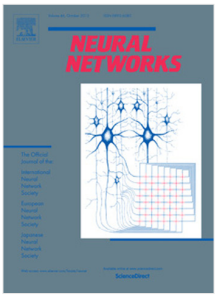
Journal Pre-proof



ProxyMix: Proxy-based Mixup training with label refinery for source-free domain adaptation

Yuhe Ding, Lijun Sheng, Jian Liang, Aihua Zheng, Ran He

PII: S0893-6080(23)00424-0
DOI: https://doi.org/10.1016/j.neunet.2023.08.005
Reference: NN 5813

To appear in: *Neural Networks*

Received date : 31 October 2022
Revised date : 3 August 2023
Accepted date : 4 August 2023

Please cite this article as: Y. Ding, L. Sheng, J. Liang et al., ProxyMix: Proxy-based Mixup training with label refinery for source-free domain adaptation. *Neural Networks* (2023), doi: https://doi.org/10.1016/j.neunet.2023.08.005.

Highlights

ProxyMix: Proxy-based Mixup Training with Label Refinery for Source-Free Domain Adaptation

Yuhe Ding, Lijun Sheng, Jian Liang, Aihua Zheng, Ran He

- We propose a simple yet effective method, ProxyMix, for source-free domain adaptation, which aims to discover a proxy source domain and utilize mixup training to implicitly bridge the gap between the target domain and the unseen source domain.
- To obtain a reliable proxy source domain, we exploit the network weights of the source model and select source-like samples from the target domain in an efficient and accurate way.
- To refine the noisy pseudo labels during alignment, we further propose a new frequency-weighted aggregation strategy, compacting the target feature clusters and avoiding bias to the majority and easy classes.

ProxyMix: Proxy-based Mixup Training with Label Refinery for Source-Free Domain Adaptation

Yuhe Ding[1], Lijun Sheng[3,4], Jian Liang[3,*], Aihua Zheng[2], Ran He[3]

Abstract

Due to privacy concerns and data transmission issues, Source-free Unsupervised Domain Adaptation (SFDA) has gained popularity. It exploits pre-trained source models, rather than raw source data for target learning, to transfer knowledge from a labeled source domain to an unlabeled target domain. Existing methods solve this problem typically with additional parameters or noisy pseudo labels, and we propose an effective method named Proxy-based Mixup training with label refinery (ProxyMix) to avoid these drawbacks. To avoid additional parameters and leverages information in the source model, ProxyMix defines classifier weights as class prototypes and creates a class-balanced proxy source domain using nearest neighbors of the prototypes. To improve the reliability of pseudo labels, we further propose the frequency-weighted aggregation strategy to generate soft pseudo labels for unlabeled target data. Our strategy utilizes target features' internal structure, increases weights of low-frequency class samples, and aligns the proxy and target domains using inter- and intra-domain mixup regularization. This mitigates the negative impact of noisy labels. Experiments on three 2D image and 3D point cloud object recognition benchmarks demonstrate that ProxyMix yields state-of-the-art performance for source-free UDA tasks.

*Code is available at https://github.com/YuheD/ProxyMix.

*Corresponding author.

Email addresses: madao3c@foxmail.com (Yuhe Ding), slj0728@mail.ustc.edu.cn (Lijun Sheng), liangjian92@gmail.com (Jian Liang), ahzheng214@foxmail.com (Aihua Zheng), rhe@nlpr.ia.ac.cn (Ran He)

[1]School of Computer Science and Technology, Anhui University.

[2]School of Artificial Intelligence, Anhui University.

[3]Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences (CASIA).

[4]University of Science and Technology of China.

Keywords: Source-free unsupervised domain adaptation, Pseudo labeling.

1. Introduction

The standard practice in the deep learning era—learning with massively labeled data—becomes expensive and laborious in many real-world scenarios. Besides, the learned models often perform poorly in generalization to new unlabeled domains due to the domain discrepancy [1]. Hence, considerable efforts are devoted to unsupervised domain adaptation (UDA) [2, 3, 4, 5], which aims to transfer knowledge from a labeled source dataset to an unlabeled target dataset. In recent years, UDA methods have been widely explored in various tasks such as image classification [4] and semantic segmentation [6]. The key problem of UDA is to alleviate the gap across different domains. Prior UDA methods mainly fall into three paradigms. The first paradigm aims to pull the statistical moments of different feature distributions closer [7, 8], and the second paradigm introduces adversarial training with additional discriminators [4, 9]. The last paradigm adopts various regularizations on the target network outputs like self-training or entropy-related objectives [10, 11]. Despite the impressive progress, it is important to note that the availability of source data remains essential for domain alignment. However, this requirement can raise data privacy concerns in today's world.

The practical demand directly motivates a novel UDA setting named *source-free domain adaptation* (SFDA) [12, 13], where only the well-trained source model instead of the well-annotated source dataset is provided to the target domain. The booming efforts in the SFDA community are either generation-based or pseudo label-based. The generation-based methods [13, 14, 15] introduce extra generative modules to recover the unseen source domain at image-level or feature-level, and then address this problem from a UDA perspective. Nevertheless, generative modules introduce additional parameters, and the recovered virtual source domain usually suffers from a mode collapse problem, which results in low-quality images or features. The pseudo label-based methods [15, 16, 17, 18] label the target samples based on the present model's prediction or feature structure. However, due to the extreme domain shift, the noises are inescapable, resulting in an inaccurate decision boundary.

To address the issues above (additional parameters and noisy labels), we propose a new and effective method called Proxy-based Mixup training with label refinery (ProxyMix), to deal with the source-free domain adaptation problem. To bridge the gap between the unseen source domain and the target domain while

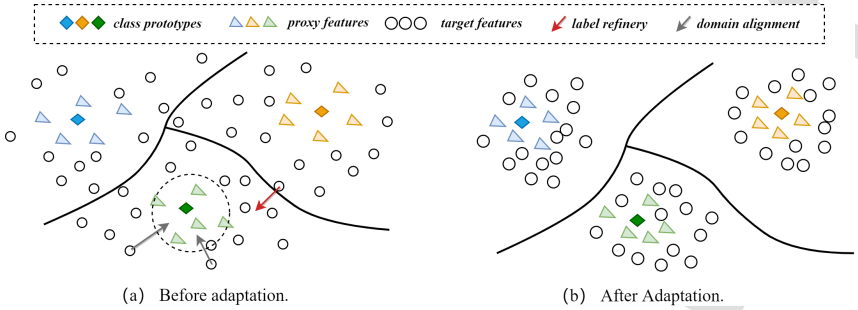


Figure 1: The motivation of ProxyMix, which aligns the unseen source domain and target domain by two aspects: 1) aligning the proxy and target domain; and 2) refining the pseudo labels.

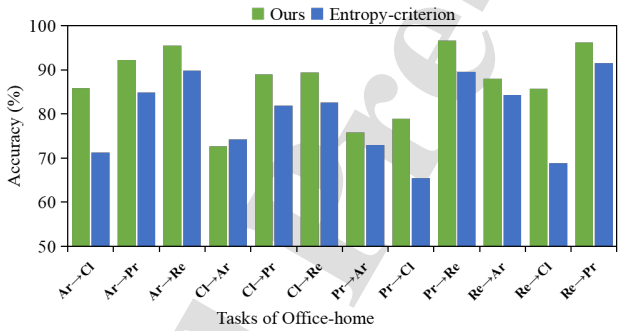


Figure 2: The accuracies per task of proxy source domain on **Office-home**.

avoiding introducing extra parameters, we first select part of source-similar samples from the target domain rather than synthesize virtual images to construct a proxy source domain. Specifically, we define the weights of the source classifier as the class prototypes [19], then select the nearest neighbors for each class prototype in angle space to construct the proxy source domain. Priors methods with proxy source domain primarily employ entropy-criterion [16, 20], which select samples with lower entropy for each class from pseudo-labeled target data. In practice, as shown in Fig. 2, we observe that the mean accuracy of our angle-induced proxy source domain is clearly higher than the entropy criterion. Another significant benefit is that our pseudo labels are determined by the corresponding prototype, rather than the predictions from the source model, allowing us to create a class-balanced proxy source domain.

To improve the reliability of pseudo labels, we propose a frequency-weighted

aggregation pseudo-labeling strategy (FA) as pseudo label refinery. FA includes three operations applied to the predictions: sharpening, re-weighting, and aggregation. Specifically, to avoid the ambiguous, we first sharpen the predictions of the classifier. At the same time, we take the frequency of each class into account and re-weight the probability of each class, to improve the contribution of low-frequency classes and avoid bias to the majority and easy classes in the target domain during gradient updating. Then we introduce a non-parametric neighborhood aggregation strategy to pull the unlabeled target features close to their semantic neighbors, aiming to reduce the impact of outlier noisy labels and compact the semantic clusters.

With the proxy source domain, we tackle the challenging SFDA problem using a semi-supervised style with the aid of refined pseudo labels. To align the proxy and target domain, while alleviating the negative consequence of noisy labels, two mixup regularizations [21, 22, 23, 24], *i.e.*, inter-domain and intra-domain mixup, are incorporated into our framework, enforcing the model to maintain consistency, thus improving the robustness against noisy labels. As illustrated in Fig. 1, the FA strategy refines the pseudo labels and compacts the feature clusters while the mixup training aligns the two domains, obtaining clear decision boundaries.

To summarize, the main contributions of this work are listed below in three-fold:

- We propose a simple yet effective method, ProxyMix, for source-free domain adaptation, which aims to discover a proxy source domain and utilize mixup training to implicitly bridge the gap between the target domain and the unseen source domain.
- To obtain a reliable proxy source domain, we exploit the network weights of the source model and select source-like samples from the target domain in an efficient and accurate way.
- To refine the noisy pseudo labels during alignment, we further propose a new frequency-weighted aggregation strategy, compacting the target feature clusters and avoiding bias to the majority and easy classes.

We conduct ablation studies to verify the contribution and effectiveness of both proxy source domain construction and pseudo label refinery. Extensive results on four datasets further validate that ProxyMix yields comparable or superior performance to the state-of-the-art SFDA methods.

2. Related Work

2.1. Unsupervised Domain Adaptation (UDA)

UDA aims to transfer knowledge from a label-rich source domain to an unlabeled target domain. UDA problems can be classified into four cases according to the relationship between the source and target domain, *i.e.*, closed-set [25], partial-set [26], open-set [27], and universal [28]. As a typical example of transfer learning, UDA provides methods to bridge domain gaps for various applications such as object recognition [29, 4, 2, 30, 3, 31, 32] and semantic segmentation [6, 10]. The most prevailing paradigm for UDA is to extract domain-invariant features to align different domains while preserving the category information from the labeled source domain. Roughly speaking, existing feature-level domain alignment could be divided into two different categories. The first line [4, 9, 5] aligns representations by fooling a domain discriminator through adversarial training, while the second line [29, 33] directly minimizes different discrepancy metrics (e.g., statistical moments) to match the feature distributions. Besides, another line [34] focuses on the image space alignment and converts the target image into a source-style image (and visa versa). By contrast, output-level regularization methods [11, 35] achieve implicit domain alignment by forcing the target outputs to be diverse one-hot encodings. [36] proposes an auxiliary classifier for target data to get the high-quality pseudo labels and [37] introduces cycle self-training by utilizing target pseudo labels to train another head and enforce them to perform well on the source domain. [38, 39] are the two most closely related works that introduce mixup training into adversarial UDA. However, our method does not require access to source data and develops a new pseudo label refinery strategy instead of focusing on the mixing manner.

2.2. Source-free Domain Adaptation (SFDA)

SFDA can be seen as a special case of Test-Time Adaptation (TTA) [40], which involves adapting a pre-trained model from the source domain to unlabeled data in the target domain before making predictions. Different from the other types of TTA methods [41, 42, 43, 44], SFDA involves utilizing all test data (target data) during adaptation and performing multi-epoch adaptation before generating final predictions. Before the deep learning era, there are a number of transfer learning works [45, 46, 47, 48, 49] without source data that have been empirically successful. The last two years have witnessed an increasing number of SFDA approaches [15, 16, 17, 18], most of which are generation-based [13, 14, 15, 50, 51] or self-training [12, 52, 53, 54, 55, 56, 57, 58] based methods. Generation-based

methods [14, 15, 13, 59, 20, 51] generate virtual high-level features of the source domain to bridge the unseen source and target distribution. Self-training-based methods seek to refine the source model by using self-supervised techniques, with the pseudo label technique [12, 52] being the most extensively employed. However, generating source samples usually introduces additional modules such as generators or discriminators, while pseudo-labeling might lead to wrong labels due to domain shift, both of which cause negative effects on the adaptation procedure. Another practice [59, 20, 16] is selecting part of the target data as a pseudo source domain, to compensate for the unseen source domain. A typical method is entropy-criterion [16], which constructs the pseudo source domain by estimating a split ratio using the target dataset's mean and maximum entropy, and then uses the split ratio to choose samples with lower entropy for all pseudo-labeled target domains within each class. The entropy criterion provides a proxy source domain with a huge number of samples. However, the existence of hard classes and domain shift, causes the entropy criterion to suffer from a severe class imbalance problem. Despite the fact that [20] attempts to tackle this problem by simply choosing the same number for each class, there is no data in some hard classes, so the class-imbalance problem is unavoidable. Unlike the previous works, our method builds the proxy source domain directly from the target domain using the source classifier weights, which is flexible and works well for SFDA. Besides, our mixup training strategy is also different from theirs, which transfers the label information from the proxy source to the unlabeled target domain.

2.3. Semi-Supervised Learning (SSL)

SSL aims to combine supervised learning and unsupervised learning, leveraging the vast amount of unlabeled data with limited labeled data to improve the performance of the classifier and to deal with the scenarios where labeled data is scarce [60]. As opposed to the domain adaptation problem, SSL deals with samples from two identical domains. SSL has flourished in recent years [61, 62, 63], temporal ensemble [64] introduces self-ensembling, forming a consensus prediction of the unknown labels using the outputs of the network-in-training on different epochs; MixMatch [22] proposes a holistic approach for data-augmented unlabeled examples and mixing labeled and unlabeled data using mixup; ReMixMatch [23] aligns the distribution of labeled and unlabeled data. FixMatch [65] demonstrates the strong performance of consistency regularizations and pseudo labels; SoftMatch [66] derives a truncated Gaussian function to weight samples based on their confidence; AdaMatch [24] proposes a unified approach to solve the unsupervised domain adaptation, semi-supervised learning, and semi-supervised

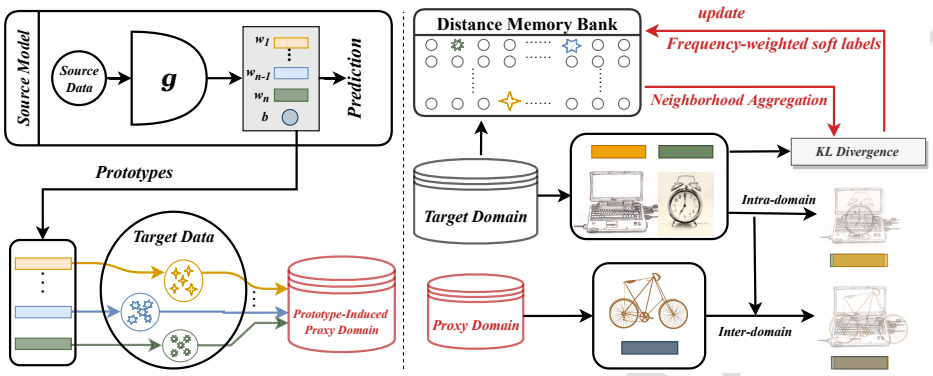


Figure 3: Overview of ProxyMix on solving source-free domain adaptation. We treat the weights of the classifier as class prototypes to choose a series of confident samples to construct a class-balanced proxy source domain. Then the proxy source samples participate in two types of mixup training based on the proposed frequency-weighted soft label.

domain adaptation problems. Existing methods demonstrate the usefulness of mixup training in aligning distributions, and the growing popularity of SSL motivates us to convert the SFDA problem to an SSL challenge. Such methods use true labels, which are not available in our task, and these labels provide strong and diverse supervision. Our data is pseudo-labeled, with little diversity and a lot of noise, so these semi-supervised learning approaches cannot be directly applied to our problem.

3. Methodology

This paper mainly follows the problem definition of SHOT [12] and focuses on a K-way visual classification task. We aim to learn a target model $f_t : \mathcal{X}_t \to \mathcal{Y}_t$, and predict the label $y_t^i \in \mathcal{Y}_t$ for an input target image $x_t^i \in \mathcal{X}_t$ with only target data $\mathcal{X}_t$ and the well-trained source model $f_s : \mathcal{X}_s \to \mathcal{Y}_s$. The model consists of two modules: the feature extractor $g : \mathcal{X} \to \mathbb{R}^d$ and the classifier $h : \mathbb{R}^d \to \mathbb{R}^K$.

Following the standard paradigm of SFDA [12], as a preliminary, we train the source model f_s with the label smoothing [67] technique:

$$\mathcal{L}_{src}^{ls}\left(f_s; \mathcal{X}_s, \mathcal{Y}_s\right) = -\mathbb{E}_{(x_s,y_s)\in\mathcal{X}_s\times\mathcal{Y}_s} \sum_{k=1}^{K} l_k^s \log \delta_k\left(f_s\left(x_s\right)\right), \tag{1}$$

where $l_k^s = (1-\alpha)q_k^s + \alpha/K$, q^s is the one-hot encoding of y_s, $\alpha = 0.1$ is the smoothing parameter, and $\delta_k(a) = \frac{\exp(a_k)}{\sum_i \exp(a_i)}$ is the soft-max output of the K-dimensional vector $a \in \mathbb{R}^K$.

During adaptation, we directly initialize the target model with the well-trained source model $f_t = f_s$, then freeze the classifier and fine-tune the feature extractor to ensure the target features are implicitly aligned with unseen source features via a same hypothesis. It is worth noting that we do not adopt the special design of normalization techniques of SHOT [12] for simplicity and commonality.

3.1. Proxy Source Domain Construction by Prototypes

Recently, semi-supervised learning approaches [22, 23] have also shown impressive achievements on the UDA problem, and Rukhovich et al. [68] even wins the VisDA competition by directly exploiting MixMatch [22] in 2019. Inspired by them, we construct the proxy source domain by pseudo-labeling portions of confident samples (source-similar samples) and try to solve the SFDA task in a semi-supervised style. Since the source data $\mathcal{X}_s$ is unavailable, we expect to mine the source information from the model f_s. Previous works [69, 70] leverage the weights of the classifier as class prototypes in other fields, and obtain positive results. Another classical practice [19] exposes that the classifier weight vector of a well-trained last-layer classifier converges to a high-dimension geometry structure, which maximally separates the pair-wise angles of all classes in the classifier. Therefore, inspired by these works, it is natural to select the nearest neighbors of classifiers' weights in angle space to construct the proxy source domain. Concretely, we first define the weights $\{w_1, w_2, ...w_K\}_{k=1}^K$ of the classifier h_s as the class prototypes, where K is the number of categories. We use the class prototype w_k as the cluster centroid to search and pseudo-label N nearest samples in the unlabeled target domain $\mathcal{X}_t$ for the purpose of forming proxy source domain $\mathcal{X}_{ps}$:

$$\begin{aligned} \{\mathcal{X}_{ps}, \mathcal{Y}_{ps}\} &= \{\mathcal{X}_{ps}^1, 1\} \cup \cdots \cup \{\mathcal{X}_{ps}^K, K\}, \\ \text{where } \mathcal{X}_{ps}^k &= \{x_{ps}; x_{ps} \in \underset{x_t}{min_N} \left(\langle g_s(x_t), w_k \rangle\right)\}, \end{aligned} \tag{2}$$

and $\underset{x_t}{min_N}(\cdot)_{k=1}^K$ denotes choosing N samples x_t with minimum distance for each class, N is a hyper-parameter, deciding how many samples we select in each class. To prevent the negative consequences caused by class imbalance, we select the same number of samples for each class. $\langle a, b \rangle$ measures the distance between a and b in angle space, we use the cosine similarity by default. For these proxy

source data, we directly calculate the cross entropy loss with labeling smoothing in the following,

$$\mathcal{L}_{ps}\left(f_t; \mathcal{X}_{ps}, \mathcal{Y}_{ps}\right) = -\mathbb{E}_{(x_{ps}, y_{ps}) \in \mathcal{X}_{ps} \times \mathcal{Y}_{ps}} \sum_{k=1}^{K} l_k^{ps} \log \delta_k \left(f_t\left(x_{ps}\right)\right), \tag{3}$$

where $l_k^{ps} = (1-\alpha) q_k^{ps} + \alpha / K$ is the smoothed label, q^{ps} denotes the one-hot encoding of y_{ps}.

3.2. Pseudo-labeling by Frequency-weighted Aggregation (FA)

Pseudo-labeling is a heuristic approach to semi-supervised learning, which progressively treats the predictions on unlabeled data as true labels, and often employs cross-entropy loss during training. However, in an unsupervised learning setting, the class distribution is unknown, and the model is biased towards easy classes. To mitigate the imbalance and sensitivity of pseudo labels, inspired by several classical works [36, 71], we propose a new pseudo label refinery strategy to get reliable soft pseudo labels in the presence of domain shift. In specific, we adjust the class distribution of the prediction to alleviate the class imbalance, and then we use the center of semantic neighbors as the pseudo label, rather than depending on a single prediction. This compacts the cluster by pulling the unlabeled target features closer to their semantic neighbors, resulting in a clear classification boundary. Note that hard labels reinforce the confidence of the current model, while losing some information. Hence we use the soft predictions rather than the one-hot vectors as the pseudo labels, which are able to provide more distribution information and decrease the negative effect of corrupted one-hot labels.

Neighborhood Aggregation. To leverage the local data structure, we employ the neighborhood aggregation strategy [36], which is based on the idea of message passing via neighbors, to adjust the predictions of the input target data. Concretely, we construct a large memory bank to store both the features and the predictions of target data. During pseudo-labeling, we retrieve m nearest neighbors from the memory bank for each sample in the current mini-batch according to their features $g_t(x_t^i)$, and calculate the soft label $\hat{q}_i$ of data point x_t^i by aggregating these predictions of feature-level neighbors:

$$\hat{q}_i = \frac{1}{m} \sum_{j \neq i, j \in \mathcal{N}_i} \check{p}_j, \tag{4}$$

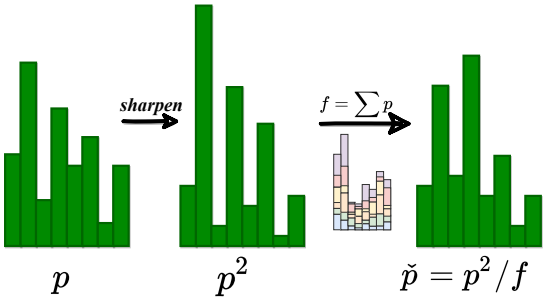


Figure 4: Illustration of the frequency-weighted strategy as label refinery. We first sharpen the predictions to the second power and then normalize the predictions by the frequency per class.

where $\mathcal{N}_i$ is the neighbor index set of the data x_t^i, $\check{p}_j$ are the frequency-weighted predictions of neighbors stored in the bank, then we explain how these predictions are obtained.

Frequency-weighted prediction. As illustrated in Fig. 4, to avoid ambiguity, we first sharpen the calculated output predictions p_i. Besides, the network will be empirically skewed towards these majority classes due to the class imbalance. Then, we further multiply the predictions by a weight based on the frequency of the class. In specific, given the soft-max output predictions $p_i = \delta(f_t(x_t^i))$, the frequency-weighted predictions can be obtained through

$$\{\check{p}_{ij}\}_{j=1}^{K} = \left\{ \frac{p_{ij}^2 / f_j}{\sum_{j'} (p_{ij'}^2 / f_{j'})} \right\}_{j=1}^{K}, \tag{5}$$

where $f_j = \sum_i p_{ij}$ are soft cluster frequencies calculated by the current batch of samples, K represents the number of the classes. Through the operation above, we expect to achieve class-balance in the predictions. At each iteration, we update the features and predictions associated with the data in the corresponding location in the memory bank.

3.3. Domain Alignment by Mixup Training

Two mixup training procedures are incorporated into our method. In essence, mixup trains a neural network on convex combinations of pairs of examples and their labels to regularize the network to support linear behavior in-between training samples. Pioneers have proved the effectiveness of mixup training on UDA

and SSL tasks [21, 22, 23, 68]. Such a simple regularization can improve the generalization and robustness to some noisy labels, so it is suitable for pseudo label-based unsupervised learning tasks. Inspired by these methods, with the prototype-induced pseudo source domain $\{\mathcal{X}_{ps}, \mathcal{Y}_{ps}\}$ and target domain $\mathcal{X}_t$, we introduce two different regularizations via mixup training.

Inter-domain Mixup. To align the proxy source domain and the target domain, we employ inter-domain mixup regularization. [22] mixes the labeled data with both unlabeled data and labeled data itself. However, the "labeled" data in our case is not completely trustworthy. As a result, we do not add any mixup training between the proxy source samples, but only between the pseudo source domain and the target domain only, constructing in virtual training samples below:

$$\tilde{x}_r = \rho x_{ps} + (1-\rho)x_t, \quad \tilde{q}_r = \rho q_{ps} + (1-\rho)\hat{q},$$

where q_{ps} denotes the one-hot encoding of y_{ps}, and $\hat{q}$ is the soft label of x_t calculated by Eq. (4), ρ is the mixup coefficient sampled from a random Beta distribution, which generates continuous random numbers between 0 and 1.

Then we adopt the KL divergence to calculate the soft label classification loss:

$$\mathcal{L}_{tgt}^{inter} = \mathrm{KL}(\tilde{q}_r \| \delta(f_t(\tilde{x}_r))). \tag{6}$$

Algorithm 1 Algorithm of the proposed ProxyMix.

Input: Target dataset $\mathcal{X}_t$; well-trained source model $f(x) = h(g(x))$, where $g : \mathcal{X} \to \mathbb{R}^d$ is the feature extractor and $h : \mathbb{R}^d \to \mathbb{R}^K$ is the classifier;

1: Build the proxy source domain $\{\mathcal{X}_{ps}, \mathcal{Y}_{ps}\}$ by Eq. (2);
2: Initialize the feature memory bank B_f and prediction memory bank B_l;
3: **repeat**
4: Randomly sample a batch of target data x_t from $\mathcal{X}_t$ and proxy source data x_{ps} from $\mathcal{X}_{ps}$;
5: Obtain the soft label $\hat{q}$ of x_t by Eq. (4);
6: Update g by Eq. (8);
7: Update the corresponding features and predictions of x_t in feature bank B_f and prediction bank B_l;
8: **until** Iterations are exhausted.

Output: New model $f(x) = h(g(x))$.

Intra-domain Mixup. To mine the inner structure of the target domain, we also adopt the mixup regularization between different target data. As is typical in

Table 1: Classification accuracies (%) of state-of-the-art methods on **Office-home** [72] (ResNet-50). SF denotes source-free. We use **Bold** to highlight the best and underline to highlight the second best among source-free methods.

SF	Method	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Avg.
×	MCD [73]	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
×	CDAN [5]	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
×	SAFN [74]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
×	SymNets [75]	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
×	MDD [76]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
×	TADA [77]	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
×	BNM [11]	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
×	BDG [78]	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
×	SRDC [79]	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	76.3	57.1	85.0	71.3
×	RSDA-MSTN [80]	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
×	ATDOC [36]	60.2	77.8	82.2	68.5	78.6	77.9	68.4	58.4	83.1	74.8	61.5	87.2	73.2
	No Adapt.	46.1	67.0	74.3	52.0	62.7	64.3	53.8	42.1	73.7	67.0	47.7	78.2	60.7
✓	SSFT-SSD [59]	51.7	76.0	79.9	66.8	75.8	77.2	63.9	52.1	80.6	73.5	57.1	83.0	69.8
✓	VDM-DA [14]	**59.3**	75.3	78.3	67.6	76.0	75.9	**68.8**	57.7	79.6	74.0	61.1	83.6	71.4
✓	CPGA [15]	**59.3**	78.1	79.8	65.4	75.5	76.4	65.7	**58.0**	81.0	72.0	**64.4**	83.3	71.6
✓	SHOT [12]	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
✓	PS [20]	57.8	77.3	81.2	68.4	76.9	78.1	67.8	57.3	82.1	**75.2**	59.1	83.4	72.1
✓	NRC [52]	57.7	80.3	82.0	68.1	79.8	78.6	65.3	56.4	83.0	71.0	58.6	**85.6**	72.2
✓	A^{2}Net [17]	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.8
✓	SCLM [53]	58.2	80.3	81.5	69.3	79.0	**80.7**	**69.0**	56.8	82.7	74.7	60.6	85.0	**73.1**
✓	U-SFAN+ [50]	57.8	77.8	81.6	67.9	77.3	79.2	67.2	54.7	81.2	73.3	60.3	83.9	71.9
✓	AaD [55]	**59.3**	79.3	82.1	68.9	79.8	79.5	67.2	57.4	83.1	72.1	58.5	85.4	72.7
✓	C&C [54]	59.0	79.5	82.0	67.6	79.2	79.5	66.7	56.5	81.3	74.2	58.3	84.7	72.4
✓	CoWA-JMDS [58]	56.9	78.4	81.0	69.1	**80.0**	79.9	67.7	57.2	82.4	72.8	60.5	84.5	72.5
✓	VMP [57]	57.9	77.6	**82.5**	68.6	79.4	80.6	68.4	55.6	83.1	**75.2**	59.6	84.7	72.8
✓	DIPE [56]	56.5	79.2	80.7	**70.1**	79.8	78.8	67.9	55.1	**83.5**	74.1	59.3	84.8	72.5
✓	ProxyMix	**59.3**	**81.0**	81.6	65.8	79.7	78.1	67.0	57.5	82.7	73.1	61.7	**85.6**	72.8

Table 2: Classification accuracies (%) on **Office-31** [81] (ResNet-50). [$*$: mean values except D↔W.]

SF	Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg.	Avg.*
	No Adapt.	77.3	73.8	59.9	96.5	60.7	98.4	77.8	67.9
×	MCD [73]	92.2	88.6	69.5	98.5	69.7	100.0	86.5	80.0
×	CDAN [5]	92.9	94.1	71.0	98.6	69.3	100.0	87.7	81.8
×	MDD [76]	90.4	90.4	75.0	98.7	73.7	99.9	88.0	82.4
×	BNM [11]	90.3	91.5	70.9	98.5	71.6	100.0	87.1	81.1
×	DMRL [39]	93.4	90.8	73.0	99.0	71.2	100.0	87.9	82.1
×	BDG [78]	93.6	93.6	73.2	99.0	72.0	100.0	88.5	83.1
×	MCC [35]	95.6	95.4	72.6	98.6	73.9	100.0	89.4	84.4
×	SRDC [79]	95.8	95.7	76.7	99.2	77.1	100.0	90.8	86.3
×	RWOT [82]	94.5	95.1	77.5	99.5	77.9	100.0	90.8	86.3
×	RSDA-MSTN [80]	95.8	96.1	77.4	99.3	78.9	100.0	91.1	87.1
×	ATDOC [36]	95.4	94.6	77.5	98.1	77.0	99.7	90.4	86.1
✓	SHOT [12]	94.0	90.1	74.7	98.4	74.3	99.9	88.6	83.3
✓	SSFT-SSD [59]	95.2	95.0	72.7	98.7	73.5	**100.0**	89.2	84.1
✓	NRC [52]	96.0	90.8	75.3	99.0	75.0	**100.0**	89.4	84.3
✓	HCL [18]	94.7	92.5	75.9	98.2	**77.7**	**100.0**	89.8	85.2
✓	CPGA [15]	94.4	94.1	76.0	98.4	76.6	99.8	89.9	85.3
✓	SCLM [53]	95.8	90.0	75.5	98.9	76.0	**100.0**	89.4	84.3
✓	AaD [55]	96.4	92.1	75.0	**99.1**	76.5	**100.0**	89.9	85.0
✓	C&C [54]	95.2	93.8	74.7	**99.1**	76.3	99.8	89.9	85.0
✓	SFDA-DE [51]	96.0	94.2	**76.6**	98.5	75.5	99.8	**90.1**	85.6
✓	DIPE [56]	**96.6**	93.1	75.5	98.4	77.2	99.6	**90.1**	**85.6**
✓	ProxyMix	95.4	**96.7**	75.1	98.5	75.4	99.8	**90.1**	**85.6**

Table 3: Classification accuracies (%) on the large-scale synthesized-to-real dataset **VisDA** [83] (ResNet-101).

SF	Method	plane	bicycle	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
×	ADR [84]	94.2	48.5	84.0	72.9	90.1	74.2	92.6	72.5	80.8	61.8	82.2	28.8	73.5
×	CDAN [5]	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
×	CDAN+BSP [85]	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
×	SAFN [74]	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
×	SWD [86]	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
×	MDD [76]	-	-	-	-	-	-	-	-	-	-	-	-	74.6
×	DMRL [39]	-	-	-	-	-	-	-	-	-	-	-	-	75.5
×	MCC [35]	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
×	STAR [87]	95.0	84.0	84.6	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
×	RWOT [82]	95.1	80.3	83.7	90.0	92.4	68.0	92.5	82.2	87.9	78.4	90.4	68.2	84.0
×	ATDOC [36]	93.0	77.4	83.4	62.3	91.5	88.4	91.8	77.1	90.9	86.4	85.8	48.2	81.4
	No Adapt.	63.2	10.4	47.6	73.0	46.9	4.5	66.4	15.6	62.1	17.7	88.5	7.2	41.9
✓	SSFT-SSD [59]	95.4	86.5	79.3	51.5	92.9	94.5	82.1	79.7	90.0	87.1	87.8	57.9	82.1
✓	SHOT [12]	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
✓	HCL [18]	93.3	85.4	80.7	68.5	91.0	88.1	86.0	78.6	86.6	88.8	80.0	**74.7**	83.5
✓	PS [20]	95.3	86.2	82.3	61.6	93.3	95.7	86.7	80.4	91.6	90.9	86.0	59.5	84.1
✓	A^{2}Net [17]	94.0	87.8	85.6	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
✓	VDM-DA [14]	96.9	89.1	79.1	66.5	95.7	96.8	85.4	83.3	96	86.6	89.5	56.3	85.1
✓	NRC [52]	96.8	**91.3**	82.4	62.4	96.2	95.9	86.1	80.6	94.8	**94.1**	90.4	59.7	85.9
✓	CPGA [15]	95.6	89.0	75.4	64.9	91.7	97.5	89.7	83.8	93.9	93.4	87.7	69.0	86.0
✓	SCLM [53]	97.1	90.7	85.6	62.0	**97.3**	94.6	81.8	84.3	93.6	92.8	88.0	55.9	85.3
✓	AaD [55]	**97.4**	90.5	80.8	76.2	**97.3**	96.1	89.8	82.9	**95.5**	93.0	**92.0**	64.7	**88.0**
✓	SFDA-DE [51]	95.3	91.2	77.5	72.1	95.7	**97.8**	85.5	86.1	**95.5**	93.0	86.3	61.6	86.5
✓	CoWA-JMDS [58]	96.2	89.7	83.9	73.8	96.4	97.4	89.3	**86.8**	94.6	92.1	88.7	53.8	86.9
✓	DIPE [56]	95.2	87.6	78.8	55.9	93.9	95.0	84.1	81.7	92.1	88.9	85.4	58.0	83.1
✓	ProxyMix	95.4	81.7	**87.2**	**79.9**	95.6	96.8	**92.1**	85.1	93.4	90.3	89.1	42.2	85.7

many SSL methods, we use data augmentation on target data. In specific, for each mini-batch of target data x_t, we concatenate it with its augmented version $\hat{x}_t$ to construct a vector notated as $x_a = \mathrm{cat}(x_t, \hat{x}_t)$. Then we mixup x_a and its shuffled version x_a^s to construct the virtual training samples below:

$$\tilde{x}_a = \rho x_a + (1-\rho)x_a^s, \quad \tilde{q}_a = \rho \hat{q}_a + (1-\rho)\hat{q}_a^s,$$

where x_a^s is the shuffled version of x_a, $\hat{q}_a$ and $\hat{q}_a^s$ are the soft label of x_a and x_a^s calculated by Eq. (4), respectively. Then we formulate the intra-domain mixup regression loss as:

$$\mathcal{L}_{tgt}^{intra} = \|f_t(\tilde{x}_a) - \tilde{q}_a\|_2^2. \tag{7}$$

Note here we use square L_2 loss. Unlike the cross entropy loss used in Eq. (6), it is bounded and more robust due to the insensitivity to corrupted labels.

3.4. Overall Objective

Combining the proxy source classification loss and two types of mixup loss, our overall objective is formulated as:

Table 4: Classification accuracies (%) on the 3D point cloud dataset **PointDA-10 [88]** (PointNet [89]). The results except ours are from NRC [52] and PointDAN [89].

SF	Method	M → S	M→ S*	S → M	S → S*	S* → M	S* → S	Avg.
×	MMD [90]	57.5	27.9	40.7	26.7	47.3	54.8	42.5
×	DANN [4]	58.7	29.4	42.3	30.5	48.1	56.7	44.2
×	ADDA [9]	61.0	30.5	40.4	29.3	48.9	51.1	43.5
×	MCD [73]	62.0	31.0	41.4	31.3	46.8	59.3	45.3
×	PointDAN [89]	64.2	33.0	47.6	33.9	49.1	64.1	48.7
	No Adapt.	21.5	21.7	18.5	29.5	18.8	25.8	22.6
✓	VDM-DA [14]	58.4	**30.9**	**61.0**	**40.8**	45.3	61.8	49.7
✓	NRC [52]	64.8	25.8	59.8	26.9	70.1	**68.1**	52.6
✓	ProxyMix	**65.2**	22.4	60.8	30.8	**81.2**	64.2	**54.1**

$$\mathcal{L}_{total} = \mathcal{L}_{ps} + \lambda \mathcal{L}_{tgt}^{inter} + \eta \mathcal{L}_{tgt}^{intra} \tag{8}$$

where λ and η are trade-off parameters to balance losses. Our method is end-to-end during the training phase, using the proxy source classification loss to help the model implicitly align the unseen source and target domains. Two types of mixup loss further help us eliminate the negative effects of outlier noise labels to improve the robustness. Empirically, we set the weights of these losses to 1. In reality, these loss functions are not sensitive, and we will verify this in the sensitivity analysis in the experimental section. The overall pipeline of ProxyMix is illustrated in Algorithm 1.

4. Experiments

Datasets. We conduct the experiments on four popular benchmark datasets: (1) **Office-31** [81] is a standard domain adaptation dataset consisting of three distinct domains, *i.e.,* Amazon (A), DSLR (D) and Webcam (W), and 31 categories in the shared label space. The specific numbers of images for each domain are 2,817 (A), 498 (D), and 795 (W), therefore the dataset suffers from severe data imbalance. (2) **Office-home** [72] is a medium-sized domain adaptation dataset with 15,500 images collected from four domains Art (Ar), Clipart (Cl), Product (Pr), and Real-World (Re). There are 65 categories per domain, which is much more than **Office-31**. (3) **VisDA** [83] is a large-scale challenging dataset which consists of a 12-class synthesize-to-real object recognition task. The source domain involves 152k synthetic images which are produced by 3D rendering model under

various conditions. The target domain contains 55k images collected from the real-world scene. (4) **PointDA-10** [89] is a common-used 3D cloud-point dataset extracted from three popular 3D object/scene datasets, *i.e,* modelnet (M), shapenet (S), and scannet (S*) for cross-domain 3D object recognition. Each domain contains its own training and testing sets. We train our models by source and target domain's training set, and show the test results on the target domain's test set.

Baselines. We compare ProxyMix with the state-of-the-art source-free domain adaptation methods: SHOT [12], CPGA [15], A^{2}Net [17], HCL [18], NRC [52], SSFT-SSD [59], PS [20], SCLM [53], AaD [55], SFDA-DE [51], CoWA-JMDS [58], DIPE [56], C&C [54], U-SFAN+ [50], VMP [57]. Moreover, to illustrate the effectiveness of ProxyMix, we further compare our method with the state-of-the-art UDA methods: SymNets [75], TADA [77], BNM [11], BDG [78], SRDC [79], RSDA-MSTN [80], ADR [84], CDAN [5], CDAN+BSP [85], SAFN [74], SWD [86], MDD [76], DMRL [39], MCC [35], STAR [87], RWOT [82], ATDOC [36], MMD [90], DANN [4], ADDA [9], MCD [73], PointDAN [89]. We use **bold** to highlight the best results and underline to highlight the second best results among *source-free methods*.

Implementation Details. We implement our method based on PyTorch. For network architecture, we adopt ResNet [91], pretrained on the ImageNet as the backbone, and replace the original fully connected layer with a bottleneck layer followed by a task-specific linear layer. Specifically, we use ResNet-50 on **Office-home** and **Office-31**, ResNet-101 on **VisDA**. In the source model training stage, we exploit SGD optimizer with learning rate $1e^{-3}$ for the backbone and $1e^{-2}$ for the bottleneck and classifier. In the target adaptation stage, we use SGD optimizer with learning rate $1e^{-3}$ for the backbone and freeze the fully connected classification layer. The numbers of epochs are set to 30, 50, 5 in the training stage and 50, 50, 1 in the adaptation stage for **Office-31**, **Office-home** and **VisDA**, respectively. Specially, for **PointDA-10**, we follow the open source code of NRC [52], use PointNet [88] as our backbone network, learning rate $1e^{-6}$ and Adam optimizer with 100 epochs each stage. For the hyper-parameters, considering the confidence of pseudo labels, we set $\lambda = 1$, $\eta = 100$, and we alter λ and η linearly by multiplying a ratio that varies linearly from 0 to 1 based on the number of the current iteration. Besides, we set $m = 5$, beta distribution parameter $\beta = 0.75$ in mixup and $N = 5, 10, 10, 50$ for **Office-31**, **Office-home**, **PointDA-10** and **VisDA**. The size of the proxy source domain N is determined empirically, while other hyperparameters are set according to prior works [36, 22, 21]. *All results are the averages of three random runs with seed* $\in \{0, 1, 2\}$.

4.1. Comparison Results

2D image datasets. We first compare our method with the state-of-the-art methods on 2D image datasets **Office-home**, **Office-31**, and **VisDA** in Table 1, 2, and 3, respectively. Note that the results of other methods are from the original papers, except ours. It can be observed that we have achieved competitive results across all three datasets. On **Office-home**, our approach achieved the second highest average accuracy, with only a marginal difference of 0.3 percentage points compared to the top-performing SCLM [53], while outperforming SCLM on the other two datasets. This demonstrates the multi-class classification capability of ProxyMix on medium-scale datasets. On **Office-31**, ProxyMix and DIPE [56] achieve the highest average accuracy. For better discriminability, we also provide the average accuracy without the two tasks D→W and W→D, where ProxyMix and DIPE still perform the best. However, our approach outperforms DIPE on the **Office-home** and **VisDA**. This validates the capability of ProxyMix in handling small-scale and few-class datasets. On **VisDA**, we achieve the highest accuracy on three classes and a competitive average accuracy compared to most state-of-the-art methods. The reason why ProxyMix does not perform well on **VisDA** compared to the other two datasets is due to the relatively small size of the proxy source domain compared to the entire dataset. This causes the network to inevitably bias towards the proxy source domain during training. In summary, our method ProxyMix achieves competitive accuracy across three benchmarks when compared with others, which demonstrates the effectiveness in dealing with the standard 2D image domain adaptation benchmarks. We achieve similar results compared with the state-of-the-art SFDA methods SCLM [53] (Neural Network-22) and DIPE [56] (CVPR-22), and UDA method ATDOC [36] (CVPR-21). The presented results clearly demonstrate the efficacy of the proposed method in dealing with domain-imbalanced, multi-class, and large-scale challenges.

3D point cloud dataset. To explore the generalization performance of ProxyMix on 3D data, we also report the results for the **PointDA-10** dataset in Table 4. Without any extra modules, our method achieves the highest average accuracy on the benchmark, even compared with UDA methods and the 3D cloud point domain adaptation method PointDAN [89].

4.2. Empirical Analysis

To explore the effectiveness of the proposed pseudo-labeling strategy, the aggregation strategy, and the construction method of the proxy source domain, we conduct a series of ablation analyses on the three common-used 2D image classification datasets **Office-31**, **Office-home** and **VisDA**. Then we explore the influence

Table 5: Analysis of different soft pseudo labels.

Choices of soft label	**Office-31**	**Office-home**	**VisDA**
MixMatch [22]	88.4	72.4	83.0
ReMixMatch [23]	88.1	71.3	80.2
ATDOC [36]	88.5	72.2	84.7
Ours	**90.1**	**72.8**	**85.7**

Table 6: Analysis of aggregation strategy.

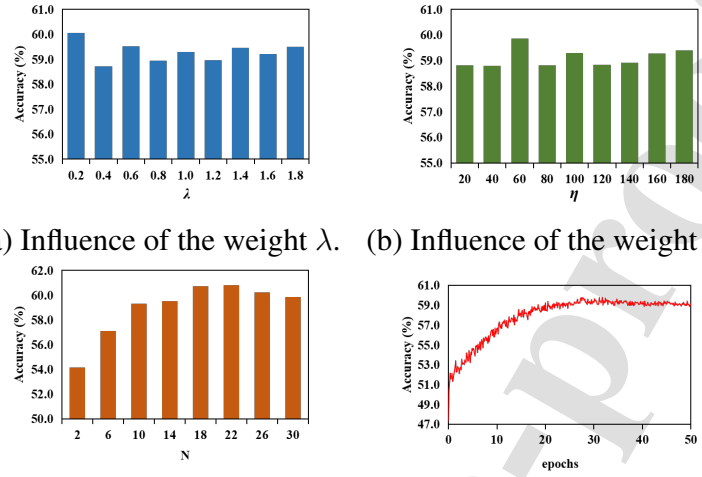
Variants	**Office-31**	**Office-home**	**VisDA**
w/o aggregation	88.4	71.3	82.4
w/ aggregation (Ours)	**90.1**	**72.8**	**85.7**

Table 7: Analysis of different selection methods of proxy source samples.

Method	**Office-31**	**Office-home**	**VisDA**
Random-selected	83.9	69.0	81.9
Entropy-guided	86.3	70.5	72.6
Ours	**90.1**	**72.8**	**85.7**

Table 8: Ablation study on the loss functions.

$\mathcal{L}_{ps}$	$\mathcal{L}_{tgt}^{inter}$	$\mathcal{L}_{tgt}^{intra}$	**Office-31**	**Office-home**	**VisDA**
✓			83.5	66.3	69.6
	✓		89.1	72.4	78.5
		✓	86.7	65.8	84.9
✓	✓		89.3	72.3	78.4
✓		✓	89.9	71.3	84.7
✓	✓	✓	**90.1**	**72.8**	**85.7**



(a) Influence of the weight λ. (b) Influence of the weight η.

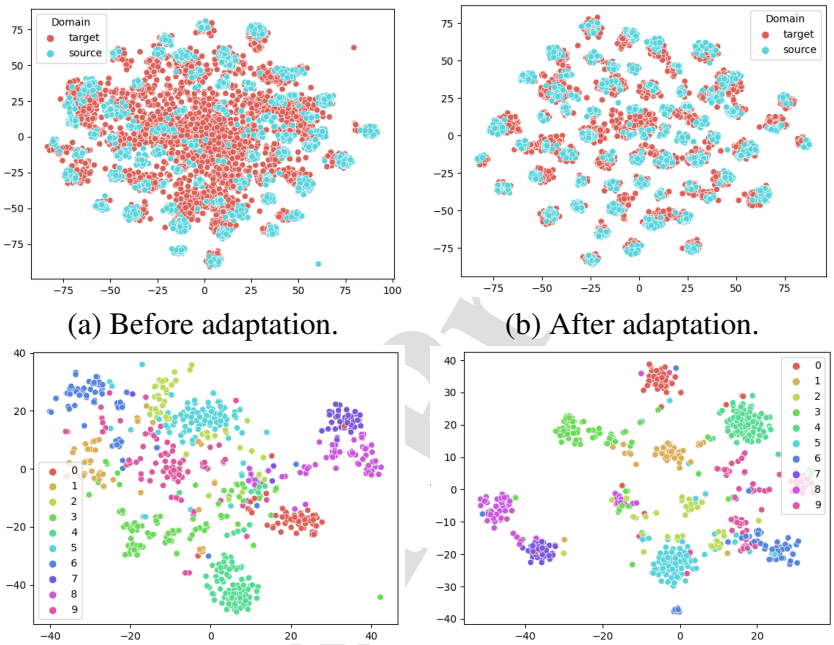
(c) Influence of N. (d) The accuracy curve.

Figure 5: Sensitivity of hyper-parameters of task Ar→Cl on **Office-home**. (a) Influence of the weight λ of $\mathcal{L}_{tgt}^{inter}$; (b) Influence of the weight η of $\mathcal{L}_{tgt}^{intra}$; (c) Influence of the number N per class in the proxy source domain. (d) The accuracy curve of the task Ar→Cl on **Office-home**.

of three loss functions in our method, the training stability, the sensitivity of the important hyper-parameters, and the time costs and computational complexity. We also show the t-SNE visualization results of task Ar→Cl to clearly validate the altering of features.

Effectiveness of the proposed frequency-weighted aggregation soft pseudo label. Our frequency-weighted aggregation strategy (FA) is a soft pseudo label generation method. To verify the influence, we compare our method with three label refinery strategies. 1) MixMatch [22] calculates the soft pseudo label by sharpening and normalizing the predictions directly. 2) ReMixMatch [23] sharpens the predictions first, then multiplies a distribution alignment ratio calculated by the current batch of samples. 3) ATDOC [36] only uses the highest possibilities that are multiplied by a balanced ratio, causing the sums to not be equal to 1, which is not conducive to the calculation of KL divergence. Therefore, we normalize the predictions of ATDOC in our experiments. The results shown in Table 5 demonstrate that the proposed frequency-weighted aggregation module effectively improves the soft label's reliability.

Effectiveness of the aggregation strategy. Our aggregation technique pulls unlabeled target data to semantic neighbors, allowing us to investigate the target



(a) Before adaptation. (b) After adaptation.

(c) Before adaptation. (d) After adaptation.

Figure 6: The t-SNE visualization of task Ar→Cl on **Office-home**. (a) and (b): the unseen source features (blue points) and the target features (red points) before and after adaptation, respectively. (c) and (d): the target features before and after adaptation, respectively. For clarity, we select first 10 classes in the 65 classes on **Office-home**.

domain's structure information and mitigate the detrimental effects of noisy labels. Table 6 shows the variant of ProxyMix without the aggregation approach to demonstrate the usefulness of the aggregation strategy. The accuracy of standard ProxyMix is higher than that of variants without aggregation, demonstrating that leveraging the semantic neighbors' center as the pseudo label is effective and reliable.

Analysis of the construction method of the proxy source domain. To study the influence of the proposed construction method of the class-balanced proxy source domain, we compare ProxyMix with a common-used method, *i.e.*, randomly-selected criterion, entropy-guided criterion, and the baseline method. 1) Randomly-selected: to ensure fairness, we randomly select N samples for each class from the target data to generate a class-balanced proxy source domain based on the classification results of the source model. Because we cannot discover N examples for some difficult classes, we choose the remaining numbers of samples from other classes at random as compensation. 2) Entropy-guided: as commonly used in other works [16], we compare our method with the entropy-guided method. In specific, we calculate the mean entropy e of the source model's prediction on the full target dataset, then obtain a split ratio $\xi = \frac{N(H(f_s(x_t))<e; x_t \in \mathcal{X}_t)}{N(x_t \in \mathcal{X}_t)}$, where $N(\phi)$ denotes the size of the subset formed by samples which satisfy the condition ϕ, $H(\cdot)$ is the entropy function. Then we compute the class distribution $\{n^k\}_{k=1}^K$ according to the predictions given by the source model, and select $n^k \cdot \xi$ samples with the lowest entropy for each class. The results are shown in Table 7. Random-selected perform unsatisfactory due to the poor confidence of the source model before adaption. Although the entropy criterion reflects the confidence of the prediction, it exacerbates the class imbalance problem and leads the model bias to the easier classes, which is not satisfactory in comparison to ours. The proposed prototype-induced method achieves the highest accuracy. We take both confidence and class balance into consideration, and as illustrated in Fig. 2, we observed that the accuracy of the proxy source domain is higher than the entropy criterion.

Ablation studies on the proposed loss functions. To investigate the proposed loss functions, we show the results of variants with different combinations of loss functions in Table 8. As shown, without the proxy source domain classification loss $\mathcal{L}_{ps}$, the accuracy of **Office-31** has the biggest drop. The accuracy of **Office-home** is more likely to be influenced by the inter-domain mixup loss $\mathcal{L}_{tgt}^{inter}$. As for the large-scale dataset **VisDA**, the intra-domain mixup loss $\mathcal{L}_{tgt}^{intra}$ contributes a lot. The effectiveness of $\mathcal{L}_{tgt}^{inter}$ and $\mathcal{L}_{tgt}^{intra}$ also illustrate the reliability of the proposed frequency-weighted soft labels from another perspective.

Table 9: Time cost (s) of one iteration on tasks Ar→Cl, Ar→Pr, Ar→Re on **Office-home**.

Method	Ar→Cl	Ar→Pr	Ar→Re	Avg.
NRC [52]	1.373	1.039	1.196	1.203
AaD [55]	1.548	1.315	2.001	1.621
ProxyMix	1.051	1.098	2.332	1.494

Training stability. We show the accuracy curve of task Ar→Cl on **Office-home** in Fig. 5 (d), the accuracy during training grows up quickly and then converges as we expected. Therefore, the training procedure of ProxyMix is stable and reliable.
Sensitivity of hyper-parameters. To better understand the effects of the hyper-parameters λ, η and N, we explore their performance sensitivity in a single task Ar→Cl on **Office-home** in Fig. 5. The accuracies around $\lambda = 1$ and $\eta = 100$ fluctuate very softly in (a) and (b). The results on the proxy source domain scale are provided in (c), showing that the accuracies change slightly around $N = 20$. Generally, in our method ProxyMix, the hyper-parameters are not sensitive.
Time cost. As can be seen in Algorithm 1, the computational complexity of our algorithm is $O(n)$, and we also provide the time cost (in seconds) of one iteration on the three tasks Ar→Cl, Ar→Pr, and Ar→Re on **Office-home** in Table 9. Our method does not incur much additional time cost, which is acceptable for an offline adaptation method.
t-SNE visualization. To evaluate the effectiveness of ProxyMix, We show the t-SNE visualization[5] of target features on task Ar→Cl in Fig. 6. To validate the effectiveness of domain alignment, we show the features of the unseen source domain (blue points) and the target domain (red points) in (a) and (b). The distribution of target features is closer to the source feature after adaptation as we expected. We also show the target feature distribution of the first 10 classes of **Office-home** in (c) and (d). Benefiting from our frequency-weighted aggregation strategy, the feature clusters after adaptation are compact, and the classification boundary is clear.

5. Conclusion

In this paper, we focus on the source-free domain adaptation problem and propose a simple yet effective method named Proxy-based Mixup training with

[5] https://lvdmaaten.github.io/tsne/

label refinery (ProxyMix). In specific, we treat weights of the fully-connected layer as class prototypes to choose a series of confident samples to construct a class-balanced proxy source domain. Then label information is expected to flow from the pseudo source domain to the unlabeled target domain via mixup training. To enhance mixup training, we further introduce a new pseudo label refinery strategy, which combines frequency-weighted sharpening and neighborhood aggregation to obtain reliable soft predictions of unlabeled target data. Experiments on four popular benchmarks prove the effectiveness of ProxyMix without access to source data. Although our method outperforms several UDA methods that are based on source data, we should recognize that removing all noisy labels in an unsupervised manner is still tough. We believe that our work is an attempt in that direction, with the intention of inspiring others in the UDA community.

6. Acknowledgements

This work is partially funded by the National Natural Science Foundation of China (62276256), Beijing Nova Program (Z211100002121108), and the University Synergy Innovation Program of Anhui Province (GXXT-2022-036).

References

[1] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, et al., Analysis of representations for domain adaptation, Proc. NeurIPS (2007).

[2] P. Dai, P. Chen, Q. Wu, X. Hong, Q. Ye, Q. Tian, C.-W. Lin, R. Ji, Disentangling task-oriented representations for unsupervised domain adaptation, IEEE Transactions on Image Processing 31 (2021) 1012–1026.

[3] S. Li, S. Song, G. Huang, Z. Ding, C. Wu, Domain invariant and class discriminative feature learning for visual domain adaptation, IEEE Transactions on Image Processing 27 (9) (2018) 4260–4273.

[4] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: Proc. ICML, 2015, pp. 1180–1189.

[5] M. Long, Z. Cao, J. Wang, M. I. Jordan, Conditional adversarial domain adaptation, in: Proc. NeurIPS, 2018, pp. 1647–1657.

[6] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, M. Chandraker, Learning to adapt structured output space for semantic segmentation, in: Proc. CVPR, 2018, pp. 7472–7481.

[7] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, S. Saminger-Platz, Central moment discrepancy (cmd) for domain-invariant representation learning, arXiv preprint arXiv:1702.08811 (2017).

[8] C. Chen, Z. Fu, Z. Chen, S. Jin, Z. Cheng, X. Jin, X.-S. Hua, Homm: Higher-order moment matching for unsupervised domain adaptation, in: Proc. AAAI, 2020, pp. 3422–3429.

[9] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proc. CVPR, 2017, pp. 7167–7176.

[10] Y. Zou, Z. Yu, B. Kumar, J. Wang, Unsupervised domain adaptation for semantic segmentation via class-balanced self-training, in: Proc. ECCV, 2018, pp. 289–305.

[11] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, Q. Tian, Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations, in: Proc. CVPR, 2020, pp. 3941–3950.

[12] J. Liang, D. Hu, J. Feng, Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, in: Proc. ICML, 2020, pp. 6028–6039.

[13] R. Li, Q. Jiao, W. Cao, H.-S. Wong, S. Wu, Model adaptation: Unsupervised domain adaptation without source data, in: Proc. CVPR, 2020, pp. 9641–9650.

[14] J. Tian, J. Zhang, W. Li, D. Xu, Vdm-da: Virtual domain modeling for source data-free domain adaptation, IEEE Transactions on Circuits and Systems for Video Technology 32 (6) (2021) 3749–3760.

[15] Z. Qiu, Y. Zhang, H. Lin, S. Niu, Y. Liu, Q. Du, M. Tan, Source-free domain adaptation via avatar prototype generation and adaptation, arXiv preprint arXiv:2106.15326 (2021).

[16] J. Liang, D. Hu, Y. Wang, R. He, J. Feng, Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (11) (2021) 8602–8617.

[17] H. Xia, H. Zhao, Z. Ding, Adaptive adversarial network for source-free domain adaptation, in: Proc. CVPR, 2021, pp. 9010–9019.

[18] J. Huang, D. Guan, A. Xiao, S. Lu, Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data, in: Proc. NeurIPS, 2021, pp. 3635–3649.

[19] V. Papyan, X. Han, D. L. Donoho, Prevalence of neural collapse during the terminal phase of deep learning training, Proceedings of the National Academy of Sciences 117 (40) (2020) 24652–24663.

[20] Y. Du, H. Yang, M. Chen, J. Jiang, H. Luo, C. Wang, Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation, arXiv preprint arXiv:2109.04015 (2021).

[21] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, Proc. ICLR (2018).

[22] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C. Raffel, Mixmatch: A holistic approach to semi-supervised learning, in: Proc. NeurIPS, 2019.

[23] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, C. Raffel, Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring, in: Proc. ICLR, 2020.

[24] D. Berthelot, R. Roelofs, K. Sohn, N. Carlini, A. Kurakin, Adamatch: A unified approach to semi-supervised learning and domain adaptation, arXiv preprint arXiv:2106.04732 (2021).

[25] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Proc. ECCV, 2010, pp. 213–226.

[26] Z. Cao, M. Long, J. Wang, M. I. Jordan, Partial transfer learning with selective adversarial networks, in: Proc. CVPR, 2018, pp. 2724–2732.

[27] P. Panareda Busto, J. Gall, Open set domain adaptation, in: Proc. ICCV, 2017, pp. 754–763.

[28] K. You, M. Long, Z. Cao, J. Wang, M. I. Jordan, Universal domain adaptation, in: Proc. CVPR, 2019, pp. 2720–2729.

[29] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: Proc. ICML, 2015, pp. 97–105.

[30] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, H. T. Shen, Maximum density divergence for domain adaptation, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (11) (2020) 3918–3930.

[31] M. Wang, P. Li, L. Shen, Y. Wang, S. Wang, W. Wang, X. Zhang, J. Chen, Z. Luo, Informative pairs mining based adaptive metric learning for adversarial domain adaptation, Neural Networks 151 (2022) 238–249.

[32] N. Ma, J. Bu, L. Lu, J. Wen, S. Zhou, Z. Zhang, J. Gu, H. Li, X. Yan, Context-guided entropy minimization for semi-supervised domain adaptation, Neural Networks 154 (2022) 270–282.

[33] B. Sun, K. Saenko, Deep coral: Correlation alignment for deep domain adaptation, in: Proc. ECCV Workshops, 2016, pp. 443–450.

[34] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, Cycada: Cycle-consistent adversarial domain adaptation, in: Proc. ICML, 2018, pp. 1989–1998.

[35] Y. Jin, X. Wang, M. Long, J. Wang, Minimum class confusion for versatile domain adaptation, in: Proc. ECCV, 2020, pp. 464–480.

[36] J. Liang, D. Hu, J. Feng, Domain adaptation with auxiliary target domain-oriented classifier, in: Proc. CVPR, 2021, pp. 16632–16642.

[37] H. Liu, J. Wang, M. Long, Cycle self-training for domain adaptation, in: Proc. NeurIPS, 2021, pp. 22968–22981.

[38] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, W. Zhang, Adversarial domain adaptation with domain mixup, in: Proc. AAAI, 2020, pp. 6502–6509.

[39] Y. Wu, D. Inkpen, A. El-Roby, Dual mixup regularized learning for adversarial domain adaptation, in: Proc. ECCV, 2020, pp. 540–555.

[40] J. Liang, R. He, T. Tan, A comprehensive survey on test-time adaptation under distribution shifts, arXiv preprint arXiv:2303.15361 (2023).

[41] M. Boudiaf, R. Mueller, I. Ben Ayed, L. Bertinetto, Parameter-free online test-time adaptation, in: Proc. CVPR, 2022, pp. 8344–8353.

[42] Q. Wang, O. Fink, L. Van Gool, D. Dai, Continual test-time domain adaptation, in: Proc. CVPR, 2022, pp. 7201–7211.

[43] T. Gong, J. Jeong, T. Kim, Y. Kim, J. Shin, S.-J. Lee, Note: Robust continual test-time adaptation against temporal correlation, Proc. NeurIPS (2022) 27253–27266.

[44] D. Brahma, P. Rai, A probabilistic framework for lifelong test-time adaptation, arXiv preprint arXiv:2212.09713 (2022).

[45] J. Yang, R. Yan, A. G. Hauptmann, Cross-domain video concept detection using adaptive svms, in: Proc. ACM-MM, 2007, pp. 188–197.

[46] T. Tommasi, F. Orabona, B. Caputo, Safety in numbers: Learning categories from few examples with multi model knowledge transfer, in: Proc. CVPR, 2010, pp. 3081–3088.

[47] I. Kuzborskij, F. Orabona, Stability and hypothesis transfer learning, in: Proc. ICML, 2013, pp. 942–950.

[48] B. Chidlovskii, S. Clinchant, G. Csurka, Domain adaptation in the absence of source domain data, in: Proc. KDD, 2016, pp. 451–460.

[49] J. Liang, R. He, Z. Sun, T. Tan, Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation, in: Proc. CVPR, 2019, pp. 2975–2984.

[50] S. Roy, M. Trapp, A. Pilzer, J. Kannala, N. Sebe, E. Ricci, A. Solin, Uncertainty-guided source-free domain adaptation, in: Proc. ECCV, 2022, pp. 537–555.

[51] N. Ding, Y. Xu, Y. Tang, C. Xu, Y. Wang, D. Tao, Source-free domain adaptation via distribution estimation, in: Proc. CVPR, 2022, pp. 7212–7222.

[52] S. Yang, J. van de Weijer, L. Herranz, S. Jui, et al., Exploiting the intrinsic neighborhood structure for source-free domain adaptation, Proc. NeurIPS (2021) 29393–29405.

[53] S. Tang, Y. Zou, Z. Song, J. Lyu, L. Chen, M. Ye, S. Zhong, J. Zhang, Semantic consistency learning on manifold for source data-free unsupervised domain adaptation, Neural Networks 152 (2022) 467–478.

[54] Y. Chen, X. Zhu, Y. Li, Y. Li, Y. Wei, H. Fang, Contrast and clustering: Learning neighborhood pair representation for source-free domain adaptation, arXiv preprint arXiv:2301.13428 (2023).

[55] S. Yang, Y. Wang, K. Wang, S. Jui, et al., Attracting and dispersing: A simple approach for source-free domain adaptation, in: Proc. NeurIPS, 2022.

[56] F. Wang, Z. Han, Y. Gong, Y. Yin, Exploring domain-invariant parameters for source free domain adaptation, in: Proc. CVPR, 2022, pp. 7151–7160.

[57] M. Jing, X. Zhen, J. Li, C. G. Snoek, Variational model perturbation for source-free domain adaptation, arXiv preprint arXiv:2210.10378 (2022).

[58] J. Lee, D. Jung, J. Yim, S. Yoon, Confidence score for source-free unsupervised domain adaptation, in: Proc. ICML, 2022, pp. 12365–12377.

[59] H. Yan, Y. Guo, C. Yang, Source-free unsupervised domain adaptation with surrogate data generation, in: Proc. BMVC, 2021.

[60] J. E. Van Engelen, H. H. Hoos, A survey on semi-supervised learning, Machine Learning 109 (2020) 373–440.

[61] X. Wang, D. Kihara, J. Luo, G.-J. Qi, Enaet: A self-trained framework for semi-supervised and supervised learning with ensemble transformations, IEEE Transactions on Image Processing 30 (2020) 1639–1647.

[62] Y. Qin, H. Wu, X. Zhang, G. Feng, Semi-supervised structured subspace learning for multi-view clustering, IEEE Transactions on Image Processing 31 (2021) 1–14.

[63] J. Li, S. Wu, C. Liu, Z. Yu, H.-S. Wong, Semi-supervised deep coupled ensemble learning with classification landmark exploration, IEEE Transactions on Image Processing 29 (2019) 538–550.

[64] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, arXiv preprint arXiv:1610.02242 (2016).

[65] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, C.-L. Li, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, in: Proc. NeurIPS, 2020, pp. 596–608.

[66] H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, M. Savvides, Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning, arXiv preprint arXiv:2301.10921 (2023).

[67] R. Müller, S. Kornblith, G. Hinton, When does label smoothing help?, Proc. NeurIPS (2019).

[68] D. Rukhovich, D. Galeev, Mixmatch domain adaptaion: Prize-winning solution for both tracks of visda 2019 challenge, arXiv preprint arXiv:1910.03903 (2019).

[69] K. Tanwisuth, X. Fan, H. Zheng, S. Zhang, H. Zhang, B. Chen, M. Zhou, A prototype-oriented framework for unsupervised domain adaptation, Proc. NeurIPS (2021) 17194–17208.

[70] Y. Yang, L. Xie, S. Chen, X. Li, Z. Lin, D. Tao, Do we really need a learnable classifier at the end of deep neural network?, arXiv preprint arXiv:2203.09081 (2022).

[71] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proc. ICML, 2016, pp. 478–487.

[72] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proc. CVPR, 2017, pp. 5018–5027.

[73] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: Proc. CVPR, 2018, pp. 3723–3732.

[74] R. Xu, G. Li, J. Yang, L. Lin, Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation, in: Proc. ICCV, 2019, pp. 1426–1435.

[75] Y. Zhang, H. Tang, K. Jia, M. Tan, Domain-symmetric networks for adversarial domain adaptation, in: Proc. CVPR, 2019, pp. 5031–5040.

[76] Y. Zhang, T. Liu, M. Long, M. Jordan, Bridging theory and algorithm for domain adaptation, in: Proc. ICML, 2019, pp. 7404–7413.

[77] X. Wang, L. Li, W. Ye, M. Long, J. Wang, Transferable attention for domain adaptation, in: Proc. AAAI, 2019, pp. 5345–5352.

[78] G. Yang, H. Xia, M. Ding, Z. Ding, Bi-directional generation for unsupervised domain adaptation, in: Proc. AAAI, 2020, pp. 6615–6622.

[79] H. Tang, K. Chen, K. Jia, Unsupervised domain adaptation via structurally regularized deep clustering, in: Proc. CVPR, 2020, pp. 8725–8735.

[80] X. Gu, J. Sun, Z. Xu, Spherical space domain adaptation with robust pseudo-label loss, in: Proc. CVPR, 2020, pp. 9101–9110.

[81] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Proc. ECCV, 2010, pp. 213–226.

[82] R. Xu, P. Liu, L. Wang, C. Chen, J. Wang, Reliable weighted optimal transport for unsupervised domain adaptation, in: Proc. CVPR, 2020, pp. 4394–4403.

[83] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, K. Saenko, Visda: The visual domain adaptation challenge, arXiv preprint arXiv:1710.06924 (2017).

[84] K. Saito, Y. Ushiku, T. Harada, K. Saenko, Adversarial dropout regularization, arXiv preprint arXiv:1711.01575 (2017).

[85] X. Chen, S. Wang, M. Long, J. Wang, Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation, in: Proc. ICML, 2019, pp. 1081–1090.

[86] C.-Y. Lee, T. Batra, M. H. Baig, D. Ulbricht, Sliced wasserstein discrepancy for unsupervised domain adaptation, in: Proc. CVPR, 2019, pp. 10285–10295.

[87] Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, T. Xiang, Stochastic classifiers for unsupervised domain adaptation, in: Proc. CVPR, 2020, pp. 9111–9120.

[88] C. R. Qi, H. Su, K. Mo, L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proc. CVPR, 2017, pp. 652–660.

[89] C. Qin, H. You, L. Wang, C.-C. J. Kuo, Y. Fu, Pointdan: A multi-scale 3d domain adaption network for point cloud representation, in: Proc. NeurIPS, 2019.

[90] M. Long, J. Wang, G. Ding, J. Sun, P. S. Yu, Transfer feature learning with joint distribution adaptation, in: Proc. ICCV, 2013, pp. 2200–2207.

[91] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. CVPR, 2016, pp. 770–778.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: