}_S$	the labeled source domain
$\mathcal{D}_T$	the unlabeled target domain
N^s (N^t)	the number of the source data (target data)
x^s (x^t)	the original source data (target data)
y^s	the ground-truth label of source data
$\hat{y}^t$	the pseudo label of target data
x^{adv}	a interpolated adversarial sample (IAS)
x^{nadv}	a noisy interpolated adversarial sample (nIAS)
λ	the learnable mixup ratio
δ	the perturbation added on the original data
ξ^{mult} (ξ^{add})	the random multiplicative (additive) noise
σ_{mult} (σ_{add})	the pre-specified multiplicative (additive) noise level
C^s	source centroids
C^t	target centroids
P_k	probability threshold for generation of synthetic data
n_k	the number of samples from kth class
$\mathcal{L}_{UDA}$	loss function of existing UDA methods
$\mathcal{L}_{CE}$	cross-entropy loss function
$\mathcal{L}_{IAS}$	loss function for generation of IAS
$\mathcal{L}_{nIAS}$	loss function for generation of nIAS
$\mathcal{L}_{CA}$	centroid alignment loss function
α, β	coefficients of loss functions

II. Related Work

In this section, we first give a brief description about unsupervised domain adaptation. Then, we introduce the existing work in class-imbalanced domain adaptation. Last, we review the relevant topic, i.e., adversarial training.

A. Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) aims to transfer the knowledge learned from a labeled source domain to an unlabeled target domain. As UDA is very useful when label information of target domain is unavailable in new application scenarios, it has received increasing attention in recent years. We roughly divide the UDA techniques into two groups: discrepancy-based methods and adversarial-based methods, and review them respectively.

The discrepancy-based methods usually aim to align source and target feature distributions in the embedding space by

using various statistical distance metrics, e.g., Maximum Mean Discrepancy (MMD) [22], [23], Correlation Alignment (CORAL) [24], [25], and Wasserstein Distance (WD) [26], [27] , to explicitly minimizing different statistical divergences. The deep adaptation network (DAN) [28] maps both source and target deep features into reproducing kernel Hilbert spaces (RKHS) and minimizes the MMD to reduce the discrepancy between both domains. The joint adaptation networks (JAN) [29] aligns source and target domains under multiple domain-specific layers with a joint MMD criterion. The deep CORAL [30] extends CORAL to deep neural networks and minimizes the difference in second-order statistics between source and target distributions. The Wasserstain distance guided representation learning (WDGPL) [31] minimizes the estimated WD between the source and target deep features in an adversarial manner. Wang et al. [32] theoretically reveals the relationship between the MMD [22], [23] and discriminative distances to each other. And degradation of feature discriminability induced by the MMD can be mitigated by the proposed discriminative MMD method [32].

The adversarial-based methods [33], [34] focus on learning domain invariant features via domain adversarial training. The intuitive idea is that the learned source and target features should be indistinguishable for the domain classifier. The adversarial-based methods [35] have shown advanced adaptation ability over discrepancy-based methods. The adversarial discriminative domain adaptation (ADDA) [36] utilizes two feature extractors for source and target domains, respectively, to learn discriminative target features by fooling the domain discriminator. The conditional domain adversarial network (CDAN) [37] exploits discriminative information provided by the classifier to help adversarial adaptation under multimodal distributions. Recently, the margin disparity discrepancy (MDD) [38] proposes a novel measurement under the rigorous generalization to bridge the gaps between theories and algorithms for domain adaptation, which can be seamlessly integrated into adversarial-based methods. Adversarial entropy optimization (AEO) [39] learned domain-invariant features by jointly unifying the minimax entropy with the conditional adversarial network. In the AEO framework, the feature extractor and the domain discriminator are trained through the minimization of the independent samples and maximization of the combined samples. Chen et al. [40] proposed domain adversarial reinforcement learning (DARL) framework to address the partial domain adaptation problem. In the DARL framework, deep Q-learning is employed for the selection of the proper source samples to ensure the positive transfer from the source domain to the target domain. Yu et al. [41] suggested that the decision boundary generated by bi-classifier paradigm would favor the source domain. To tackle this problem, the uneven bi-classifier learning [41] is proposed to refine the decision boundary by leveraging the F-norm of classifier predictions. Mei et al. [42] proposed an automatic loss function search for adversarial domain adaptation (ALSDA) [42] to mitigate the degradation of the domain discriminators caused by the dominating gradients of aligned target samples during training. In the new loss an adjustable hyper-parameter is introduced to re-weight the gradients assigned to target samples.

Li et al. [43] tackle the unsupervised domain adaptation problem from the perspective of adversarial learning and argue that the samples from the target domain can be regarded as the visually limitless, naturally occurring adversarial samples from the source domain. By incorporating the adversarial examples into the training, the model could achieve better generalization ability on the target domain. Though the adversarial examples are both adopted in the previous work and ours, our method is distinct from the previous one. First, conventional adversarial examples are typically ℓ_p-bounded, while our noisy interpolated adversarial samples (nIAS) are generated on the line between the perturbed source data and target data. Compared to the conventional ℓ_p-bounded adversarial examples, our nIAS can be regarded as the transitional data between source domain and target domain. Our new synthetic data could better fuse the semantic information from the target data and bridge the gap between the source domain and target domain. Second, in our work we focus on unsupervised class imbalance domain adaptation in which the class distribution in either source domain or target domain is biased. To tackle the imbalance issue, a dynamic data generation technique is adopted in our work to alleviate the bias. While the previous works focus on the conventional UDA problem, and there is no relevant method to deal with the imbalance issue in these works.

Apart from the traditional domain adaptation methods that assume that both domains share the same class space, recent works [44]–[46] consider some more realistic settings such as open-set domain adaptation (OSDA) and universal domain adaptation (UniDA). In OSDA the target domain includes the unique categories which are unseen in the source domain while in UniDA both domains contain the private classes. Li et al. [44] pointed out the limitation of current open-set recognition benchmarks and suggested that the ambiguous definitions of semantic classes in the current datasets hinder the progress in this field. As a result, the fine-grained visual categorization (FGVC) [44] datasets that provide more clear definitions of semantic classes are proposed. Furthermore, Li et al. [44] proposed the weighted discriminative adversarial network with dual classifiers (WDAN) framework [44] which aims to recognize unknown classes by learning the domain-specific discriminative information via adversarial training. To tackle the challenges in universal domain adaptation, a simple GAN-based architecture [45] is proposed to transform features into the latent representations that contain domain-relevant information. Besides, Lv et al. [45] proposed new objectives to enable the discriminator to distinguish the representations of common classes and private classes in both domains. Different from the works aforementioned, our work focused on class-imbalanced unsupervised domain adaptation in which the class distributions of both domains are imbalanced and we proposed new adversarial training to generate the synthetic samples to refine the biased decision boundary during the training.

B. Class-imbalanced Unsupervised Domain Adaptation

As a branch of domain adaptation, the class-imbalanced unsupervised domain adaptation (CDA) aims to deal with data with biased class distribution. This task is more challenging

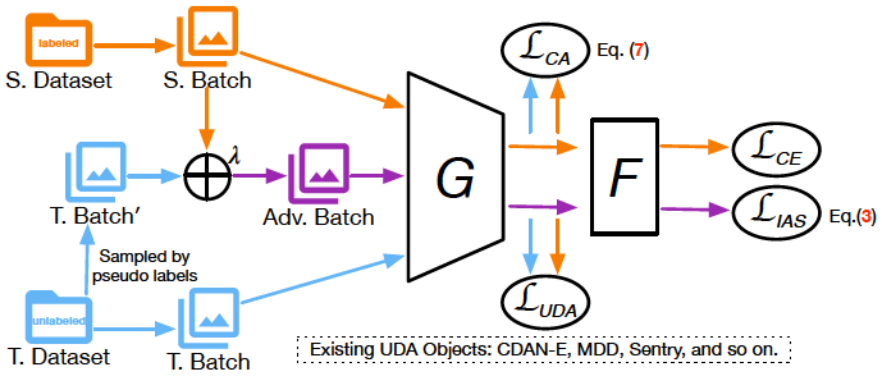


Fig. 2: Illustration of Pairwise Adversarial Training (PAT). Proposed PAT consists of a feature extractor G and an optimal classifier F. It also includes four losses: interpolated adversarial samples loss $\mathcal{L}_{IAS}$, centroid alignment loss $\mathcal{L}_{CA}$, cross-entropy loss $\mathcal{L}_{CE}$, and unsupervised domain adaptation loss $\mathcal{L}_{UDA}$ which represents existing unsupervised domain adaptation methods. λ is the learnable hyper-parameter in Eq. 2 which controls the generation of the adversarial samples and dynamically updated by maximizing $\mathcal{L}_{IAS}$.

than conventional unsupervised domain adaptation (UDA) since it is more difficult to align the minority classes from two domains than that of majority classes. One of the first CDA methods [14] is proposed by exploiting the pseudo labelled target samples to reduce the negative effect of label shift. Wu et al. [13] proposed the asymmetrically-relaxed distance as replacement of the standard one under biased label distribution. Jiang et al. [15] adopted the implicit sampling strategy to ensure class alignment at the minibatch level. The previous work [47] avoided the use of highly unreliable pseudo labels by assessing the reliability of target samples with predictive consistency under random image transformations. Apart from the prior generalization bounds theory [48] on the unsupervised domain adaptation, The early works [1], [49], [50] separately provided the generalization bound for the CDA problem. Specifically, the former work [1] is the first to propose the generalization bound for the class-imbalanced unsupervised domain adaptation. Their generalization bound theory suggests that the error bound only depends on the complexity of the function class. Another work [49] suggested there exists a domain-invariant feature representation between source and target domains under generalized label shift (GLS). Under the GLS condition, the error bound can be decomposed into the sum of balanced error rate and conditional error gap. Lipton et al. [50] further proposed e Black Box Shift Estimation (BBSE) to detect the unknown label distribution of the target data and adjust the model accordingly.

C. Adversarial Training

Adversarial training (AT) [51]–[53] is an effective regularization method for enhancing the robustness and generalization ability of deep learning models. In particular, adversarial samples are incorporated in the model training process, which are intentionally designed to deceive the deep learning model by adding small perturbation on the original samples. However, extensive research [54]–[57] has revealed that adversarial training may compromise the generalization accuracy of the neural network and is also susceptible to overfitting. Recently, some work has made some modifications to the conventional adversarial training proposed by [53]. For instance, PAT [58] attempted to alleviate the harm of adversarial training to the generalization ability of the models by putting an early stop to the search of the adversarial examples. TRADES [56] resolved the conflict between natural accuracy and robustness by making a trade-off between the classification error and boundary error.

Apart from the application of the adversarial training in supervised learning, some work [59], [60] extend the adversarial training to unsupervised training. The previous work [59] suggested that both the natural accuracy and the robustness can be improved by learning from more unlabelled data and they proposed the RST framework to jointly combine the adversarial training with the self-training method. Virtual adversarial training (VAT) [60] seeks the adversarial direction for regularization without using label information. Both AT and VAT have been employed to tackle the standard UDA problems [61]. However, to the best of our knowledge, our work is the first attempt to address the class-imbalanced domain adaptation problem using adversarial training.

III. Proposed Approach

In this section, we start by introducing the CDA problem and presenting notations used throughout the paper as summarized in Table I in Section III-A. Then, we introduce the pairwise adversarial training (PAT) in Section III-B, including the interpolated adversarial sample generation in Section III-B1, centroid alignment in Section III-B2 and the loss function of PAT in Section III-B3. In Section III-C we introduce the noisy pairwise adversarial training (nPAT), which is an extended version of the original PAT. Finally, we theoretically analyse the proposed method in Section III-D.

A. Problem Definition

In class-imbalanced domain adaptation, both the source and target domains suffer from label distribution shift. We are given a source domain $\mathcal{D}_S = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ with N^s labelled samples and a target domain $\mathcal{D}_T = \{x_i^t\}_{i=1}^{N_t}$ with N^t

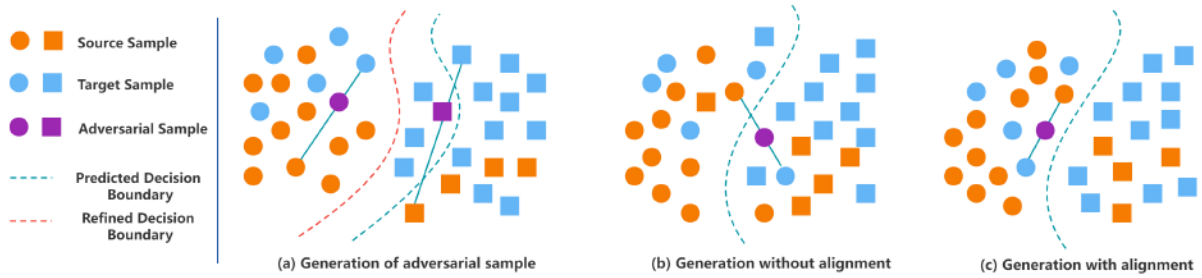


Fig. 3: (a) Illustration of interpolated adversarial samples, which lies on the interpolated line of the source and target samples with the same semantic information. Adversarial samples can alleviate the bias of the decision boundary. (b) Without centroid alignment, adversarial samples may easily violate the decision boundary. (c) With centroid alignment. the problem of violating the decision boundary can be alleviated.

unlabelled samples. Each domain contains K classes, and the class label is denoted as $y^s \in \{1, 2, ..., K\}$. Let p and q denote the probability distributions of the source and target domains, respectively. We assume that both the covariate shift (i.e., $p(x) \neq q(x)$) and label distribution shift (i.e., $p(y) \neq q(y)$ and $p(x \mid y) \neq q(x \mid y)$) exist in two domains. The model typically consists of a feature extractor $g : \mathcal{X} \to \mathcal{Z}$ and a classifier $f : \mathcal{Z} \to \mathcal{Y}$. The predicted label $\hat{y} = f(g(x))$ and empirical risk is defined as $\epsilon = \Pr_{x \sim \mathcal{D}}(\hat{y} \neq y)$, where y is ground-truth label. The source error and target error are denoted as ϵ_S and ϵ_T, respectively. Our goal is to train a model that can reduce gap between source and target domains and minimize ϵ_S and ϵ_T under label distribution shift.

B. Pairwise Adversarial Training (PAT)

We investigate how to mitigate the challenging issue of label distribution shift in CDA, as illustrated in Figure 1. Previous study [14] finds that the performance of the model on the target domain significantly drops when the source and target domains are imbalanced. An intuitive solution is to augment the training data in two domains, such that the model training would not be dominated by the majority classes in either domain. However, this task is not trivial, considering the mixed effects of domain gap and imbalanced class distributions.

Inspired by adversarial training, we aim to alleviate the imbalanced problem in source and target domains by generating adversarial samples. In adversarial training [51], [52], the adversarial samples are exploited to enhance the robustness and generalization of the model. The loss function of adversarial training is:

$$\begin{aligned} &\mathcal{L}_{ce}(x + \delta^*, y; \theta) \\ \text{where} \quad &\delta^* := \arg\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}_{ce}(x + \delta, y; \theta), \end{aligned} \tag{1}$$

where x is the original sample, y is the ground-truth label of x, θ refers to model parameters, and δ is the perturbation added to x.

However, directly applying adversarial training to address the CDA problem is not feasible. First, existing methods [51], [52] only generate adversarial samples from the neighborhood of the original samples without considering the domain gap between source and target domains. Second, these methods fail to handle the class imbalance issue without regard to the ratio of majority classes to minority classes. In this paper, we propose pairwise adversarial training (PAT), as shown in Figure 2, to jointly alleviate the class imbalance problem by dynamically generating adversarial samples from the linear interpolation of source and target pairwise samples (Section III-B1), and reduce domain discrepancy by explicitly aligning the conditional feature distributions of source and target domains (Section III-B2).

However, directly applying adversarial training to address the CDA problem is not feasible. First, existing methods [51], [52] only generate adversarial samples from the neighborhood of the original samples without considering the domain gap between source and target domains. Second, these methods fail to handle the class imbalance issue without regard to the ratio of majority classes to minority classes. In this paper, we propose pairwise adversarial training (PAT) and its extended version noisy pairwise adversarial training (nPAT), as shown in Figure 2, to jointly alleviate the class imbalance problem by dynamically generating adversarial samples from the linear interpolation of source and target pairwise samples (Section III-B1), and reduce domain discrepancy by explicitly aligning the conditional feature distributions of source and target domains (Section III-B2).

1) Interpolated Adversarial Samples Generation: As shown in Figure 3 (a), we generate adversarial samples by linearly interpolating pairwise source and target samples from the same class. The interpolated adversarial samples (IAS) should have the same semantic meaning as its corresponding source and target samples. We explicitly address the data imbalance issue in the source domain by dynamically exploiting interpolated adversarial samples. As a result, the generalization of the unbiased model is improved and the data imbalance issue in the target domain could be implicitly addressed. For the k-th class, the interpolated adversarial samples can be defined as:

$$\begin{aligned} \mathcal{X}_k^{adv} = \quad &\{x_i^{adv} \mid x_i^{adv} = x_i^s + \lambda(x_i^t - x_i^s), \lambda \in [0, 1)^C, \\ &y_i^s = \hat{y}_i^t = k\}, \end{aligned} \tag{2}$$

where $x_i \in \mathbb{R}^{C \times H \times W}$, λ is the learnable hyper-parameter measuring the contributing weights of the source and target samples from the same class. $\hat{y}_i^t$ is the pseudo label of the

target sample assigned by the optimal classifier F. It is used for the match of the corresponding source sample. Although we adopt pseudo labels to generate adversarial samples, the proposed PAT is robust to the potential error accumulation issue as it is robust to unreliability of pseudo labels from two aspects: (1) The misclassified target samples often exist at the decision boundary. Even though the pairwise source and target samples are not actually correct, the generated adversarial samples still maintain the same semantic information as corresponding source samples. (2) The generated adversarial samples are dynamically produced, as the model gradually converges, the adverse effect of bad adversarial samples could be mitigated. In our method, unlike the previous work [14], [15] in which the pseudo labels of the data are not directly utilized in the training pipeline, the pseudo labels of the target data is implicitly utilized for generation of our noisy interpolated adversarial examples (nIAS).The primary concern for the pseudo labels is the mismatch of the semantic information assigned by pseudo labels with that of the ground-truth labels of the data. However, since our nIAS incorporates partial semantic information from the source data. As a result, the mismatch can be diminished and errors can be considerably alleviated. Besides, We have proven the robustness of PAT in terms of unreliable pseudo labels in ablation studies. Note that in our method, not all the classes have equal chance to generate the adversarial samples. We adopt the probability threshold P_k to control the generation of the adversarial sample from a pair of source and target samples of kth class. The details of the probability threshold are discussed in Section III-B2.

When the label distribution of the data is imbalanced, the previous works [62] suggest that one of the heuristic and effective methods is data augmentation. However, these methods, such as SMOTE [62]–[65], might not be effective when there is a domain gap between the source data and target data. Compared to the previous data augmentation methods [62]–[65], Our proposed method possesses some advantages for the class-imbalanced domain adaptation. First, similar to the previous methods [62], [63], our method can augment the data from the minority classes and make the label distribution of the training data less biased. As a result, the biased decision boundary can be refined with more balanced training data. Second, since our synthetic data contains the semantic information from both source domain and target domain. the complexity of our training data could be boosted and the decision boundary could also benefit from the enhanced data complexity. Last, by smoothing the output distribution of the interpolated synthetic data,the domain gap between source data and target data can be minimized and the model can achieve better generalization on the target domain.

The generation of interpolated adversarial samples can be achieved by solving the following optimization problem:

$$\begin{aligned} \mathcal{L}_{IAS} &:= \mathcal{L}_{CE}(\hat{x}^{adv}, y; \theta) \\ \text{where} \quad \hat{x}^{adv} &= \underset{x^{adv} \in \mathcal{X}^{adv}}{\arg\max}\ \mathcal{L}'_{CE}(x^{adv}, y; \theta). \end{aligned} \tag{3}$$

The outer minimization problem involves the standard cross-entropy loss $\mathcal{L}_{CE}$, i.e.,

$$\mathcal{L}_{CE}(\hat{x}^{adv}, y; \theta) = -\log(\sigma_y(f(g(\hat{x}^{adv})))), \tag{4}$$

Algorithm 1 Optimization Algorithm for Eq. 3

Input: Source samples $\{(x_i^s, y_i^s)\}_{i=1}^{N_s}$,target samples with pseudo labels $\{(x_i^t, \hat{y}_i^t)\}_{i=1}^{N_t}$, probability threshold for each class $\{P_k\}_{k=1}^{K}$.
Output: Adversarial samples $\{\hat{x}_i^{adv}\}_{i=1}^{N_{adv}}$

```
for Each source sample x_i^s do
  if rand() > P_k(k = y_i^s) then
    Randomly select a target sample x_i^t where ŷ_i^t = y_i^s.
    Initialize λ and generate x^adv by Eq. 2.
    repeat
      Apply Eq. 5 to get g_λ = ∇_λ L'_CE(x^adv, y_i^s; θ).
      Update λ ← λ + α g_λ.
      Apply Eq. 2 to update x^adv with new λ.
    until the optimization converges
  end if
end for
```

where σ is the softmax function.

For the inner maximization problem in Eq. 3, we use a modified cross-entropy loss $\mathcal{L}'_{CE}$ proposed by [66]. The modified loss can alleviate the problem of gradient exploding or vanishing when the entropy loss is maximized. It is written as:

$$\mathcal{L}'_{CE}(x^{adv}, y; \theta) = \log(1 - \sigma_y(f(g(x^{adv})))). \tag{5}$$

Several optimization algorithms, such as the fast gradient sign method [52] and projected gradient descent [53], have been proposed for adversarial training. However, these algorithms aim to obtain the adversarial samples in the ℓ_p ball of the original samples, which cannot be directly applied to solve our problem. In our case, the generated adversarial samples are confined by the interpolated line of source and target samples. We propose a new optimization algorithm to solve the inner maximization optimization in Eq. 3 by initializing the interpolated adversarial samples with random λ and updating λ by back propagation in each iteration. The main procedures are summarized in Algorithm 1.

2) Class-Imbalanced Semantic Centroid Alignment: Without careful control of the generation mechanism, the interpolated adversarial samples may not alleviate the issue of imbalanced class distribution. Moreover, although the interpolated adversarial samples bridge the source and target domains to some extent, the discrepancy between source and target domains is not explicitly reduced. To address these issues, we propose to explicitly align the source and target domains with imbalanced class distributions using two strategies.

First, we propose a strategy to guide the generation of interpolated adversarial samples. For training samples in each mini-batch from the source domain, they should not have the equal opportunity to generate interpolated adversarial samples. Since the decision boundary usually favors the majority classes, the probability of generating adversarial samples for minority classes should be larger than that for majority classes. For the k-th class, we set a probability threshold P_k as follows:

$$P_k = \frac{n_k}{n_{max} + \tau}, \tag{6}$$

where n_k is the number of the samples from the kth class. $n_{max} = \max_k\{n_k\}_{k=1}^{K}$, and τ is the bias. For a specific class, if a random number $r \in [0, 1)$ is larger than the corresponding threshold, the adversarial sample is generated, as shown in Algorithm 1. We also adopt class-balanced sampling on the source samples to alleviate the biased occurrence of the majority classes. Specifically, each class is selected with an equal chance, in order to reduce the model prediction bias towards the majority classes.

Second, we incorporate the moving average centroid alignment [67] to align the conditional feature distributions of source and target domains by explicitly matching the centroids of two domains. As illustrated in Figure 3b, without centroid alignment, the adversarial samples may be generated from a pair of samples in which one of the samples is misaligned to other classes, thus making the embedding of adversarial samples fall out of the decision boundary. With centroid alignment as illustrated in Figure 3c, we can eliminate the occurrence of such out-of-bound adversarial samples, and the interpolated adversarial samples could provide meaningful support for the minority class in the target domain. The loss function of moving average centroid alignment is defined as

$$\mathcal{L}_{CA} = \sum_{k=1}^{K} \text{dist}(\mathcal{C}_k^s, \mathcal{C}_k^t), \tag{7}$$

where $\mathcal{C}_k^s$ and $\mathcal{C}_k^t$ denote the centroids of the kth class in the source and target domains, respectively. dist() can be implemented by the Euclidean distance or cosine distance.

3) PAT for Class Imbalanced Domain Adaptation: The proposed PAT approach can be integrated with many existing unsupervised domain adaptation (UDA) frameworks to enhance their performance on the class-imbalanced domain adaptation (CDA) problem. In this paper, we adopt CDAN-E [37], MDD [38] and Sentry [47] as the UDA models, respectively. As shown in Figure 2, the PAT framework includes a feature extractor G and an optimal classifier F. There are four losses: (1) interpolated adversarial samples loss $\mathcal{L}_{IAS}$ which aims to dynamically generate adversarial samples to alleviate imbalance issue, (2) centroid alignment loss $\mathcal{L}_{CA}$ is designed to align the conditional feature distributions of source and target, (3) standard cross-entropy loss $\mathcal{L}_{CE}$, and (4) unsupervised domain adaptation loss $\mathcal{L}_{UDA}$ which is adopted from existing UDA methods, e.g., CDAN-E [37], MDD [38], and Sentry [47]. The overall loss function is defined as:

$$\mathcal{L}_{PAT} = \mathcal{L}_{UDA} + \mathcal{L}_{CE} + \alpha\mathcal{L}_{IAS} + \beta\mathcal{L}_{CA}, \tag{8}$$

where α and β are two trade-off parameters.

C. Noisy Pairwise Adversarial Training (nPAT)

Generating interpolated samples has been explored in literature, such as mix-up [18] and its variants [68], [69]. However, Our interpolated strategy is significantly different from mix-up and its variants. First, our method addresses the imbalance problem in domain adaptation which is hardly touched by mix-up based methods. Second, our generated adversarial samples are based on pairwise source and target samples with same semantic information and the interpolated ratio is adaptively updated. While mix-up based methods generate new samples by randomly merging two samples from the same domain with a fixed interpolated ratio.

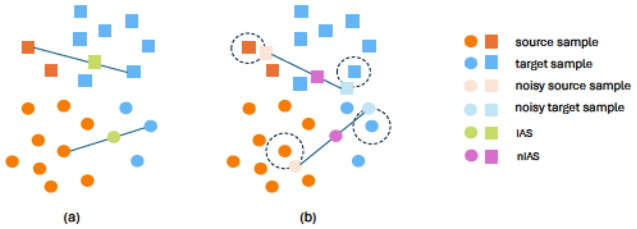


Fig. 4: The generation of (a) interpolated adversarial samples (IAS) and (b) noisy interpolated adversarial samples (nIAS). The nIAS is built on the convex combination of a pair of noise-perturbed input samples.

As one of the latest follow-up work of mixup [18], Noisy Feature Mixup (NFM) [20] was recently proposed to introduce the noise injection scheme in the generation of the interpolated samples, In this work the new synthetic data is generated from convex combinations of pairs of noise-perturbed inputs and embedding features. With the effect of regularization from both mixup [18] and noise injection, NFM [20] is supposed to improve the generalization and robustness of the model. Inspired by this method, we propose noisy pairwise adversarial training (nPAT) in which the interpolated adversarial samples (IAS) are replaced by noisy interpolated adversarial samples (nIAS). The main difference between nIAS and IAS is illustrated in Figure 4. It is evidently illustrated that noisy interpolated adversarial samples are not confined in the linear interpolation space of pairs of raw inputs. Compared to IAS, the nIAS can provide more diversity of the semantic meaningful adversarial samples, which is supposed to better refine the biased decision boundary.

Specifically, the nIAS can be generated by two steps. First, we generate the intermediate samples x^{int_adv} from the pairs of source and target samples.

$$\begin{aligned}\mathcal{X}_k^{int\_adv} = \quad & \{x_i^{int\_adv} \mid x_i^{int\_adv} = x_i^s + \lambda(x_i^t - x_i^s), \\ & \lambda \in [0, 1)^C, y_i^s = \hat{y}_i^t = k\},\end{aligned} \tag{9}$$

where x_i and λ are the same as Eq. 2. Note that this generation method is equivalent to adding noise on the pairs of input data and then performing the mixup operation on the noise-perturbed input data.

Then we add the noisy perturbation on the intermediate samples to generate the initial noisy interpolated adversarial samples.

$$\begin{aligned}\mathcal{X}_k^{nadv} = \quad & \{x_i^{nadv} \mid x_i^{nadv} = (\mathbb{I} + \sigma_{mult}\xi_k^{mult}) \odot x_i^{int\_adv} \\ & + \sigma_{add}\xi_k^{add}\},\end{aligned} \tag{10}$$

where $\odot$ denotes Hadamard product, $\mathbb{I}$ denotes the vector with all components equal to one. ξ_k^{mult} and ξ_k^{add} are random variables that model the multiplicative and additive noise, respectively. σ_{mult} and σ_{add} are pre-specified noise levels for multiplicative additive and additive noise.

The final interpolated adversarial samples can also be achieved by solving the same optimization problem as Eq. 3.

$$\begin{aligned} \mathcal{L}_{nIAS} &:= \mathcal{L}_{CE}(\hat{x}^{nadv}, y; \theta), \\ \hat{x}^{nadv} &= \underset{x^{nadv} \in \mathcal{X}^{nadv}}{\arg\max} \mathcal{L}'_{CE}(x^{nadv}, y; \theta), \end{aligned} \tag{11}$$

where $\mathcal{L}_{CE}$ and $\mathcal{L}'_{CE}$ are cross-entropy loss and modified cross-entropy loss defined in Eq. 4 and Eq. 5, respectively. And the overall loss function for noisy pairwise adversarial training (nPAT) is defined as:

$$\mathcal{L}_{nPAT} = \mathcal{L}_{UDA} + \mathcal{L}_{CE} + \alpha \mathcal{L}_{nIAS} + \beta \mathcal{L}_{CA}, \tag{12}$$

where α and β are two trade-off parameters.

D. Theoretical Justifications

In this subsection, we adopt the generalization bound theory proposed in [49] to justify why our method would be able to address the CDA problem. First, we introduce two definitions: Balanced Error Rate and Conditional Error Gap. Then we present the error decomposition theorem and discuss how our method helps in reducing the error with some theoretical insights.

Definition 1: (Balanced Error Rate) [49]. The balanced error rate of the classifier on $\mathcal{D}_S$ is

$$BER_{\mathcal{D}_S}(\hat{y} \| y) := \max_{j \in \mathcal{Y}} \mathcal{D}_S(\hat{y} \neq y \mid y = j). \tag{13}$$

Definition 2: (Conditional Error Gap) [49]. Given a joint distribution $\mathcal{D}$, the conditional error gap of a classifier is

$$\Delta_{CE}(\hat{y}) := \max_{i \neq j \in \mathcal{Y}^2} \mid \mathcal{D}_S(\hat{y} = i \mid y = j) - \mathcal{D}_T(\hat{y} = i \mid y = j) \mid \tag{14}$$

Theorem 1: (Error Decomposition Theorem) [49]. For any classifier $\hat{y} = (f \circ g)(x)$,

$$\begin{aligned} \mid \epsilon_S(f \circ g) - \epsilon_T(f \circ g) \mid \leq \quad & \|\mathcal{D}_S^y - \mathcal{D}_T^y\|_1 \cdot BER_{\mathcal{D}_S}(\hat{y} \| y) \\ & + 2(K-1)\Delta_{CE}(\hat{y}), \end{aligned} \tag{15}$$

where $\mathcal{D}_S^y$ and $\mathcal{D}_T^y$ represent label distribution on source and target domains, respectively. $\|\mathcal{D}_S^y - \mathcal{D}_T^y\|_1 := \sum_{i=1}^{K} \mid \mathcal{D}_S(y = i) - \mathcal{D}_T(y = i) \mid$ is the L_1 distance between $\mathcal{D}_S^y$ and $\mathcal{D}_T^y$.

The upper bound in Theorem 1 suggests the error gap can be decomposed into two terms: the weighted L_1 distance of source and target marginal label distributions and conditional error gap. Since $\|\mathcal{D}_S^y - \mathcal{D}_T^y\|_1$ is only determined by the dataset itself, the error gap is only affected by the balanced error rate and conditional error gap. When the label shift occurs on the source and target domains, the upper bound will be increased for two reasons. First, the classifier favors the majority classes, which causes higher error rates on minority classes. Second, the label shift makes it hard to align the conditional features between source and target domains, which induces a larger conditional error gap. Our PAT method helps obtain a tight upper bound in Eq. (15) and reduce such errors from two aspects. First, PAT augments the source training data and thus makes the marginal label distribution less biased. Consequently, it can lower the balanced error rate. Second, to tackle the misalignment issue, we adopt semantic centroid alignment strategy to align the conditional features from source and target domains, which can further reduce the conditional error gap. Thus, by incorporating PAT to an existing UDA method, the integrated model would have an improved capability on dealing with data with label shift across the source and target domains.

Similar to the previous work [43], apart from the original source domain D_S, the augmented data generated either by our method or ℓ_p-based augmented method (e.g., FGSM [52] and PGD [70]) can be regarded as a new source domain (i.e., D_{adv}). Given a hypothesis function $h = f \circ g$, suppose the number of total source domains (including the original source domain and the augmented domain) is N^D, the empirical μ-weighted empirical risk over the multiple source domain can be formulated as:

$$\hat{\epsilon}_\mu(h) = \sum_{i=1}^{N^D} \mu_i \epsilon_i(h) = \sum_{i=1}^{N^D} \frac{\mu_i}{N_i^D} \sum_{x \in D_i} |h(x) - y_i|, \tag{16}$$

where μ is the coefficient and N_i^D denotes the number of samples in ith domain. According to the previous work [71], the generalization error bound of the hypothesis function h on multiple source domain can be formulated as following,

Theorem 2: (Generalization bound with multiple source domains) [71]. Let $\mathcal{H}$ be a hypothesis space of VC dimension d. For each $j \in \{1, 2, ..., N^D\}$,let $\mathcal{D}_j$ be labeled sample of size $\gamma_j m$ generated by drawing $\gamma_j m$ points from $\mathcal{D}_j$ and labeling them according to f_j. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\hat{\epsilon}_\mu(h)$ for a fixed weight vector μ on these samples and $h^* = \min_{h \in \mathcal{H}} \epsilon_T(h)$ is the target error minimizer, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\begin{aligned} \epsilon_T(\hat{h}) \leq \epsilon_T(h_T^*) + 4 \sqrt{(\sum_{j=1}^{N^D} \frac{\mu_j^2}{\gamma_j})(\frac{d \log(2m) - \log(\delta)}{2m})} \\ + \sum_{j=1}^{N^D} \mu_j (2\eta_j + d_{\mathcal{H}\Delta\mathcal{H}}(D_j, D_T)), \end{aligned} \tag{17}$$

where $\eta_j = \min_{h \in \mathcal{H}} \{\epsilon_T(h) + \epsilon_j(h)\}$. $m = \sum_{j=1}^{N^D} N_j^D$ and $\gamma_j = \frac{N_j^D}{m}$. The $d_{\mathcal{H}\Delta\mathcal{H}}$ denotes the $\mathcal{H}\Delta\mathcal{H}$-divergence of two domains and it is expressed as:

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) = 2 \sup_{h, h' \in \mathcal{H}} |P_{x \sim \mathcal{D}_S}(h(x) \neq h'(x)) - \\ P_{x \sim \mathcal{D}_T}(h(x) \neq h'(x))|. \end{aligned} \tag{18}$$

According to Theorem 2, the generalization error of the hypothesis h on the target domain is associated with the divergence of the source domain and the target domain. Compared to the ℓ_p-based augmentation methods that generate the new data within the ℓ_p-ball of the source data, the augmented data synthesized by our method would incorporate the semantic information from the target data. As a result, the $\mathcal{H}\Delta\mathcal{H}$-divergence between our data and the target data is smaller than that of the data generated by ℓ_p-based augmentation methods. Thus, under the same condition, our method could achieve a smaller error gap than that of ℓ_p-based augmentation methods.

TABLE II: Per-class average accuracy(%) on Office-Home dataset with RS→UT label shift. nPAT is the method proposed in this paper. Each task is repeated 5 times for PAT and nPAT. The average and the variance are reported.

Method	Rw→Pr	Rw→Cl	Pr→Rw	Pr→Cl	Cl→Rw	Cl→Pr	AVG
source †	70.74	44.24	67.33	38.68	53.51	51.85	54.39
BSP [ICML 2019] [72] †	72.80	23.82	66.19	20.05	32.59	30.36	40.97
F-DANN [ICML 2019] [13] †	58.56	40.57	67.32	37.33	55.84	53.67	53.88
COAL [ECCV 2020] [14] †	73.65	42.58	73.26	40.61	59.22	57.33	58.40
InstaPBM [Arxiv 2020] [73] †	75.56	42.93	70.30	39.32	61,87	63.40	58.90
CDAN-E [NeurIPS 2018] [37] †	70.78	45.94	69.72	40.40	53.82	54.86	55.92
CDAN-E+PAT [KDD 2022] [19]	78.00 (1.50)	46.01 (0.77)	75.36 (0.43)	41.84 (1.75)	63.10 (0.66)	63.64 (0.68)	61.32
CDAN-E+nPAT (Ours)	**79.34** (0.66)	**49.86** (0.37)	**75.51** (1.21)	**44.25** (0.92)	**63.89** (1.24)	**64.46** (0.74)	**62.88**
MDD [ICML 2019] [38] †	71.21	44.78	69.31	42.56	52.10	52.70	55.44
MDD+implicit [ICML 2020] [15] †	76.08	50.04	74.21	45.38	61.15	63.15	61.67
MDD+PAT [KDD 2022] [19]	79.30 (0.64)	54.10 (0.33)	76.87 (0.46)	49.92 (0.72)	**67.38** (0.18)	67.23 (0.71)	65.80
MDD+nPAT (Ours)	**80.12** (0.34)	**55.40** (0.67)	**78.00** (0.24)	**50.58** (1.19)	66.43 (0.76)	**68.86** (0.84)	**66.56**
Sentry [ICCV 2021] [47] †	76.12	56.80	73.60	54.75	65.94	64.29	65.25
Sentry+PAT [KDD 2022] [19]	79.32 (0.91)	63.44 (0.56)	78.11 (1.08)	63.07 (1.24)	69.17 (0.46)	70.30 (0.70)	70.56
Sentry+nPAT (Ours)	**84.44** (0.16)	**67.39** (0.52)	**79.63** (0.96)	**64.37** (0.80)	**72.52** (0.80)	**73.78** (0.92)	**73.68**

† Data of the baseline methods are cited from [19]

TABLE III: Per-class average accuracy(%) on minority classes from imbalanced Office-Home (Cl→Pr).

Method	Batteries	Bed	Bike	Bottle	Calculator	Chair	Clipboards	AVG
MDD †	45.16	0.0	93.18	50.00	70.37	72.91	0.0	47.37
MDD+PAT †	51.61	65.11	95.45	66.66	81.48	72.91	6.17	62.77
MDD+nPAT	**62.90**	**72.09**	**97.02**	**70.00**	**83.33**	**72.91**	**8.69**	**66.70**

† Data of the baseline methods are cited from [19]

TABLE IV: Per-class average accuracy(%) on DomainNet dataset. nPAT is the method proposed in this paper. Each task is repeated 5 times for PAT and nPAT. The average and the variance are reported.

Method	R→C	R→P	R→S	C→R	C→P	C→S	P→R	P→C	P→S	S→R	S→C	S→P	AVG
source †	65.75	68.84	59.15	77.71	60.60	57.87	84.45	62.35	65.07	77.10	63.00	59.72	66.80
F-DANN [ICML 2019] [13] †	66.15	71.80	61.53	81.85	60.06	61.22	84.46	66.81	62.84	81.38	69.62	66.50	69.52
BSP [ICML 2019] [72] †	67.29	73.47	69.31	86.50	67.52	70.90	86.83	70.33	68.75	84.34	72.40	71.47	74.09
COAL [ECCV 2020] [14] †	73.58	75.37	70.50	89.63	69.98	71.29	89.81	68.01	70.49	87.97	73.21	70.53	75.89
InstaPBM [Arxiv 2020] [73] †	80.10	75.87	70.84	89.67	70.21	72.76	89.60	74.41	72.19	87.00	79.66	71.75	77.84
CDAN-E [NeurIPS 2018] [37] †	75.95	75.82	73.60	85.55	70.17	70.50	86.06	68.91	71.51	86.44	78.28	70.34	76.09
CDAN-E+PAT [KDD 2022] [19]	77.81 (1.39)	76.36 (0.89)	73.04 (0.80)	88.53 (1.17)	71.36 (0.58)	**76.37** (1.35)	89.87 (0.50)	79.80 (1.37)	73.77 (1.39)	89.79 (0.36)	83.04 (0.48)	**75.19** (1.30)	79.58
CDAN-E+nPAT (Ours)	**79.83** (2.17)	**76.58** (1.05)	**74.90** (0.39)	**88.97** (0.52)	**73.55** (1.00)	75.87 (0.83)	**89.97** (1.02)	**79.96** (1.17)	**77.02** (0.76)	**90.60** (0.81)	**83.21** (0.17)	74.91 (1.02)	**80.44**
MDD [ICML 2019] [38] †	71.89	72.47	66.92	86.18	66.55	66.71	86.29	67.41	68.37	84.96	71.97	68.22	73.16
MDD+implicit [ICML 2020] [15] †	75.54	74.30	70.02	88.17	70.50	70.30	87.94	72.03	72.29	88.85	76.12	71.21	76.44
MDD+PAT [KDD 2022] [19]	79.59 (1.02)	76.99 (0.82)	76.20 (0.91)	88.89 (0.26)	**72.36** (1.02)	75.52 (0.63)	88.70 (0.30)	**79.18** (1.03)	76.42 (0.32)	89.02 (0.51)	80.65 (0.48)	75.41 (0.16)	79.91
MDD+nPAT (Ours)	**79.64** (0.95)	**77.07** (0.16)	**76.41** (0.78)	**89.15** (0.56)	72.06 (0.70)	**76.66** (0.29)	**88.96** (0.36)	78.52 (0.96)	**77.84** (0.38)	**89.42** (0.79)	**80.69** (0.72)	**76.33** (0.71)	**80.22**
Sentry [ICCV 2021] [47] †	83.89	76.72	74.43	90.61	76.02	79.47	90.27	82.91	75.60	90.41	82.40	73.98	81.39
Sentry+PAT [KDD 2022] [19]	86.94 (0.50)	78.64 (0.88)	**80.47** (0.65)	**91.13** (0.38)	77.97 (0.19)	81.21 (0.32)	91.51 (0.50)	85.55 (0.84)	79.74 (1.68)	**91.94** (0.91)	84.97 (0.53)	77.71 (0.81)	83.98
Sentry+nPAT (Ours)	**87.61** (0.64)	**80.36** (1.20)	79.55 (0.81)	90.89 (0.14)	**79.09** (0.30)	**82.91** (0.77)	**91.56** (0.33)	**86.66** (0.81)	**81.07** (1.79)	91.88 (0.87)	**87.38** (0.42)	**78.89** (0.68)	**85.36**

† Data of the baseline methods are cited from [19]

IV. EXPERIMENTS

A. Experimental Settings

Datasets: We leverage three datasets from the domain adaptation field. **(1) Office-31** is a widely used benchmark image dataset for domain adaptation [74]. It contains 31 classes in three domains: Amazon (**A**), Dslr (**D**) and Webcam (**W**). The standard Office-31 doesn't exhibit obvious label distribution shift (LDS), so a new imbalanced Office-31 is created by sampling from standard one as suggested by [14]. The distribution conforms to Paredo distribution [75] and follows the **R**eversely-unbalanced **S**ource and **U**nbalanced **T**arget (**RS-UT**) protocol. Both the source and target domains have shifted label distributions, and the label distribution of source domain is a reversed version of that of target domain. **(2) Office-Home** is a large benchmark dataset containing 65 classes of objects commonly found in office and home scenarios [76]. It has four domains: Real-World (**Rw**), Clipart (**Cl**), Product (**Pr**) and Art (**Ar**). In our experiments, we use the existing imbalanced Office-Home with **RS-UT** distributions generated in [14] to train and test our approach. Since there are very limited samples in the art (**Ar**) domain, we only conduct domain adaptation tasks on the other three domains. **(3) DomainNet** is a large-scale benchmark dataset for domain adaptation [11]. Since there are mislabeled samples in some classes and domains, we follow [14] and adopt only 40 common classes from four domains: Real (**R**), Clipart (**C**), Painting (**P**), and Sketch (**S**). Different from Office-31 and Office-Home, the selected samples in DomainNet already exhibit obvious label distribution shift in the source and target domains. So there is no need to sample this dataset again. Figure 6 illustrates the label distribution of imbalanced Office-31 datasets.

Baselines: We choose three UDA models: CDAN-E [37], MDD [38] and Sentry [47] and integrate them with our nPAT. We compare with two main streams of the state-of-the-art methods: (1) *Class-imbalanced domain adaptation methods*, namely Sentry [47], MDD+implicit [15], COAL [14], and F-DANN [13]. (2) *Unsupervised Domain Adaptation methods*, namely InstaPBM [73] and BSP [72]. We also compare our nPAT with the previous PAT [19] in the experiment.

Evaluation Metric: We adopt per-class average accuracy to evaluate the performance of all methods.

TABLE V: Per-class average accuracy(%) on Office-31 with RS→UT label shift. nPAT is the method proposed in this paper. Each task is repeated 5 times for PAT and nPAT. The average and the variance are reported.

Method	A→W	D→W	W→D	A→D	D→A	W→A	AVG
source †	71.77	90.86	93.06	72.25	59.03	58.34	74.21
F-DANN [ICML 2019] [13] †	69.83	93.56	93.95	76.45	58.57	58.11	75.07
COAL [ECCV 2020] [14] †	81.18	91.12	95.46	81.67	66.08	66.60	80.35
CDAN-E [NeurIPS 2018] [37] †	76.25	**95.78**	94.85	79.92	64.04	58.69	78.25
CDAN-E+PAT [KDD 2022] [19]	84.04 (0.68)	94.90 (1.26)	94.59 (1.66)	83.53 (1.65)	67.97 (1.59)	65.60 (1.05)	81.77
CDAN-E+nPAT (Ours)	**85.99** (0.42)	94.50 (0.95)	**95.95** (1.58)	**85.09** (1.80)	**69.37** (0.58)	**67.92** (0.45)	**83.13**
MDD [ICML 2019] [38] †	83.99	**96.69**	96.71	83.94	67.23	61.36	81.65
MDD+implicit [ICML 2020] [15] †	85.79	96.20	**97.40**	84.25	68.11	66.63	83.06
MDD+PAT [KDD 2022] [19]	89.25 (1.14)	95.87 (0.78)	96.89 (0.17)	86.79 (0.82)	71.66 (0.57)	**70.26** (1.13)	85.12
MDD+nPAT (Ours)	**90.42** (0.61)	96.27 (0.28)	97.13 (0.12)	**87.68** (0.57)	**73.93** (0.83)	69.92 (1.02)	**85.89**
Sentry [ICCV 2021] [47] †	81.77	90.95	93.50	83.91	62.72	64.00	79.48
Sentry+PAT [KDD 2022] [19]	87.35 (0.54)	94.30 (1.41)	95.22 (0.26)	84.49 (1.83)	68.98 (0.47)	67.41 (0.29)	82.95
Sentry+nPAT (Ours)	**89.80** (0.68)	**94.90** (1.41)	**95.30** (0.28)	**89.02** (1.76)	**72.30** (0.40)	**69.04** (1.73)	**85.06**

† Data of the baseline methods are cited from [19]

TABLE VI: Per-class average accuracy(%) of MDD, MDD+nPAT w/o nIAS, MDD+nPAT w/o CA, and MDD+nPAT on imbalanced Office-Home dataset.

Method	Rw→Pr	Rw→Cl	Pr→Rw	Pr→Cl	Cl→Rw	Cl→Pr	AVG
MDD †	75.96	47.38	71.56	42.73	57.46	58.76	58.97
MDD+nPAT w/o nIAS †	76.59	52.32	76.33	49.59	64.95	65.39	64.19
MDD+nPAT w/o CA †	77.54	51.15	76.70	47.24	**67.24**	64.40	64.04
MDD+nPAT †	**79.57**	**55.53**	**78.00**	**50.40**	66.21	**69.05**	**66.46**

† We adopt class-balanced source sampling on all these methods.

TABLE VII: Per-class average accuracy(%) of MDD, MDD+nPAT w/o nIAS, MDD+nPAT w/o CA, and our full MDD+nPAT on three tasks from DomainNet dataset.

Method	R→C	C→P	C→R	AVG
MDD †	75.30	70.38	87.94	77.87
MDD+nPAT w/o nIAS †	76.61	70.25	88.13	78.33
MDD+nPAT w/o CA †	73.42	**72.34**	88.63	78.13
MDD+nPAT (full) †	**78.19**	71.95	**89.20**	**79.78**

† We adopt class-balanced source sampling on all these methods.

TABLE VIII: Per-class average accuracy(%) of MDD, MDD+nPAT w/o nIAS, MDD+nPAT w/o CA, and our full MDD+nPAT on imbalanced Office-31 dataset.

Method	A→W	W→D	W→A	AVG
MDD †	85.86	96.12	65.20	82.39
MDD+nPAT w/o nIAS †	85.65	96.88	66.02	82.85
MDD+nPAT w/o CA †	85.34	97.00	68.04	83.46
MDD+nPAT (full) †	**90.53**	**97.02**	**69.65**	**85.73**

† We adopt class-balanced source sampling on all these methods.

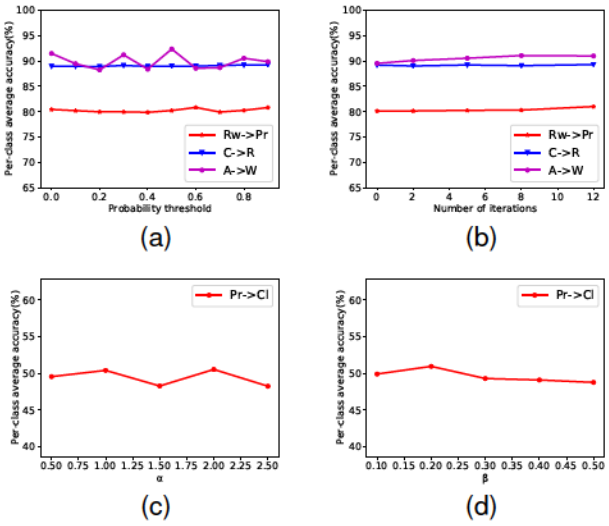


Fig. 5: Per-class average accuracy (%) of MDD+nPAT with different probability thresholds for pseudo labels: (a) on Rw → Pr, C → R and A → W; (b) with different numbers of iterations in the inner maximization of nPAT on Rw → Pr, C → R and A → W; (c) with varying α when $\beta = 0.1$ on Pr →Cl; (d) with varying β when $\alpha = 0.5$ on Pr →Cl.

Implementation Details: We use PyTorch to implement our approach. We train our model with the mini-batch SGD with a Nesterov momentum of 0.9 and a weight decay of 0.0005. The learning rate of classifiers is 10 times larger than that of feature extractor, and all the learning rates are adjusted by every iteration. In order to obtain the interpolated adversarial sample from a pair of source and target samples from the same class, we utilize a memory pool to store the pseudo labels of all the target samples. The pseudo labels are updated in every iteration. Note that we utilize a class-balanced sampler on the source samples, which can be referred to as N-way (number of classes per batch) and K-shot (number of examples per class). The coefficient α is set to 1.0 and β is set to 0.2 for all models.For nPAT, the σ_{mult} and σ_{add} are initialized as 0.2 and 0.4, respectively. All the experiments are implemented on Nvidia RTX A5000 platform.

B. Experimental Results

Table II shows the per-class average accuracy and overall average accuracy of our approach and baselines on the imbalanced Office-Home dataset. First, The experiments show that

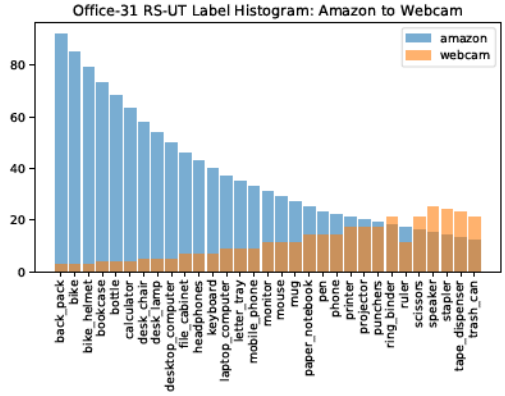


Fig. 6: The biased label distribution shift on Amazon → Webcam from imbalanced Office-31.

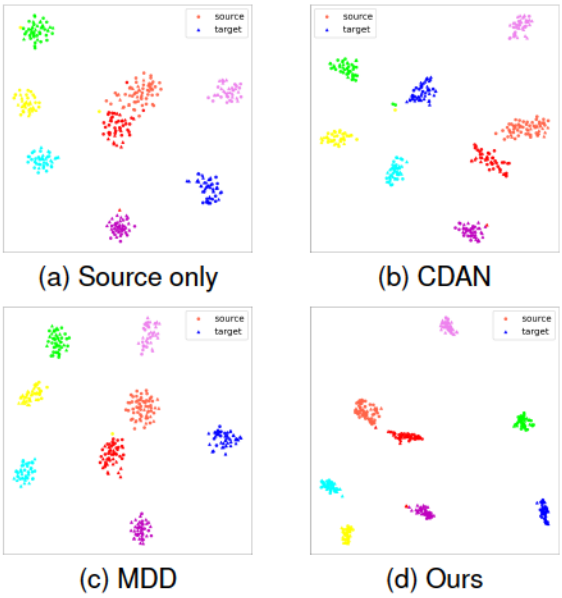


Fig. 7: The t-SNE figure of source only, CDAN, MDD and our method on Rw → Pr task from Office-Home. Circle and triangle represent the source data and target data, respectively. Different colors represent data of different classes.

our method (CDAN-E+PAT(nPAT), MDD+PAT(nPAT) and Sentry+PAT(nPAT)) outperforms their original UDA models on all tasks. The average performance of our method is approximately 8%, 11% and 8% higher than that of their original UDA methods. Besides, the results of nPAT method are better than that of the PAT method. Among them, the Sentry+nPAT achieves best performance and the average accuracy can reach 73.77%. Besides, the results of our method are also higher than that of other standard domain adaptation and CDA specific methods. For instance, the average accuracy of MDD+nPAT is on average 5% higher than that of MDD+implicit [15]. The experimental results validate the effectiveness of our method in dealing with the CDA problem. As we suggest, the improvement comes from the reduced balanced error rate and conditional error gap. To validate our idea, we investigate the accuracy of our MDD+nPAT model on each class and compare the results with the pure MDD [38] and MDD+PAT [19]. Table III shows that our nPAT method has better performance than

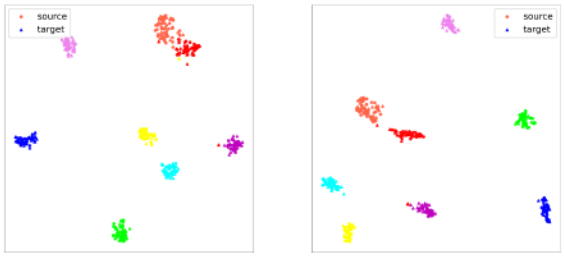


Fig. 8: The t-SNE figure of (a) ℓ_p-based augmentation method and (b) our method on Rw → Pr task from Office-Home. Circle and triangle represent the source data and target data, respectively. Different colors represent data of different classes.

that of MDD and MDD+PAT on the minority classes from the imbalanced Office-Home dataset.

Table IV shows the per-class average accuracy and overall average accuracy of our approach and baselines on the DomainNet dataset. Our method also shows better performance than the baseline methods. The average performance improvement from our method can reach about 4%, 7% and 3% on CDAN-E+nPAT, MDD+nPAT and Sentry+nPAT compared to their original UDA models, respectively. Still the nPAT method outperforms the PAT method on the majority of the tasks. The Sentry+nPAT achieves the best result and can reach 84.83%.

We manually sample the standard Office-31 dataset and construct the imbalanced Office-31 dataset, in which the label distribution conforms to the Paredo distribution [75]. Table V shows the experimental results on the imbalanced Office-31 dataset, which demonstrate that our method still achieves better performance than the baseline models. The improvement can reach an average of 5%, 4% and 6% for CDAN-E+nPAT (nPAT), MDD+nPAT (nPAT) and Sentry+nPAT (nPAT) compared to their original UDA models. The best result on the imbalanced Office-31 comes from MDD+PAT, which can achieve 85.79%.

C. Ablation Studies

We further investigate the performance of our approach from several aspects. First, we evaluate the contribution of each component in the loss function. Second, we investigate how the unreliable pseudo labels affect the performance of our method by varying the probability threshold when selecting the pseudo labels. We also evaluate the sensitivity of the model to the change of the hyper-parameters and the effect of the number of the iterations in the inner maximization of nPAT.

The proposed noisy pairwise adversarial training approach includes two major components, i.e., noisy interpolated adversarial samples (nIAS) and centroid alignment (CA). We choose six tasks from Office-Home and three tasks from DomainNet and Office-31, respectively. Table VI, Table VII and Table VIII show the performance of MDD [38], MDD+nPAT w/o nIAS, MDD+nPAT w/o CA, and our full MDD+nPAT model on the imbalanced Office-Home, DomainNet and imbalanced Office-31. MDD+nPAT w/o nIAS and MDD+nPAT w/o CA consistently achieve better performance than the baseline

model MDD [38], which validates the effectiveness of both components in dealing with the biased label distribution. Our full MDD+nPAT model further improves the classification accuracy, demonstrating the complementary roles of pairwise adversarial samples and centroid alignment in our approach.

Next, We investigate how reliability of the pseudo labels affect our method. For previous methods [14], [15], to ensure the reliability of the pseudo labels, they choose the pseudo labels with high confidence. In our study, we choose pseudo labels with various probability thresholds ranging from 0.0 to 0.9. Figure 5a illustrates the performance of MDD+nPAT on Rw $\rightarrow$ Pr, C $\rightarrow$ R and A $\rightarrow$ W with different probability thresholds for pseudo labels. The results show that on Office-Home and DomainNet, the performance is not very sensitive to the probability threshold. Even if we utilize unreliable pseudo labels under a low probability threshold, the performance of our method still remains. The results validate the robustness of our method to the unreliability of the pseudo labels.

Then, we evaluate the sensitivity of our model to the change of two hyperparameters α and β. In particular, we first set β to 0.2 and choose α from $[0.5, 2.5]$. Then, α is set to 1.0, and β is chosen from $[0.1, 0.5]$. Figure 5c and Figure 5d show the mean average precision of MDD+nPAT on Pr $\rightarrow$ Cl when varying the hyper-parameter values. Figure 5c shows that the best result is achieved when the α is 1.0 or 2.0 when β is fixed. When α is fixed, Figure 5d shows that the best results are achieved when β is 1.0 and Then the accuracy gradually declines as β increases.

Finally, We evaluate the effect of the number of iterations of the inner maximization in our MDD+nPAT model. We choose Rw $\rightarrow$ Pr from Office-Home, C $\rightarrow$ R from DomainNet and A $\rightarrow$ W from Office-31 as our evaluation tasks. The results are illustrated in Figure 5b. It shows that on average the performance would be slightly improved along with more iterations that generate the synthetic data on all datasets. Though performance reaches the maximum value at iteration of 12, it is relatively time-consuming in the training process.

D. Visualization

For better illustration of our proposed method against the baselines, the tsne figures of the features of the source data and target data are plotted, as shown in Figure 7. We choose the source only, CDAN [37] and MDD [38] for comparison..The experiment is performed on Rw $\rightarrow$ Pr task from Office-Home dataset. The illustration suggests that compared to the baselines, our method can achieve better alignment of the source data and target data in the feature space. On one hand, our synthetic data generated by nPAT contains both semantic information from source data and target data and can reduce the domain gap accordingly during the training. Besides, the conditional centroid alignment can further make the features of the data from the same classes more compact. As a result, our proposed method can achieve better performance on the CDA problem.

To compare our adversarial example generation method with the ℓ_p-based augmentation method, the tsne figures of the features of the source data and target data are plotted, as shown in Figure 8. The experiment is performed on Rw $\rightarrow$ Pr task from Office-Home dataset. In the experiment, our nPAT is replaced with the ℓ_p-based augmentation method and other conditions such as hyper-parameters remain the same. The illustration shows that both the ℓ_p-based augmentation method and ours can achieve good alignment on most classes. However, as for the ℓ_p-based augmentation method, since the augmented data is generated within the ball of ℓ_p space, there is a chance that the augmented data violates the decision boundary and training with such data would make the model yield less discriminative features for the data from different classes. Thus, the performance would be lowered in this case. As for our method, the direction of the generation of the augmented data is pointed to the data from the same class in the target domain. The chance that generates the synthetic data that violates the decision boundary would be considerably diminished.

V. Conclusion

In this paper, we propose a pairwise adversarial training approach to address the class-imbalanced unsupervised domain adaptation (CDA) problem. Our approach generates interpolated adversarial samples across source and target domains. In order to alleviate the biased label distribution issue, we use the noisy interpolated adversarial samples to augment the training data (especially the minority classes) and meanwhile adopt the centroid alignment strategy to explicitly align source and target domains. Experimental results on three CDA benchmark datasets show that, when integrated with our PAT or nPAT method, the integrated models can yield considerable improvement on performance compared with the original models and other state-of-the-art CDA methods. Though our proposed approach can achieve promising performance on different benchmark datasets in the CDA problem. it takes multiple inner iterations to generate the synthetic data and as a result the additional computational overhead is inevitable. The more time-efficient method to generate the synthetic data can be explored in the future.

Acknowledgement

The work is in part supported by the U.S. Army Research Office Award under Grant Number W911NF-21-1-0109 and the National Science Foundation under Grant IIS-2316306.

References

[1] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar, "Regularized learning for domain adaptation under label shifts," 2019.
[2] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Thirty-second AAAI conference on Artificial Intelligence*, 2018.
[3] S. Cui, S. Wang, J. Zhuo, C. Su, Q. Huang, and Q. Tian, "Gradually vanishing bridge for adversarial domain adaptation," pp. 12 452–12 461, 2020.
[4] L. Hu, M. Kan, S. Shan, and X. Chen, "Unsupervised domain adaptation with hierarchical gradient synchronization," pp. 4042–4051, 2020.
[5] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," pp. 6028–6039, 2020.
[6] J. Na, H. Jung, H. J. Chang, and W. Hwang, "Fixbi: Bridging domain spaces for unsupervised domain adaptation," pp. 1094–1103, 2021.

[7] L. Zhou, S. Xiao, M. Ye, X. Zhu, and S. Li, "Adaptive mutual learning for unsupervised domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[8] Z. Yu, J. Li, L. Zhu, K. Lu, and H. T. Shen, "Classification certainty maximization for unsupervised domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[9] X.-Q. Liu, X.-Y. Ding, X. Luo, and X.-S. Xu, "Unsupervised domain adaptation via class aggregation for text recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[10] Y. Cao, H. Zhang, X. Lu, Y. Chen, Z. Xiao, and Y. Wang, "Adaptive refining-aggregation-separation framework for unsupervised domain adaptation semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[11] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," pp. 1406–1415, 2019.

[12] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," pp. 740–755, 2014.

[13] Y. Wu, E. Winston, D. Kaushik, and Z. Lipton, "Domain adaptation with asymmetrically-relaxed distribution alignment," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6872–6881.

[14] S. Tan, X. Peng, and K. Saenko, "Class-imbalanced domain adaptation: an empirical odyssey," in *European Conference on Computer Vision*. Springer, 2020, pp. 585–602.

[15] X. Jiang, Q. Lao, S. Matwin, and M. Havaei, "Implicit class-conditioned domain alignment for unsupervised domain adaptation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4816–4827.

[16] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," pp. 1321–1330, 2017.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[18] H. Chou, S. Chang, J. Pan, W. Wei, and D. Juan, "Remix: Rebalanced mixup," pp. 95–110, 2020.

[19] W. Shi, R. Zhu, and S. Li, "Pairwise adversarial training for unsupervised class-imbalanced domain adaptation," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1598–1606.

[20] S. H. Lim, N. B. Erichson, F. Utrera, W. Xu, and M. W. Mahoney, "Noisy feature mixup," in *International Conference on Learning Representations*, 2021.

[21] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[22] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[23] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4893–4902.

[24] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[25] Z. Zhang, M. Wang, Y. Huang, and A. Nehorai, "Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3437–3445.

[26] J. Lee and M. Raginsky, "Minimax statistical learning with wasserstein distances," *Advances in Neural Information Processing Systems*, vol. 2018, pp. 2687–2696, 2018.

[27] Y. Balaji, R. Chellappa, and S. Feizi, "Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6500–6508.

[28] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *Advances in Neural Information Processing Systems*, vol. 29, pp. 136–144, 2016.

[29] M. Long, H. Zhu, and J. Wang, "Deep transfer learning with joint adaptation networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2208–2217.

[30] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European Conference on Computer Vision*. Springer, 2016, pp. 443–450.

[31] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[32] W. Wang, H. Li, Z. Ding, F. Nie, J. Chen, X. Dong, and Z. Wang, "Rethinking maximum mean discrepancy for visual domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 264–277, 2023.

[33] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1180–1189.

[34] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[35] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[36] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[37] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *NeurIPS*, 2018, pp. 1647–1657.

[38] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," pp. 7404–7413, 2019.

[39] A. Ma, J. Li, K. Lu, L. Zhu, and H. T. Shen, "Adversarial entropy optimization for unsupervised domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6263–6274, 2022.

[40] J. Chen, X. Wu, L. Duan, and S. Gao, "Domain adversarial reinforcement learning for partial domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 539–553, 2022.

[41] Z. Yu, J. Li, L. Zhu, K. Lu, and H. T. Shen, "Uneven bi-classifier learning for domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[42] Z. Mei, P. Ye, H. Ye, B. Li, J. Guo, T. Chen, and W. Ouyang, "Automatic loss function search for adversarial unsupervised domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[43] J. Li, Z. Du, L. Zhu, Z. Ding, K. Lu, and H. T. Shen, "Divergence-agnostic unsupervised domain adaptation by adversarial attacks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8196–8211, 2021.

[44] J. Li, L. Yang, Q. Wang, and Q. Hu, "Wdan: A weighted discriminative adversarial network with dual classifiers for fine-grained open-set domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[45] Q. Lv, Y. Li, J. Dong, and Z. Guo, "Lafea: Learning latent representation beyond feature for universal domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[46] Y. Luo, Z. Wang, Z. Chen, Z. Huang, and M. Baktashmotlagh, "Source-free progressive graph learning for open-set domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[47] V. Prabhu, S. Khare, D. Karthik, and J. Hoffman, "Selective entropy optimization via committee consistency for unsupervised domain adaptation," in *International Conference in Computer Vision (ICCV)*, 2021.

[48] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, 2010.

[49] R. T. des Combes, H. Zhao, Y. Wang, and G. J. Gordon, "Domain adaptation with conditional distribution matching and generalized label shift," in *NeurIPS*, 2020.

[50] Z. C. Lipton, Y. Wang, and A. J. Smola, "Detecting and correcting for label shift with black box predictors," pp. 3128–3136, 2018.

[51] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[52] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR (Poster)*, 2015.

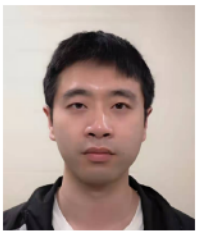
[53] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2018.

[54] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8093–8104.

[55] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," *arXiv preprint arXiv:1805.12152*, 2018.

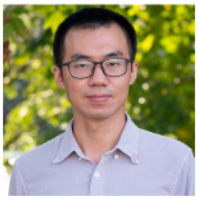
[56] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy,"

in *International Conference on Machine Learning*. PMLR, 2019, pp. 7472–7482.
[57] H. Wang, T. Chen, S. Gui, T. Hu, J. Liu, and Z. Wang, "Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7449–7461, 2020.
[58] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli, "Attacks which do not kill training make adversarial learning stronger," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 278–11 287.
[59] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[60] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.
[61] R. Shu, H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," in *International Conference on Learning Representations*, 2018.
[62] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[63] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and smote," *Information sciences*, vol. 465, pp. 1–20, 2018.
[64] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional random oversampling for imbalanced text classification," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 805–808.
[65] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine learning with oversampling and undersampling techniques: overview study and experimental results," in *2020 11th international conference on information and communication systems (ICICS)*. IEEE, 2020, pp. 243–248.
[66] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.
[67] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," pp. 5419–5428, 2018.
[68] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6023–6032.
[69] J.-H. Kim, W. Choo, and H. O. Song, "Puzzle mix: Exploiting saliency and local statistics for optimal mixup," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5275–5285.
[70] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018.
[71] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, pp. 151–175, 2010.
[72] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," pp. 1081–1090, 2019.
[73] B. Li, Y. Wang, T. Che, S. Zhang, S. Zhao, P. Xu, W. Zhou, Y. Bengio, and K. Keutzer, "Rethinking distributional matching based domain adaptation," *arXiv preprint arXiv:2006.13352*, 2020.
[74] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," pp. 213–226, 2010.
[75] W. J. Reed, "The pareto, zipf and other power laws," *Economics letters*, vol. 74, no. 1, pp. 15–19, 2001.
[76] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," pp. 5385–5394, 2017.



Weili Shi received the BSc degree in applied physics from Northeastern University, China in 2012 and MSc degree in applied physics from National University of Singapore in 2014. From 2019 to 2022 He was a student in Department of Computer Science, University of Georgia. He served as program committee member for the AAAI-23 AI for web advertising workshop. Now he is a PhD student in School of Data Science, University of Virginia. His research interests include vision transformer, domain adaptation and imbalanced learning.

Ronghang Zhu received the B.S. and M.S. degree in the College of Computer Science from Sichuan University, China, in 2014 and 2017. He is now working towards the Ph.D degree in the Department of Computer Science at University of Georgia. His research interests include domain adaptation and computer vision.



Sheng Li (S'11-M'17-SM'19) received the B.Eng. degree in computer science and engineering and the M.Eng. degree in information security from Nanjing University of Posts and Telecommunications, China, and the Ph.D. degree in computer engineering from Northeastern University, Boston, MA, in 2010, 2012 and 2017, respectively. He is a Quantitative Foundation Associate Professor of Data Science and an Associate Professor of Computer Science (by courtesy) at the University of Virginia. He was a Tenure-Track Assistant Professor at the School of Computing, University of Georgia from 2018 to 2022, and was a data scientist at Adobe Research from 2017 to 2018. He has published over 170 papers at peer-reviewed conferences and journals, and has received over 10 research awards, such as the INNS Aharon Katzir Young Investigator Award, Fred C. Davidson Early Career Scholar Award, Adobe Data Science Research Award, Cisco Faculty Research Award, SDM Best Paper Award, and IEEE FG Best Student Paper Honorable Mention Award. He serves as an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cognitive and Developmental Systems, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Computational Intelligence Magazine. He has also served as area chair for IJCAI, NeurIPS, ICML, ICLR, and SDM. His research interests include trustworthy machine learning, graph mining, computer vision, and causal inference.