# Adversarial Reweighting with $\alpha$-Power Maximization for Domain Adaptation

Xiang Gu[1], Xi Yu[1], Yan Yang[1], Jian Sun[1*], Zongben Xu[1]

[1]School of Mathematics and Statistics, Xi'an Jiaotong University, Xianning West Road, Xi'an, 710049, Shaanxi, China.

*Corresponding author(s). E-mail(s): jiansun@xjtu.edu.cn;
Contributing authors: xianggu@stu.xjtu.edu.cn; ericayu@stu.xjtu.edu.cn;
yangyan@xjtu.edu.cn; zbxu@xjtu.edu.cn;

**Abstract**

The practical Domain Adaptation (DA) tasks, *e.g.*, Partial DA (PDA), open-set DA, universal DA, and test-time adaptation, have gained increasing attention in the machine learning community. In this paper, we propose a novel approach, dubbed Adversarial Reweighting with $\boldsymbol{\alpha}$-Power Maximization (ARPM), for PDA where the source domain contains private classes absent in target domain. In ARPM, we propose a novel adversarial reweighting model that adversarially learns to reweight source domain data to identify source-private class samples by assigning smaller weights to them, for mitigating potential negative transfer. Based on the adversarial reweighting, we train the transferable recognition model on the reweighted source distribution to be able to classify common class data. To reduce the prediction uncertainty of the recognition model on the target domain for PDA, we present an $\boldsymbol{\alpha}$-power maximization mechanism in ARPM, which enriches the family of losses for reducing the prediction uncertainty for PDA. Extensive experimental results on five PDA benchmarks, *i.e.*, Office-31, Office-Home, VisDA-2017, ImageNet-Caltech, and DomainNet, show that our method is superior to recent PDA methods. Ablation studies also confirm the effectiveness of components in our approach. To theoretically analyze our method, we deduce an upper bound of target domain expected error for PDA, which is approximately minimized in our approach. We further extend ARPM to open-set DA, universal DA, and test time adaptation, and verify the usefulness through experiments.

**Keywords:** Partial domain adaptation, adversarial reweighting, adversarial training, $\boldsymbol{\alpha}$-power maximization, Wasserstein distance

## 1 Introduction

Deep learning approaches have achieved great success in visual recognition [1–3], but at the expense of laborious large-scale training data annotation. To alleviate the burden of data labeling, Domain Adaptation (DA) transfers the knowledge from a related but different source domain with rich labels to the label-scarce target domain. DA methods mainly learn the transferable model for the target domain by self-training [4–7] or by mitigating the domain shift using moment matching [8–12] or adversarial training [11, 13–15]. Conventional unsupervised DA is the closed-set DA setting, which assumes known target label space (identical to source label space) that is of the "closed-world" paradigm. However, it is often not easy to find a source domain with identical label space to the target domain in practice. Therefore, DA with label space mismatch, *e.g.*,

Partial Domain Adaptation (PDA) [16, 17], open-set DA [18, 19], and universal DA [20, 21], has gained increasing attention in the machine learning community. PDA, open-set DA, and universal DA are related to the more realistic "open-world" paradigm. Specifically, open-world visual recognition does not assume a fixed set of categories (*i.e.*, label space) as in closed-world visual recognition. For PDA, the target label space, being a subset of the source label space, is not fixed, because there exist numerous possible subsets of the source label space. For open-set DA and universal DA, the target domain contains unknown/open classes that are absent in the source domain. Another practical DA setting is the Test-Time Adaptation (TTA) [22], allowing model adaptation at test time. This paper first focuses on the methodology design for PDA, and then extends the developed approach to open-set DA, universal DA, and TTA.

PDA [16, 17, 23, 24] tackles the setting that the source domain contains private classes absent in the target domain, while the target domain classes belong to the set of source domain classes. Besides the domain shift between source and target domains, another main challenge of PDA is the possible negative transfer [25] (see Sect. 3.1), *i.e.*, the knowledge from source domain harms the learning in the target domain, caused by the source-private class data. To mitigate the negative transfer, previous PDA methods [16, 17, 23, 26–30] commonly reweight the source domain data to decrease the importance of data belonging to the source-private classes. The target and reweighted source domain data are used to train the feature extractor by adversarial training [16, 17, 23, 26, 28, 30] or kernel mean matching [27, 29] to align distributions in feature space.

In this paper, we propose a novel approach, dubbed Adversarial Reweighting with $\alpha$-Power Maximization (ARPM), for PDA. To alleviate the potential negative transfer caused by source-private class data, we propose an adversarial reweighting model to reweight the source domain data to decrease the importance of source-private class data in adaptation by assigning them with smaller weights. The learning of source data weights is conducted by minimizing the Wasserstein distance between the target distribution and the reweighted source distribution. The intuition is that the source domain common class data are possibly closer to the target domain data than the source-private class data. This is reasonable and is the assumption taken in [26], and otherwise, PDA could be hardly realized. Using the dual formulation of the Wasserstein distance, the idea is further transformed into an adversarial reweighting model, in which we introduce a discriminator to distinguish domains and adversarially learn the source data weights to fool the discriminator.

Based on the reweighted source data distribution, we define a reweighted classification loss to train the model to recognize objects of common classes, in which the importance of source-private class data is reduced using the learned data weights. Inspired by [28] that bridges domain gap in feature space by entropy minimization [31] to reduce the prediction uncertainty[1] of recognition model on target domain, we also aim to reduce the prediction uncertainty on target domain. Instead of entropy minimization, we propose an $\alpha$-power maximization mechanism that maximizes the sum of $\alpha$-power of the classification score outputted by the recognition model. The $\alpha$-power maximization enriches the family of losses for minimizing the prediction uncertainty. We experimentally show that the $\alpha$-power maximization could be more effective for PDA than the widely adopted entropy minimization [31]. We also utilize the neighborhood reciprocity clustering [32], which is shown to be effective for closed-set DA, to enforce the robustness of the recognition model for PDA.

The above techniques are unified in our total training loss. To train the recognition model, we design an iterative training algorithm that alternately updates the parameters of the recognition model and learns the source domain data weights by solving the adversarial reweighting model. To evaluate our proposed method, we apply our approach to the PDA tasks on five benchmark datasets: Office-31, Office-Home, VisDA-2017, ImageNet-Caltech, and DomainNet. On all five datasets, our proposed ARPM outperforms the recent PDA methods. Ablation studies and empirical analysis also show the effectiveness of each component in our method.

[1] In this paper, by "prediction uncertainty", we refer to the uncertainty of the classification probability distribution (classification score) outputted by the recognition model, *e.g.*, the uniform distribution has larger uncertainty while the one-hot distribution has smaller uncertainty.

To further theoretically analyze our method, we study the theoretical analysis of PDA from the perspective of robustness and prediction uncertainty of the recognition model. More specifically, we prove theoretically that the expected error of the recognition model on target domain can be bounded by the expected error on source domain common class data, and the robustness and prediction uncertainty on target domain of the recognition model. Our approach approximately realizes the minimization of this bound so as to minimize the expected error on target domain.

Additionally, we extend our approach to open-set DA, universal DA, and TTA. For open-set DA, the target domain contains private classes that are absent in the source domain. For universal DA, both source and target domains possibly contain private classes. The goals of open-set and universal DA are to identify the target-private class data as the "unknown" class and meanwhile classify the target domain common class data. To extend our approach to open-set and universal DA, we apply our adversarial reweighting model to reweight target domain data, such that the target domain common (*resp.*, private) data are assigned with larger (*resp.*, smaller) weights. Based on learned weights, we reduce (*resp.*, increase) the prediction uncertainty of target domain possibly common (*resp.*, private) data using our $\alpha$-power loss. As a result, the target-private class can be identified based on prediction uncertainty. For TTA, the goal is to evaluate the model on a target domain that may be different from the source domain in data distribution. Different from vanilla machine learning which directly makes predictions for a mini-batch of test samples at test time, TTA allows adapting the model for a few steps on the mini-batch of test samples in an unsupervised manner and then makes predictions for them. Inspired by the TTA method [22] that updates the parameters of the batch normalization (BN) layers by entropy minimization for one step, we update the parameters of the BN layers by our proposed $\alpha$-power maximization for one step to achieve the extension to TTA. Experiments show the usefulness of our approach for open-set DA, universal DA, and TTA.

Our contributions are summarized as follows:

- We propose a novel ARPM approach for PDA. In ARPM, we propose an adversarial reweighting model for learning to reweight source domain data to decrease the importance of source-private class data in adaptation for PDA. We also propose the $\alpha$-power maximization mechanism to reduce the prediction uncertainty.
- We present a theoretical bound of PDA based on the robustness and prediction uncertainty. We analyze that our proposed ARPM can realize the minimization of the bound.
- Extensive experimental results show the superiority of ARPM against recent PDA methods, along with sufficient ablation studies verifying the effectiveness of each component.
- We extend our approach to open-set DA, universal DA, and TTA that are closely related to "open-world vision recognition".

This paper extends our conference version [33] published at NeurIPS, in which we devised the adversarial reweighting model and reduced the prediction uncertainty by entropy minimization for PDA. In this journal version, we make the following additional contributions. (1) We propose to maximize the sum of $\alpha$-power of the classification score outputted by the recognition model, enriching the family of losses to minimize the prediction uncertainty. We experimentally show that the $\alpha$-power maximization could be more effective for PDA than the widely adopted entropy minimization. (2) We present a theoretical analysis for PDA based on the prediction uncertainty and robustness of the recognition model, which theoretically grounds our approach. (3) We also enhance the robustness of the recognition model using neighborhood reciprocity clustering. (4) To ensure reproducibility, more techniques, *e.g.*, spectral normalization to the discriminator and initializing the classification layer using PCA, are introduced. (5) The performance of our approach is further improved compared with the conference version, and the journal version of our method outperforms recent PDA methods. (6) We extend our approach to other "open-world" tasks, including open-set DA, universal DA, and TTA. (7) More related works are included and summarized. (8) The paper is restructured and rewritten to include the above contributions better.

In the following sections, we summarize the related works in Sect. 2, elaborate our ARPM

approach in Sect. 3, and present the theoretical analysis in Sect. 4. Section 5 discusses the experimental results.

## 2 Related Work

We summarize the related closed-world DA, PDA, open-set DA, and universal DA approaches below.

**Closed-world domain adaptation.** Unsupervised DA [25] aims to transfer knowledge learned from the labeled source domain to the unlabeled target domain, which generally assumes that the source and target domains share the same label space. A group of unsupervised DA methods [10–12, 34–36] attempt to reduce the distribution gap between the source and target domains by moment matching. Another line of methods [13, 37–43] alleviate the domain discrepancy by introducing a domain discriminator to discriminate domains and training the feature extractor to fool the discriminator in an adversarial manner, to learn domain-invariant features. Recently, DA approaches based on self-training [5–7], transferable attention [44], neighborhood consistency [32, 45], progressive adaptation [46], and other techniques [47, 48] without domain alignment have been proposed, achieving promising results. Differently, we mainly tackle the PDA setting by proposing a novel adversarial reweighting with $\alpha$-power minimization method for PDA. In methodology, our method may be mostly related to the adversarial training-based methods [13, 37–41]. Different from them, our adversarial reweighting model conducts the adversarial training by learning to reweight data to fool the discriminator instead of training the feature extractor as in these methods.

**Partial domain adaptation.** SAN [16] and IWAN [23] circumvent negative transfer by training the domain discriminator with reweighted source domain samples. PADA [17] and ETN [26] reweight source domain data in losses for training both the classifier and the domain discriminator. DRCN [27] uses reweighted class-wise domain alignment with a plug-in residual block that automatically uncovers the most relevant source domain classes to target domain data. TSCDA [29] introduces a soft-weighed maximization mean discrepancy criterion to partially align feature distributions to alleviate negative transfer, and proposes a target-specific classifier to further address the classifier shift. SLM [49] exploits "select", "label" and "mix" modules to mitigate negative transfer, enhance discriminability of features, and learn domain-invariant latent space, respectively. ISRA [50] aligns the source and target data distributions based on the implicit semantic topics shared between two domains that are extracted by a plug-in module, to boost the positive transfer. IDSP [47] introduces intra-domain structure preserving without domain alignment, achieving improved results. MOT [51] utilizes a masked optimal transport on conditional distribution by defining the mask using label information to align class-wise distributions for PDA. Different from the above PDA methods, we adversarially learn to reweight the source domain data to decrease the importance of source-private class data in the classification loss, and propose $\alpha$-power maximization to reduce the prediction uncertainty.

**Open-set domain adaptation.** Busto and Gall [18] first study the setting that both the source and target domains contain private categories in addition to the common categories. Saito *et al.* [19] consider the problem setting that the source domain only covers a subset of the target domain label space, which is a common setting in the other open-set DA methods [19, 24, 52–60]. STA [52] adopts a coarse-to-fine weighting mechanism to progressively separate the target domain data into known and unknown classes. Feng *et al.* [24] exploit the semantic structure of open-set data by semantic categorical alignment and contrastive mapping to encourage the known classes more separable and push the unknown class away from the decision boundary. Baktashmotlag *et al.* [53] tackle the open-set DA problem with a method based on subspace learning that models the common classes by a shared subspace and the unknown classes by a private subspace. The method in [57] introduces a graph learning-based adversarial training strategy to align the known class samples from target domain with samples from source domain. Pan *et al.* [58] augment Self-Ensembling for both closed-set and open-set DA scenarios by integrating category-agnostic clusters into DA procedure. Bucci *et al.* [55] utilize a new

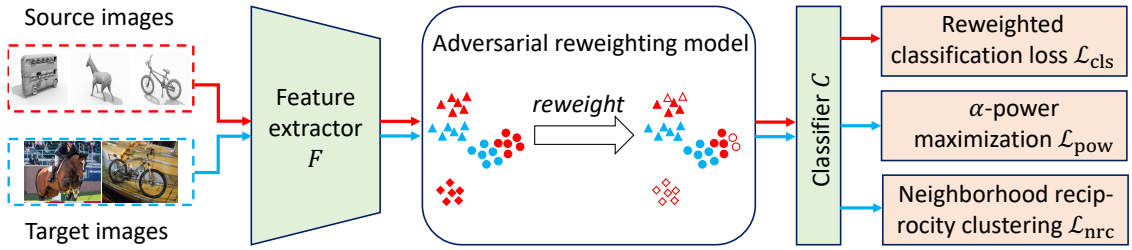



**Fig. 1** Architecture of our ARPM. Red (*resp.* blue) arrows indicate the computational flow for source (*resp.* target) domain data. Both source and target images are mapped to feature space by the feature extractor. Our adversarial reweighting model automatically reweights the importance of source domain data to match the target domain distribution in feature space to decrease the importance of the data of source-private class data. We then define a reweighted classification loss on the reweighted source domain data distribution to train the recognition model to classify common class data. An $\alpha$-power maximization is proposed to reduce the prediction uncertainty on the target domain. We also utilize neighborhood reciprocity clustering [32] to impose the robustness of the recognition model on the target domain.

open-set metric that properly balances the contribution of recognizing the known classes and rejecting the unknown samples, and investigate the self-supervised task of rotation recognition for facilitating open-set DA. Jing *et al.* [60] develop structure-preserving partial alignment to recognize the seen categories and discover the unknown classes. ANNA [61] tackles Open-set DA utilizing front-door adjustment theory. Different from the above methods, we propose the adversarial reweighting model to identify target-private class data for open-set DA. We further respectively decrease and increase the prediction uncertainty of the recognition model on target domain common and private class data, to classify/detect the common/private class data.

**Universal domain adaptation.** UAN [20] proposes a criterion to quantify sample-level transferability based on entropy and domain similarity, thereby promoting the adaptation in the automatically discovered common label set and recognizing the "unknown" samples successfully. CMU [21] designs a better criterion based on a mixture of entropy, confidence, and consistency from a multi-classifier ensemble model to measure sample-level transferability. DANCE [62] uses entropy-based feature alignment and rejection to align target domain features with the source domain or reject the target domain features as unknown categories based on their entropy. OVANet [63] introduces one-vs-all classifiers for each class to automatically learn the threshold for identifying the unknown class data. DCC [64] proposes a cluster-based method to exploit the domain consensus knowledge to discover discriminative clusters for separating the private classes from the common ones in target domain. The method in [65] explores the intrinsic geometrical relationship between the two domains and designs a universal incremental classifier to separate "unknown" samples. Motivated by Bag-of-visual-Words, the method in [66] introduces subsidiary prototype-space alignment to tackle universal DA, avoiding negative transfer. GLC [67] introduces a global and local clustering learning technique for source-free universal DA. PPOT [68] tackle universal DA based on a proposed prototypical partial optimal transport model to identify private class data. SAKA [69] introduces knowability-guided detection of known and unknown samples and refines target pseudo labels based on neighborhood consistency. Differently, we propose the novel adversarial reweighting model to reweight data for identifying the private class data of target domain, and perform domain adaptation relying on reducing/increasing prediction uncertainty based on the learned data weights.

# 3 Method

PDA assumes two related but different distributions, namely source distribution $P$ over space $\mathcal{X} \times \mathcal{Y}$ and the target distribution $Q$ over space $\mathcal{X} \times \mathcal{Y}_{\text{com}}$, where $\mathcal{Y}_{\text{com}} \subset \mathcal{Y}$. In training, we are given labeled source samples $S = \{(\mathbf{x}_i^s, y_i)\}_{i=1}^m$ drawn *i.i.d.* from $P$, and unlabeled target samples $T = \{\mathbf{x}_j^t\}_{j=1}^n$ drawn *i.i.d.* from $Q_{\mathbf{x}}$ where $Q_{\mathbf{x}}$ is the marginal distribution of $Q$ in space

$\mathcal{X}$. The goal of PDA is to train a recognition model using the training samples to predict the class labels of target samples. $\mathcal{Y}_{\text{com}} \subset \mathcal{Y}$ implies that the source domain contains private classes absent in the target domain, which may cause negative transfer in adaptation (see Sect. 3.1). In this paper, we implement the recognition model using deep neural networks. Specifically, the recognition model is composed of a feature extractor $F$ (*e.g.*, ResNet [1]) and a classifier $C$. Detailed architectures of $F$ and $C$ will be given in Sect. 3.5.

To tackle the PDA task, we propose a novel approach, dubbed Adversarial Reweighting with $\alpha$-Power Maximization (ARPM). The overall framework of ARPM is illustrated in Fig. 1. We apply the feature to the input images to extract features for both source and target domains. In the feature space, we propose the adversarial reweighting model to reweight source features such that the source-private class features are assigned smaller weights. We then perform PDA based on the adversarial reweighting. Specifically, on the reweighted source domain data distribution, we define a reweighted classification loss to train the recognition model to be able to classify common class data. On the target domain data, we propose an $\alpha$-power maximization mechanism to reduce the prediction uncertainty of the recognition model. We also utilize the neighborhood reciprocity clustering [32] to enforce the robustness of the recognition model on target domain data. We next discuss our intuitive motivation in Sect. 3.1, the adversarial reweighting model in Sect. 3.2, and adaptation based on adversarial reweighting in Sect. 3.3, in which we introduce the reweighted classification loss, the $\alpha$-power maximization, and the neighborhood reciprocity clustering, followed by our training algorithm in Sect. 3.4. Finally, we extend our method to open-set DA and universal DA in Sect. 3.6 and to TTA in Sect. 3.7.

## 3.1 Motivation

We explain the negative transfer in PDA and the intuitive motivation of our method as follows.

**Negative transfer.** The challenges of PDA arise from the distribution difference and the possible negative transfer caused by the source-private class data in adaptation. To enable the recognition model to be transferred from source to target domain, DA methods often align source and target distributions in feature space to adapt the feature extractor to tackle the challenge of distribution difference. However, some of the target domain data are unavoidably aligned with the source-private class data if directly aligning distributions, and thus are incorrectly recognized, as illustrated in Fig. 2. In other words, the source-private class data can cause negative transfer when aligning distributions in PDA, *i.e.*, these source-private class data harm the learning in the target domain.

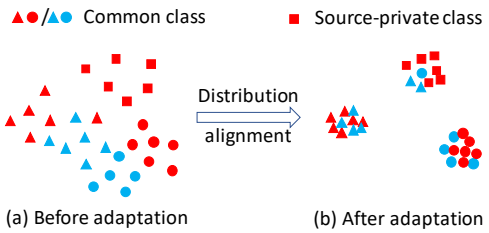



**Fig. 2** Illustration of negative transfer caused by the source-private class data in PDA. The source and target features are respectively in red and blue. Some of the target domain samples are unavoidably aligned with the source-private class data in feature adaptation by distribution alignment, and are incorrectly recognized by the recognition model.

**Intuitive motivation.** Figure 3 shows the motivations of our approach. Intuitively, our adversarial reweighting model aims to learn to reweight source domain data to assign smaller weights to source-private class data, as illustrated in Fig. 3(b). We then define the classification loss on the reweighted source domain data to train the recognition model, in which the source-private class data are less important because they are reweighted by smaller weights. The trained recognition model could be mainly discriminative on source domain common class data, *i.e.*, the predictions are certain on these data. We propose the $\alpha$-power maximization to reduce the prediction uncertainty on the target domain, as will be discussed in Sect. 3.3. By using our $\alpha$-power maximization loss to train the feature extractor, the learned target features will be pushed toward the source common class features to achieve a lower prediction uncertainty, as shown in Fig. 3(c). After adaptation, the target features will be aligned with source domain common class features, as illustrated by Fig. 3(d) (also Fig. 13 on real

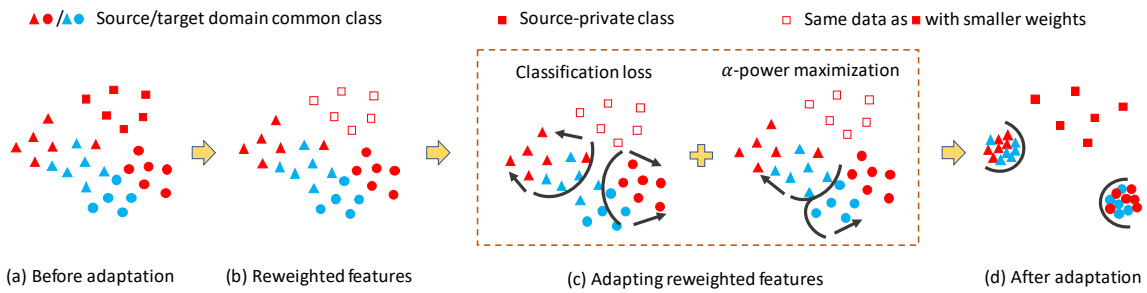



**Fig. 3** Intuitive motivations of ARPM. We reweight source domain data by our adversarial reweighting model to assign smaller weights to source-private class data. The classification loss can enforce lower prediction uncertainty mainly on source domain common class data. We propose the $\alpha$-power maximization to lower prediction uncertainty on target samples. Intuitively, to achieve lower prediction uncertainty, the target samples will be pushed toward the regions of source domain common class data.

data). Therefore, the negative transfer could be alleviated in our approach.

## 3.2 Adversarial Reweighting

We follow [26] to assume that the source domain data of common classes $\mathcal{Y}_{\text{com}}$ are closer to the target domain data than the source domain data belonging to the source-private classes $\mathcal{Y}\backslash\mathcal{Y}_c$. This is reasonable, and otherwise, PDA could be hardly realized. We then learn the weights of source domain data by minimizing the Wasserstein distance between the reweighted source and target distributions. The weight learning process is formulated as an adversarial reweighting model. Figure 4 illustrates our idea. We first introduce the Wasserstein distance.

**Wasserstein distance.** The Wasserstein distance is a metric from optimal transport that measures the discrepancy between two distributions. The Wasserstein distance between distributions $\mu$ and $\nu$ is defined by $W(\mu,\nu) = \min_{\pi\in\Pi}\mathbb{E}_{(\mathbf{x},\mathbf{x}')\sim\pi}\|\mathbf{x}-\mathbf{x}'\|$, where $\Pi$ is the set of couplings of $\mu$ and $\nu$, *i.e.*, $\Pi = \{\pi|\int\pi(\mathbf{x},\mathbf{x}')\mathrm{d}\mathbf{x}' = \mu(\mathbf{x}), \int\pi(\mathbf{x},\mathbf{x}')\mathrm{d}\mathbf{x} = \nu(\mathbf{x}')\}$, and $\|\cdot\|$ is the $l_2$-norm. Leveraging the Kantorovich-Rubinstein duality, the Wasserstein distance has the dual form of $W(\mu,\nu) = \max_{\|g\|_L\leq 1}\mathbb{E}_{\mathbf{x}\sim\mu}g(\mathbf{x}) - \mathbb{E}_{\mathbf{x}'\sim\nu}g(\mathbf{x}')$, where the maximization is over all 1-Lipschitz functions $g:\mathbb{R}^d\to\mathbb{R}$. To compute the Wasserstein distance, we parameterize $g$ by a neural network $D$ (called discriminator). Then, the Wasserstein distance becomes

$$W(\mu,\nu)\approx\max_{\|D\|_L\leq 1}\mathbb{E}_{\mathbf{x}\sim\mu}D(\mathbf{x})-\mathbb{E}_{\mathbf{x}'\sim\nu}D(\mathbf{x}'). \quad (1)$$

In the conference version [43], we enforce the constraint in Eq. (1) with the gradient penalty technique as in [70], which adds in a regularization term $-\beta\mathbb{E}_{\tilde{\mathbf{x}}\sim\tilde{P}_{\mu,\nu}}(\|\nabla_{\tilde{\mathbf{x}}}D(\tilde{\mathbf{x}})\|-1)^2$ to the objective function in Eq. (1), and $D$ is unconstrained. $\tilde{P}_{\mu,\nu}$ denotes the samples uniformly along lines between pairs of points sampled from distributions $\mu$ and $\nu$, *i.e.*, $\tilde{\mathbf{x}}$ is constructed by $\tilde{\mathbf{x}} = \tau\mathbf{x}+(1-\tau)\mathbf{x}'$ where $\mathbf{x}\sim\mu, \mathbf{x}'\sim\nu, \tau\sim\mathcal{U}(0,1)$. However, this strategy introduces additional hyper-parameter $\beta$ and additional randomness from $\tau$, making the results less reproducible. In this journal version, we implement the Lipschitz constraint using spectral normalization [71], which normalizes the weight of each layer in $D$ by its spectral norm. We experimentally show in Sect. 5.5 that the spectral normalization results in better reproducibility of our method than the gradient penalty. Equation (1) allows us to approximately compute the Wasserstein distance using gradient-based optimization algorithms on large-scale datasets. Compared with the other popular statistical distances, *e.g.*, the JS-divergence, the Wasserstein distance enjoys better continuity for learning distributions [72].

### 3.2.1 Adversarial Reweighting Model

We introduce source data weight $\frac{w_i}{m}$ (divided by $m$ is for the convenience of description) for each $i$, and denote $\mathbf{w} = (w_1, w_2, \cdot, w_m)$. Our adversarial reweighting model is defined in the feature space. We extract features by $\mathbf{z}_i^s = F(\mathbf{x}_i^s)$ and $\mathbf{z}_j^t = F(\mathbf{x}_j^t)$ for source and target domain data. The empirical distribution of the target domain features $\hat{Q}_{\mathbf{z}}$ is denoted as $\hat{Q}_{\mathbf{z}} = \frac{1}{n}\sum_{j=1}^n\delta_{\mathbf{z}_j^t}$.

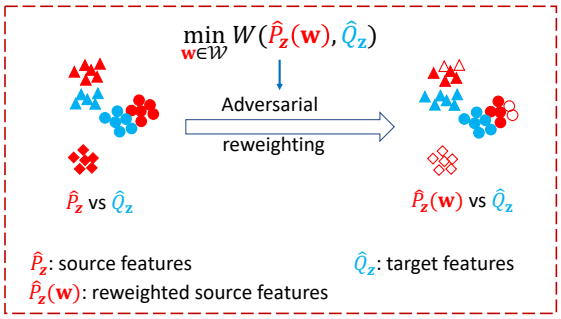



**Fig. 4** We minimize the Wasserstein distance between reweighted source feature distribution $\hat{P}_{\mathbf{z}}(\mathbf{w})$ and target feature distribution $\hat{Q}_{\mathbf{z}}$ to learn weights $\mathbf{w}$. This idea is further transformed into the adversarial reweighting model.

The reweighted source domain feature distribution using $\mathbf{w}$ is denoted as $\hat{P}_{\mathbf{z}}(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} w_i \delta_{\mathbf{z}_i^s}$. Based on the aforementioned analysis that the source-private class data are more distant from target domain data than the source data of common classes, we minimize the Wasserstein distance between the reweighted source domain and target domain feature distributions to learn the weights (as illustrated in Fig. 4) as follows:

$$\min_{\mathbf{w}\in\mathcal{W}} W(\hat{P}_{\mathbf{z}}(\mathbf{w}), \hat{Q}_{\mathbf{z}}). \tag{2}$$

To avoid the mode collapse, *i.e.*, the reweighted distribution is only supported on a few data, we enforce $\sum_{i=1}^{m}(w_i - 1)^2 < \rho m$, where $\rho$ is a hyper-parameter and is set to 5 in this paper. We will study the effect of $\rho$ in Sect. 5.5. By this constraint, the difference between the learned data weights and the all-one vector (corresponding to unweighted data distribution) is not too large, avoiding the case that most samples are assigned with zero weight. Then, the solution space is $\mathcal{W} = \{\mathbf{w} : \mathbf{w} = (w_1, w_2, \cdots, w_m)^T, w_i \geq 0, \sum_{i=1}^{m} w_i = m, \sum_{i=1}^{m}(w_i - 1)^2 < \rho m\}$. With the approximation of the dual form in Eq. (1), Eq. (2) is transformed to the following adversarial reweighting model:

$$\min_{\mathbf{w}\in\mathcal{W}} \max_{\|D\|_L \leq 1} \frac{1}{m}\sum_{i=1}^{m} w_i D(\mathbf{z}_i^s) - \frac{1}{n}\sum_{j=1}^{n} D(\mathbf{z}_j^t). \tag{3}$$

In Eq. (3), the discriminator $D$ is trained to maximize (*resp.* minimize) the average of its outputs on the source (*resp.* target) domain to discriminate the source and target domain data. Adversarially, the source data weights $\mathbf{w}$ are learned to minimize the reweighted average of the outputs of the discriminator on the source domain. As a result, the source data (closer to the target domain) with smaller discriminator outputs will be assigned with larger weights. We will discuss the adversarial training of Eq. (3) in Sect. 3.4.

## 3.3 Adaptation Based on Adversarial Reweighting

Based on the adversarial reweighting model, we perform PDA by defining a reweighted classification loss on reweighted source domain data, proposing an $\alpha$-power maximization mechanism to reduce prediction uncertainty on target domain, and utilizing the neighborhood reciprocity clustering to enforce the robustness. We next discuss these three techniques.

### 3.3.1 Reweighted Classification loss

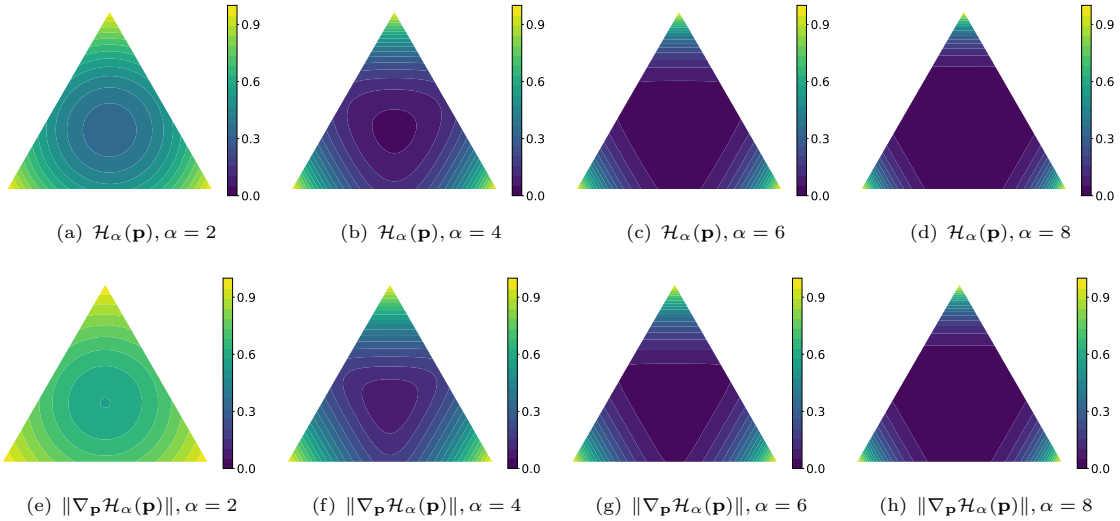
Based on the learned source domain data weights by the adversarial reweighting model, we define the reweighted classification loss on reweighted source domain data to implement the supervised training of the recognition model. The reweighted classification loss is defined using the cross-entropy by

$$\mathcal{L}_{\text{cls}}(F, C) = \frac{1}{m}\sum_{i=1}^{m} w_i \mathcal{J}(C(F(\mathbf{x}_i^s)), y_i^s). \tag{4}$$

Following [73], we employ the cross-entropy loss with label smoothing, *i.e.*, $\mathcal{J}(\mathbf{p}, y) = -\sum_k a_k \log p_k$ for distribution $\mathbf{p} = (p_1, p_2, \cdots, p_{|\mathcal{Y}|})^T$ where $a_k = 1 - \alpha$ if $k = y$, otherwise $a_k = \frac{\alpha}{|\mathcal{Y}|-1}$. $\alpha$ is set to 0.1. In the reweighted classification loss, the importance of the source-private class data is decreased because they are reweighted by smaller weights learned from the adversarial reweighting model. Minimizing the reweighted classification loss encourages the ability of the trained recognition model to classify common class data only.

### 3.3.2 $\alpha$-Power Maximization

Reducing the prediction uncertainty is shown to be effective in DA and even in PDA. In our conference version [33] of this work, we utilize entropy




(a) $\mathcal{H}_\alpha(\mathbf{p}), \alpha = 2$ (b) $\mathcal{H}_\alpha(\mathbf{p}), \alpha = 4$ (c) $\mathcal{H}_\alpha(\mathbf{p}), \alpha = 6$ (d) $\mathcal{H}_\alpha(\mathbf{p}), \alpha = 8$

(e) $\|\nabla_{\mathbf{p}}\mathcal{H}_\alpha(\mathbf{p})\|, \alpha = 2$ (f) $\|\nabla_{\mathbf{p}}\mathcal{H}_\alpha(\mathbf{p})\|, \alpha = 4$ (g) $\|\nabla_{\mathbf{p}}\mathcal{H}_\alpha(\mathbf{p})\|, \alpha = 6$ (h) $\|\nabla_{\mathbf{p}}\mathcal{H}_\alpha(\mathbf{p})\|, \alpha = 8$

**Fig. 5** Contour of (a-d) $\mathcal{H}_\alpha(\mathbf{p})$ and (e-h) gradient norm $\|\nabla_{\mathbf{p}}\mathcal{H}_\alpha(\mathbf{p})\|$ for 3-dimensional distribution $\mathbf{p}$ in probability simplex $\Delta^3$ under different $\alpha$. To show the relative magnitude for different $\mathbf{p}$, we normalize the gradient norm such that the maximum value is 1 in each figure of (e-h).

minimization that is widely adopted in DA methods to reduce the prediction uncertainty. In this journal paper, we propose to maximize the sum of $\alpha$-power of the output of the recognition model. The $\alpha$-power loss is defined by

$$\mathcal{L}_{\text{pow}}(F) = -\frac{1}{n}\sum_{j=1}^{n}\mathcal{H}_\alpha(C(F(\mathbf{x}_j^t))), \qquad (5)$$

where $\mathcal{H}_\alpha(\mathbf{p}) = \sum_k p_k^\alpha$. We empirically set $\alpha = 6$ in experiments. $\mathcal{H}_\alpha(\mathbf{p})$ with $\alpha > 1$ takes its maximum value when $\mathbf{p}$ is a one-hot distribution (with low uncertainty) and minimum value when $\mathbf{p}$ is a uniform distribution (with high uncertainty). Minimizing the $\alpha$-power loss (*i.e.*, maximizing $\mathcal{H}_\alpha$) will reduce the uncertainty of $\mathbf{p}$. We plot the contour of $\mathcal{H}_\alpha(\mathbf{p})$ and its gradient norm $\|\nabla_{\mathbf{p}}\mathcal{H}_\alpha(\mathbf{p})\|$ under different $\alpha$ in Fig. 5. We can see that for larger $\alpha$, the samples with high prediction uncertainty (points near to the center in Figs. 5(a-h)) have smaller or even near-to-zero gradients of $\mathcal{H}_\alpha$, implying that these samples may not contribute to the training. The $\alpha$-power loss enriches the family of losses for reducing prediction uncertainty. Following [23], we use the $\alpha$-power loss to update the feature extractor $F$. We show that $\alpha$-power maximization could be more effective for PDA than entropy minimization in Sect. 5.5.

**Comparison of different uncertainty losses.** In the learning tasks with unlabeled data, *e.g.*, semi-supervised learning and DA, reducing the uncertainty of model's prediction on unlabeled data in training can often improve the performance of the model [23, 31]. The most used uncertainty loss in DA could be conditional entropy. Some methods [27, 74] also investigate the mutual information and the nuclear norm for balancing the prediction over classes. Liu et al. [6] minimize the $\alpha$-Tsallis entropy, *i.e.*, $\frac{1}{\alpha-1}(1-\sum_k p_k^\alpha)$, but choose $\alpha$ in a narrow interval $[1, 2]$, possibly limiting its ability. Chen et al. [75] propose to maximize the square loss of prediction probability, *i.e.*, $\sum_k p_k^2$. Our $\alpha$-power loss is mostly related to the square loss and the $\alpha$-Tsallis entropy. The square loss is a special case ($\alpha = 2$) of the $\alpha$-power loss. Compared with the $\alpha$-Tsallis entropy, the $\alpha$-power loss is possibly more stable to $\alpha$ because of the presence of $\frac{1}{\alpha-1}$ in the $\alpha$-Tsallis entropy. We experimentally find that $2 < \alpha \leq 10$ often yields better results than $1 < \alpha \leq 2$ for PDA tasks, as in Sect. 5.5.

### 3.3.3 Neighborhood Reciprocity Clustering

The robustness or local consistency [32] that enforces the outputs of the recognition model on the neighborhood samples are similar, is proven

effective for closed-set DA. We utilize the neighborhood reciprocity clustering [32] to impose the robustness for PDA in this paper.

We first build the feature and score banks on target domain data by

$$\mathcal{Z} = \{\mathbf{z}_1^t, \mathbf{z}_2^t, \cdots, \mathbf{z}_n^t\}, \mathcal{S} = \{\mathbf{s}_1^t, \mathbf{s}_2^t, \cdots, \mathbf{s}_n^t\} \quad (6)$$

where $\mathbf{z}_j^t = F(\mathbf{x}_j^t)$ is the feature, and $\mathbf{s}_j^t = C(F(\mathbf{x}_j^t))$ is the classification score outputted by the recognition model. At each training step, feature and score banks are updated by replacing the old items with the corresponding items from the current mini-batch samples.

Given target sample $\mathbf{x}_j^t$ with feature $\mathbf{z}_j^t$, the index set of its $K$-nearest neighbors[2] in the feature bank is denoted as $\mathcal{N}_K^j$. To better identify the true neighbors, we introduce the affinity[3] as

$$A_{j,j'} = \begin{cases} 1 & \text{if } j' \in \mathcal{N}_K^j \text{ and } j \in \mathcal{N}_M^{j'}; \\ 0.1 & \text{otherwise.} \end{cases} \quad (7)$$

The intuition of the affinity in Eq. (7) is that if $\mathbf{z}_j^t$ and $\mathbf{z}_{j'}^t$ are both neighbors of each other, they could be true neighbors (reciprocal neighbors). Finally, the neighborhood reciprocity clustering loss is defined as

$$\mathcal{L}_{\text{nrc}}(F) = -\frac{1}{n}\sum_{j=1}^{n}\sum_{j' \in \mathcal{N}_K^j} A_{j,j'} \langle \mathbf{s}_{j'}^t, C(F(\mathbf{x}_j^t)) \rangle. \quad (8)$$

We use $\mathcal{L}_{\text{nrc}}(F)$ to update $F$. Minimizing $\mathcal{L}_{\text{nrc}}(F)$ encourages the similar output of the neighborhood samples so that the robustness could be imposed.

Note that Zhong et al. [76] proposed a $k$-reciprocal encoding method to re-rank the re-identification results for the person re-identification task, which may be related to the neighborhood reciprocity clustering in our approach. They encode the $k$-reciprocal nearest neighbors of a given image into a single vector, which is used for re-ranking under the Jaccard distance to help accomplish the person re-identification task. Differently, we utilize the reciprocal neighbors to enforce the robustness of the deep recognition model, instead of aggregating them as in [76].

[2] Following [32], we use the cosine distance to find the neighbors.

[3] As in [32], we set $K = M = 5$ for VisDA-2017 dataset and $K = 4$, $M = 3$ for the other datasets in experiments.

## 3.4 Training

The overall training loss is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{pow}} + \mathcal{L}_{\text{nrc}}, \quad (9)$$

where $\lambda$ is a hyper-parameter. Note that in the classification loss $\mathcal{L}_{\text{cls}}$ in Eq. (4), the source domain data weights $\mathbf{w}$ should be learned in training. We then devise an iterative training algorithm to alternately train $F$ and $C$, and learn $\mathbf{w}$.

### 3.4.1 Training Algorithm

We initialize $\mathbf{w}$ by $w_i = 1$ for all $i$. Then, we alternately run the following two procedures when training the networks.

**Updating $F$ and $C$ with fixed $\mathbf{w}$.** Fixing $\mathbf{w}$, we update $F$ and $C$ to minimize the loss in Eq. (9) for $N$ steps, using the mini-batch stochastic gradient descent algorithm.

**Updating $\mathbf{w}$ with fixed $F$ and $C$.** Fixing $F$ and $C$, we extract the features for all training data on both source and target domains, and learn $\mathbf{w}$ in Eq. (3). Since Eq. (3) is a min-max optimization problem, we can alternately optimize the weights $\mathbf{w}$ and the discriminator $D$ by fixing the other one as known. To reduce the computational cost, we only perform the alternate optimization once, which yields satisfactory performance in experiments. Therefore, we first fix $w_i = 1$ for all $i$ and optimize $D$ to maximize the objective function in Eq. (3). Then, fixing the discriminator, we optimize $\mathbf{w}$ as follows. We denote $d_i = D(\mathbf{z}_i^s)$ and $\mathbf{d} = (d_1, d_2, \cdots, d_m)^T$. The optimization problem for $\mathbf{w}$ becomes

$$\begin{aligned} &\min_{\mathbf{w}} \mathbf{d}^T \mathbf{w}, \\ &\text{s.t. } w_i \geq 0, \sum_{i=1}^{m} w_i = m, \sum_{i=1}^{m} (w_i - 1)^2 \leq \rho m. \end{aligned} \quad (10)$$

Equation (10) is a second-order cone program. We use the CVXPY [77] package to solve Eq. (10).

The above algorithm faces challenges when $m$ is large. We next discuss how to tackle the cases

**Algorithm 1** Training algorithm.

**Input:** Source and target domain training datasets $S$ and $T$
**Output:** Trained networks $F, C$

$S' = S$
Initialize $\mathbf{w}$ by $w_i = 1$, $i = 1, 2, \cdots, |S'|$
**for** $step = 0, 1, 2, \cdots$ **do**
  Sample a mini-batch data $(X_s, Y_s)$ and $X_t$ from $S'$ and $T$, respectively
  Update $F$ and $C$ with the loss in Eq. (9) computed on $(X_s, Y_s)$ and $X_t$, using the SGD
  **if** $step\%N = 0$ and $step > 0$ **then**
    **if** $|S|$ >1000k **then**
      Randomly select $N'$ samples from $S$ to construct $S'$
    **end if**
    Extract features for all training samples in both $S'$ and $T$
    Train $D$ as in Eq. (1) with spectral normalization to enforce the 1-Lipschitz constraint
    **if** $|S'|$ >20k **then**
      Split source data indexes into $\left[\frac{m}{20\text{k}}\right]$ groups
      Solve Eq. (10) for each group to update $\mathbf{w}$
    **else**
      Solve Eq. (10) for all source data indexes to update $\mathbf{w}$
    **end if**
  **end if**
**end for**

with larger size $m$ of the source dataset. (1) In the first case (20k< $m$ <1000k), solving Eq. (10) for all source data is time-consuming or even infeasible. For such a case, once the discriminator is trained and fixed, we split the source data indexes into several groups and solve the problem (10) for each group. The number of groups is $l = \left[\frac{m}{20\text{k}}\right]$, where $[\cdot]$ is the floor function. The $i$-th group is $\{i, l+i, 2l+i, \cdots\}$. Such a splitting strategy ensures that the empirical feature distribution corresponding to each group can approximate the empirical distribution of all features. (2) In the second case ($m$ >1000k), extracting the features of all source samples is time-consuming. For such a case, we randomly sample a subset (with size $N'$) of the source dataset to learn their weights $\mathbf{w}$, and then update $F$ and $C$ on the subset and weights, and iterate these two above procedures. We give the pseudo-code of the training algorithm in Algorithm 1.

### 3.5 Network Details

For the discriminator $D$, we use the same architecture as [13] (three fully connected layers with 1024, 1024 and 1 nodes respectively), excluding the last sigmoid function. For the feature extractor $F$, we use the ResNet-50 [1] pre-trained on ImageNet [78], excluding the last fully-connected layer. The classifier $C$ is a fully-connected layer. Inspired by [43, 73], we perform $L_2$-normalization after the feature extractor to enforce the same norm[4] of features, and normalize each row of the weight of classifier $C$ to a unit vector[5].

Since the features are normalized, the feature distribution is not a Gaussian distribution. Therefore, the commonly adopted initialization strategies [79, 80] that assume the Gaussian distribution could not be suitable. While the initialization of the classifier can affect the reproducibility as we experimentally show in Sect. 5.5. Following the idea of preserving the variance in [79, 80], we use Principal Component Analysis (PCA) to initialize the weight of the classifier to preserve the feature variance. Specifically, we compute principal components $V = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{|\mathcal{Y}|})$ of target features. We then compute the principal component scores of each source feature and assign the

[4] We set the norm as in [43].
[5] On VisDA-2017 dataset, we do not normalize the weight of $C$. We empirically find that on VisDA dataset, the unnormalized weight of $C$ yields better result.

source feature to the principal component with the largest score. Between the class label and the assigned principal component, we can calculate the confusion matrix $\mathcal{M} \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$, of which the entry $\mathcal{M}_{ij}$ is the ratio of the $i$-th class source samples being assigned to the $j$-th principal component. Finally, the weight $W$ of $C$ is initialized by $W = \mathcal{M}V^T$. This PCA-based initialization strategy reduces the randomness compared with [79, 80], and may achieve better reproducibility, as shown in Sect. 5.5.

### 3.6 Extension to Open-set and Universal DA

We extend our approach to open-set DA and universal DA in this section. In open-set DA, the unlabeled target domain contains private classes that are absent in the source domain. In universal DA, both the labeled source domain and unlabeled target domain possibly contain private classes. The goals of both open-set DA and universal DA are to identify the target-private class data as "unknown" and classify the target domain common class data. To extend our approach ARPM to open-set DA and universal DA, we employ our adversarial reweighting model to reweight target domain data, such that the target domain common (*resp.*, private) class data are assigned larger (*resp.*, smaller) weights. That is, in Eq. (3), we reweight $D(\mathbf{z}_j^t)$ by a weight $w_j$ and $D(\mathbf{z}_i^s)$ is unweighted. Based on the reweighted target domain data, we reduce (*resp.*, increase) the prediction uncertainty on the target domain common (*resp.*, private) class data so that the target-private class data can be detected using the prediction uncertainty.

More specifically, we sort the target samples according to the learned data weights in ascending order to obtain $\mathbf{x}_{(1)}^t, \mathbf{x}_{(2)}^t, \cdots, \mathbf{x}_{(n)}^t$. Since the target samples with larger weights possibly belong to common classes, we define the following reweighted $\alpha$-power loss to reduce the prediction uncertainty on these data:

$$\mathcal{L}_{\text{com}}(F) = -\frac{1}{n\tau}\sum_{l=n-n\tau}^{n} w_{(l)}\mathcal{H}_\alpha(C(F(\mathbf{x}_{(l)}^t))), \tag{11}$$

where we discard the $n-n\tau$ samples with smaller weights to further enforce robustness. We also maximize the following $\alpha$-power loss on the $n\tau$ samples with smaller weights that are more possibly private classes, to increase their prediction uncertainty:

$$\mathcal{L}_{\text{pri}}(F) = -\frac{1}{n\tau}\sum_{l=1}^{n\tau} \mathcal{H}_\alpha(C(F(\mathbf{x}_{(l)}^t))). \tag{12}$$

The total training loss for open-set DA and universal DA is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda'(\mathcal{L}_{\text{com}} - \mathcal{L}_{\text{pri}}). \tag{13}$$

$\mathcal{L}_{\text{cls}}$ is defined in Eq. (4) in the revised paper, in which we do not reweight the source domain data. For universal DA, one may also reweight the source domain data in $\mathcal{L}_{\text{cls}}$ to reduce the importance of source-private class data. In this extension, for the sake of a unified formulation for open-set DA and universal DA, we do not reweight the source domain data.

### 3.7 Extension to TTA

This section extends our approach to TTA. TTA trains the recognition model on a source domain and evaluates it on an unknown target domain. Different from vanilla machine learning which directly makes predictions for a mini-batch of test samples at test time, TTA allows adapting the model for a few steps on the mini-batch of test samples in an unsupervised manner and then makes predictions for them. At the test time of TTA, we are given the source-trained recognition model. The test data arrive sequentially. At each time, we only access a mini-batch of target data $B = \{\mathbf{x}_j^t\}_{j=1}^b$ where $b$ is the batch size. The goal of TTA is to update the model on the mini-batch data $B$ and then make predictions on them. Inspired by TENT [22] that updates the parameters of the BN layers by entropy minimization for one step, we update the parameters of the BN layers by our proposed $\alpha$-power maximization for one step. Specifically, if we denote the set of parameters of the BN layers in the model as $\Theta_{\text{bn}}$, the $\alpha$-power loss is defined as

$$\mathcal{L}_{\text{pow}}(\Theta_{\text{bn}}) = \frac{1}{b}\sum_{j=1}^{b} \mathcal{H}_\alpha(C(F((\mathbf{x}_j^t))). \tag{14}$$

We update $\Theta_{\text{bn}}$ using $\mathcal{L}_{\text{pow}}$ by one-step gradient descent, and then predict the label of $B$. Note that the updated $\Theta_{\text{bn}}$ will be taken as the initial value for the next mini batch. Our approach for TTA is dubbed Test $\alpha$-Power Maximization (TPM).

# 4 Theoretical Analysis

This section presents the theoretical analysis of our method for PDA. We first provide the notations and assumptions, then present an upper bound for PDA, and finally analyze that our proposed ARPM approximately realizes the minimization of the upper bound.

**Notations.** We denote $P^c$ as the source domain common class data distribution, *i.e.*, $P^{\text{c}}(\mathbf{x}, y) = P(\mathbf{x}, y | y \in \mathcal{Y}_{\text{com}})$. For any $\mathbf{x} \in \mathcal{X}$, its neighborhood is defined by $\mathcal{N}(\mathbf{x}) = \{\mathbf{x}' : d(\mathbf{x}, \mathbf{x}') \leq \xi\}$ for some $\xi > 0$, where $d(\cdot, \cdot)$ is the distance on $\mathcal{X}$. For any set $A \subset \mathcal{X}$, we define $\mathcal{N}(A) = \bigcup_{\mathbf{x} \in A} \mathcal{N}(\mathbf{x})$. For convenience, we denote $f : \mathcal{X} \to [0,1]^{|\mathcal{Y}|}$ as the recognition model, *i.e.*, $f = C \circ F$. The decision function corresponding to $f$ is $\tilde{f}$ defined by $\tilde{f}(\mathbf{x}) = \arg\max_{i \in \mathcal{Y}} f(\mathbf{x})_i$. We use $\mathcal{F}/\tilde{\mathcal{F}}$ to denote the sets of all possible $f/\tilde{f}$. For any $f \in \mathcal{F}$, its margin on sample $\mathbf{x}$ is defined by $M(f(\mathbf{x})) = f(\mathbf{x})_{i^*} - \max_{i \neq i^*} f(\mathbf{x})_i$, where $i^* = \arg\max_i f(\mathbf{x})_i$. $1 - M(f(\mathbf{x}))$ reflects the *prediction uncertainty* of $f$. Specifically, if $1 - M(f(\mathbf{x}))$ is smaller, $f(\mathbf{x})$ approaches the one-hot vector leading to low prediction uncertainty. The expected margin of $f$ on distribution $P$ is $M_P(f) = \mathbb{E}_{(\mathbf{x},y)\sim P} M(f(\mathbf{x}))$. The *robustness* of $f$ on distribution $P$ is defined by $R_P(f) = P(\{\mathbf{x}, \exists \mathbf{x}' \in \mathcal{N}(\mathbf{x}), \tilde{f}(\mathbf{x}) \neq \tilde{f}(\mathbf{x}')\})$. For any $i \in \mathcal{Y}_{\text{com}}$, we denote $P_i^{\text{c}} = P^{\text{c}}(\mathbf{x}|y = i)$ as class-wise data distribution of the $i$-th class. $Q_i$ is similarly defined.

**Assumptions.** To develop the theory for PDA, we assume that:
(i) $\forall i \in \mathcal{Y}_{\text{com}}$, $\frac{Q(y=i)}{P^c(y=i)} \leq r$;
(ii) $\forall i, j \in \mathcal{Y}_{\text{com}}$, the supports of $Q_i$ and $Q_j$ are disjoint for $i \neq j$;
(iii) For any $i \in \mathcal{Y}_{\text{com}}$, $\frac{1}{2}(P_i^{\text{c}} + Q_i)$ satisfies $(q, \epsilon)$-constant expansion (see Definition 1) for some $q, \epsilon \in (0, 1)$.

**Definition 1** ($(q, \epsilon)$-constant expansion)**.** *We say that a distribution $P$ satisfies $(q, \epsilon)$-constant expansion for some $q, \epsilon \in (0, 1)$, if for any set $A \subset \mathcal{X}$ with $\frac{1}{2} \geq P(A) \geq q$, we have $P(\mathcal{N}(A) \backslash A) > \min\{\epsilon, P(A)\}$.*

Assumption (i) is realistic because $Q(y = i)$ is finite. Assumption (ii) implies that any target sample has a unique class label. We follow [6, 81, 82] to take the expansion assumption (Assumption (iii)) of the mixture distribution. Intuitively, this assumption indicates that the conditional distributions $P_i^{\text{c}}$ and $Q_i$ are closely located and regularly shaped, enabling knowledge transfer from the source domain to the target domain. Wei et al. [81] justified this assumption on real-world datasets with BigGAN [83].

**Theorem 1.** *Suppose the above Assumptions hold. For any $f \in \mathcal{F}$ and any $\eta \in (0, 1)$, if $f$ is $L$-Lipschiz w.r.t. $d(\cdot, \cdot)$, we have*

$$\begin{aligned} \varepsilon_Q(f) \leq & r\varepsilon_{P^{\text{c}}}(f) + c_1 R_{\frac{P^{\text{c}}+Q}{2}}(f) \\ & + c_2(1 - M_{\frac{P^{\text{c}}+Q}{2}}(f)) + 2rq, \end{aligned} \tag{15}$$

*where the coefficients $c_1 = \frac{2\eta r}{\min\{\epsilon, q\}(1+r)}$ and $c_2 = \frac{2r(1-\eta)}{\min\{\epsilon, q\}(1-2L\xi)(1+r)}$ are constants to $f$.*

The proof is given in Appendix. Theorem 1 implies that the target domain expected error $\varepsilon_Q(f)$ is bounded by the expected error $\varepsilon_{P^{\text{c}}}(f)$ on source domain common class data, the robustness $R_{\frac{P^{\text{c}}+Q}{2}}(f)$ and prediction uncertainty $1 - M_{\frac{P^{\text{c}}+Q}{2}}(f)$ on mixture distribution of source and target domains.

In our method, the adversarial reweighting model aims to assign larger weights to source domain common class data, and the reweighted source data distribution is expected to approach the data distribution of source common classes. Minimizing the classification loss $\mathcal{L}_{\text{cls}}$ on the reweighted source domain data distribution enforces the recognition model to predict the label of source common class data. Therefore, the expected error $\varepsilon_{P^{\text{c}}}(f)$ on source domain common class data could be minimized.

We notice that $R_{\frac{P^{\text{c}}+Q}{2}}(f) = \frac{1}{2}(R_{P^{\text{c}}}(f) + R_Q(f))$. Since $f$ is trained using the classification loss with the class labels on the source domain, the prediction of $\tilde{f}$ on the neighborhood samples should be their class labels and thus are similar, because the neighborhood samples should belong to the same class. This implies that $R_{P^{\text{c}}}(f)$ is minimized. The neighborhood reciprocity clustering

loss $\mathcal{L}_{\text{nrc}}$ encourages similar outputs of the recognition model on neighborhood samples on target domain, which implies $R_Q(f)$ is minimized.

Note that $1 - M_{\frac{P^c+Q}{2}}(f) = \frac{1}{2}(1 - M_{P^c}(f)) + \frac{1}{2}(1 - M_Q(f))$. By the classification loss, the outputs of $f$ on the source domain common class data are near to one-hot vectors, so that $1 - M_{P^c}(f)$ is minimized. Our $\alpha$-power loss $\mathcal{L}_{\text{pow}}$ reduces the prediction uncertainty on the target domain, realizing the minimization of $1 - M_Q(f)$ which reflects the prediction uncertainty on target domain.

# 5 Experiments

We conduct experiments on benchmark datasets to evaluate our ARPM approach, and compare it with recent methods. The source code is available at https://github.com/XJTU-XGU/ARPM.

## 5.1 Setup

For ease of understanding, we discuss the setup for PDA in this section. We will discuss the setup for open-set and universal DA in Sect 5.3 and for TTA in Sect. 5.4.

**Datasets.** *Office-31* dataset [84] contains 4,652 images of 31 categories, collected from three domains: Amazon (A), DSLR (D), and Webcam (W). *ImageNet-Caltech* is built with ImageNet (I) [78] and Caltech-256 (C) [85], respectively including 1000 and 256 classes. *Office-Home* [86] consists of four domains: Artistic (A), Clip Art (C), Product (P), and Real-World (R), sharing 65 classes. *VisDA-2017* [87] is a large-scale challenging dataset, containing two domains: Synthetic (S) and Real (R), with 12 classes. *DomainNet* [88] is another large-scale challenging dataset, composed of six domains with 345 classes.

**Adaptation tasks.** On Office-31, ImageNet-Caltech, Office-Home datasets, we set every domain as the source domain in turn and use each of the rest domain(s) to build the target domain, forming the adaptation tasks. On Office-31, we follow [16] to select images from the 10 categories shared by Office-31 and Caltech-256 [85] to build the target domain in each task. On ImageNet-Caltech dataset, we utilize the 84 shared classes by ImageNet and Caltech-256 to build the target domain in each task. As most networks are pre-trained on the training set of ImageNet, for task C→I, we use images from ImageNet validation set to build the target domain. On Office-Home, we use images of the first 25 classes in alphabetical order to build the target domain in each task. On VisDA-2017, we set Synthetic (S) as source domain and Real (R) as target domain to perform synthetic to real domain transfer, and use the first 6 classes in alphabetical order as the target domain. On DomainNet, since the labels of some domains and classes are very noisy, we follow [95] to adopt four domains (Clipart (C), Painting (P), Real (R), and Sketch (S)) with 126 classes for PDA. We use the first 40 classes in alphabetical order to build the target domain in each task. In each adaptation task, we report the average classification accuracy on target domain.

**Implementation details.** We implement our method using Pytorch [96] on a single Nvidia Tesla v100 GPU. We use the SGD algorithm with momentum 0.9 to update the parameters of $F$ and $C$. The learning rate of $C$ is ten times that of $F$. The parameters of $D$ are updated by the Adam algorithm with learning rate 0.001. Following [13], we adjust the learning rate of $C$ by $\frac{\kappa}{(1+10p)^{0.75}}$, where $p$ represents the training progress linearly changing from 0 to 1. Bath size is set to 64. For Office-Home, ImageNet-Caltech, and DomainNet datasets, we set $\kappa = 0.01$ and $\lambda = 0.3$. Since the training processes on VisDA-2017 and Office-31 datasets converge faster, we set $\kappa$ to 0.001 for VisDA and 0.005 for Office-31. Correspondingly, $\lambda$ is set to 1.0 for VisDA-2017 and Office-31 datasets. $N$ in Algorithm 1 is set to 500 for Office-Home and Office-31 datasets, and is set to 1000 for the larger VisDA-2017, DomainNet, and ImageNet-Caltech datasets. $N'$ is set to $64 * 2000$.

On VisDA-2017, ImageNet-Caltech, and DomainNet datasets, we sample the mini-batch data according to the learned weights using a reweighted random sampler and then calculate the unweighted classification loss on the sampled mini-batch data in training. We find that this strategy makes training more stable on these datasets than the commonly used strategy that we first uniformly sample mini-batch data and then reweight the classification loss for each sample in the mini-batch. The reason could be that the number of samples with zero weights is large in these large-sized datasets and the uniformly

**Table 1** Accuracy (%) on Office-Home dataset for PDA. The best results are bolded. *AR is our conference version.

| Method | A→C | A→P | A→R | C→A | C→P | C→R | P→A | P→C | P→R | R→A | R→C | R→P | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet-50 [1] | 46.3 | 67.5 | 75.9 | 59.1 | 59.9 | 62.7 | 58.2 | 41.8 | 74.9 | 67.4 | 48.2 | 74.2 | 61.4 |
| ADDA [15] | 45.2 | 68.8 | 79.2 | 64.6 | 60.0 | 68.3 | 57.6 | 38.9 | 77.5 | 70.3 | 45.2 | 78.3 | 62.8 |
| CDAN+E [37] | 47.5 | 65.9 | 75.7 | 57.1 | 54.1 | 63.4 | 59.6 | 44.3 | 72.4 | 66.0 | 49.9 | 72.8 | 60.7 |
| IWAN [23] | 53.9 | 54.5 | 78.1 | 61.3 | 48.0 | 63.3 | 54.2 | 52.0 | 81.3 | 76.5 | 56.8 | 82.9 | 63.6 |
| PADA [17] | 52.0 | 67.0 | 78.7 | 52.2 | 53.8 | 59.0 | 52.6 | 43.2 | 78.8 | 73.7 | 56.6 | 77.1 | 62.1 |
| ETN [26] | 59.2 | 77.0 | 79.5 | 62.9 | 65.7 | 75.0 | 68.3 | 55.4 | 84.4 | 75.7 | 57.7 | 84.5 | 70.5 |
| DRCN [27] | 54.0 | 76.4 | 83.0 | 62.1 | 64.5 | 71.0 | 70.8 | 49.8 | 80.5 | 77.5 | 59.1 | 79.9 | 69.0 |
| BA$^3$US [28] | 60.6 | 83.2 | 88.4 | 71.8 | 72.8 | 83.4 | 75.5 | 61.6 | 86.5 | 79.3 | 62.8 | 86.1 | 76.0 |
| ISRA+BA$^3$US [50] | 64.7 | 83.0 | 89.1 | 75.7 | 75.5 | 85.4 | 78.5 | 64.2 | 88.1 | 81.3 | 65.3 | 86.7 | 78.2 |
| SHOT++ [5] | 65.0 | 85.8 | **93.4** | **78.8** | 77.4 | 87.3 | 79.3 | 66.0 | 89.6 | 81.3 | 68.1 | 86.8 | 79.9 |
| SPDA [89] | 64.2 | **87.8** | 88.0 | 74.3 | 75.1 | 79.1 | 79.4 | 58.9 | 85.1 | 81.4 | 67.4 | 84.1 | 77.1 |
| APDA-CI [90] | 61.7 | 86.9 | 90.5 | 77.2 | 76.9 | 83.8 | 79.6 | 63.8 | 88.5 | 85.0 | 65.8 | 86.2 | 78.8 |
| CLA [91] | 66.7 | 85.6 | 90.9 | 75.6 | 76.9 | 86.8 | 78.8 | 67.4 | 88.7 | 81.7 | 66.9 | 87.8 | 79.5 |
| RAN [92] | 63.3 | 83.1 | 89.0 | 75.0 | 74.5 | 82.9 | 78.0 | 61.2 | 86.7 | 79.9 | 63.5 | 85.0 | 76.8 |
| STCPDA [93] | 63.1 | **87.8** | 90.1 | 77.2 | 75.4 | 85.6 | **81.4** | 62.4 | **90.5** | 82.6 | 69.5 | 88.2 | 79.5 |
| SLM [49] | 61.1 | 84.0 | 91.4 | 76.5 | 75.0 | 81.8 | 74.6 | 55.6 | 87.8 | 82.3 | 57.8 | 83.5 | 76.0 |
| SAN++ [94] | 61.3 | 81.6 | 88.6 | 72.8 | 76.4 | 81.9 | 74.5 | 57.7 | 87.2 | 79.7 | 63.8 | 86.1 | 76.0 |
| IDSP [47] | 60.8 | 80.8 | 87.3 | 69.3 | 76.0 | 80.2 | 74.7 | 59.2 | 85.3 | 77.8 | 61.3 | 85.7 | 74.9 |
| MOT [51] | 63.1 | 86.1 | 92.3 | 78.7 | **85.4** | **89.6** | 79.8 | 62.3 | 89.7 | 83.8 | 67.0 | **89.6** | 80.6 |
| *AR [33] | 67.4 | 85.3 | 90.0 | 77.3 | 70.6 | 85.2 | 79.0 | 64.8 | 89.5 | 80.4 | 66.2 | 86.4 | 78.3 |
| **ARPM** | **68.3** | **87.8** | 92.3 | 77.8 | 84.6 | 86.3 | 81.1 | **69.2** | 89.5 | **86.2** | **70.0** | 89.1 | **81.8** |

**Table 2** Accuracy (%) on ImageNet-Caltech dataset for PDA. The best results are bolded. *AR is our conference version.

| Method | C→I | I→C | Avg |
|---|---|---|---|
| ResNet-50 [1] | 71.3 | 69.7 | 70.5 |
| DAN [10] | 60.1 | 71.3 | 65.7 |
| DANN [38] | 67.7 | 70.8 | 69.2 |
| IWAN [23] | 73.3 | 78.1 | 75.7 |
| PADA [17] | 70.5 | 75.0 | 72.8 |
| ETN [26] | 74.9 | 83.2 | 79.1 |
| DRCN [27] | 78.9 | 75.3 | 77.1 |
| BA$^3$US [28] | 83.4 | 84.0 | 83.7 |
| ISRA+BA$^3$US [50] | 83.7 | 85.3 | 84.5 |
| SLM [49] | 81.4 | 82.3 | 81.9 |
| SAN++ [94] | 81.1 | 83.3 | 82.2 |
| ∗AR [33] | 82.2 | **87.1** | 84.7 |
| **ARPM** | **87.1** | 84.6 | **85.9** |

**Table 3** Accuracy (%) on VisDA-2017 dataset for PDA. The best results are bolded. *AR is our conference version.

| Method | S→R |
|---|---|
| ResNet-50 [1] | 45.3 |
| IWAN [23] | 48.6 |
| PADA [17] | 53.5 |
| DRCN [27] | 58.2 |
| SHOT++ [5] | 78.6 |
| SPDA [89] | 82.9 |
| APDA-CI [90] | 69.8 |
| RAN [92] | 75.1 |
| STCPDA [93] | 70.1 |
| SLM [49] | 91.7 |
| SAN++ [94] | 63.1 |
| MOT [51] | 92.4 |
| ∗AR [33] | 88.8 |
| **ARPM** | **93.2** |

sampled mini-batch data may contain only a few samples having non-zero weights.

## 5.2 Results for PDA

We implement our approach with three different random seeds $\{2019, 2021, 2023\}$, and report the

**Table 4** Accuracy (%) on Office-31 dataset for PDA. The best results are bolded. *AR is our conference version.

| Method | A→D | A→W | D→A | D→W | W→A | W→D | Avg |
|---|---|---|---|---|---|---|---|
| ResNet-50 [1] | 83.4 | 75.6 | 83.9 | 96.3 | 85.0 | 98.1 | 87.1 |
| DAN [10] | 61.8 | 59.3 | 75.0 | 73.9 | 67.6 | 90.5 | 71.3 |
| DANN [38] | 81.5 | 73.6 | 82.8 | 96.3 | 86.1 | 98.7 | 86.5 |
| IWAN [23] | 90.5 | 89.2 | 95.6 | 99.3 | 94.3 | 99.4 | 94.7 |
| PADA [17] | 82.2 | 86.5 | 92.7 | 99.3 | 95.4 | **100.0** | 92.7 |
| ETN [26] | 95.0 | 94.5 | 96.2 | **100.0** | 94.6 | **100.0** | 96.7 |
| DRCN [27] | 86.0 | 88.5 | 95.6 | **100.0** | 95.8 | **100.0** | 94.3 |
| TSCDA [29] | 98.1 | 96.8 | 94.8 | **100.0** | 96.0 | **100.0** | 97.6 |
| BA$^3$US [28] | 99.4 | 99.0 | 94.8 | **100.0** | 95.0 | 98.7 | 97.8 |
| ISRA+BA$^3$US [50] | 98.7 | 99.3 | 95.4 | **100.0** | 95.4 | **100.0** | 98.2 |
| SPDA [89] | 96.2 | 99.3 | 96.0 | **100.0** | 96.6 | **100.0** | 98.0 |
| APDA-CI [90] | 96.8 | 99.7 | 96.2 | **100.0** | 96.6 | **100.0** | 98.2 |
| CLA [91] | **100.0** | **100.0** | 94.5 | **100.0** | 96.7 | **100.0** | 98.5 |
| RAN [92] | 97.8 | 99.0 | 96.3 | **100.0** | 96.2 | **100.0** | 98.2 |
| SLM [49] | 98.7 | 99.8 | 96.1 | **100.0** | 95.9 | 99.8 | 98.4 |
| SAN++ [94] | 98.1 | 99.7 | 94.1 | **100.0** | 95.5 | **100.0** | 97.9 |
| IDSP [47] | 99.4 | 99.7 | 95.1 | 99.7 | 95.7 | **100.0** | 98.3 |
| MOT [51] | 98.7 | 99.3 | 96.1 | **100.0** | 96.4 | **100.0** | 98.4 |
| *AR [33] | 96.8 | 93.5 | 95.5 | **100.0** | 96.0 | 99.7 | 96.9 |
| **ARPM** | 99.6 | 99.4 | **96.6** | 99.9 | **96.8** | **100.0** | **98.7** |

**Table 5** Accuracy (%) on DomainNet dataset for PDA. The best results are bolded. *AR is our conference version. The results of the compared methods are produced using their official source codes.

| Method | C→P | C→R | C→S | P→C | P→R | P→S | R→C | R→P | R→S | S→C | S→P | S→R | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet-50 [1] | 41.2 | 60.0 | 42.1 | 54.5 | 70.8 | 48.3 | 63.1 | 58.6 | 50.3 | 45.4 | 39.3 | 49.8 | 52.0 |
| DANN [38] | 27.8 | 36.6 | 29.9 | 31.8 | 42.0 | 36.6 | 47.6 | 46.8 | 40.9 | 25.8 | 29.5 | 32.7 | 35.7 |
| CDAN+E [37] | 37.5 | 48.3 | 46.6 | 45.5 | 61.0 | 52.6 | 62.0 | 60.6 | 54.7 | 35.4 | 38.5 | 43.6 | 48.9 |
| PADA [17] | 22.5 | 32.9 | 30.0 | 25.7 | 56.5 | 30.5 | 65.3 | 63.4 | 54.2 | 17.5 | 23.9 | 26.9 | 37.4 |
| BA$^3$US [28] | 42.9 | 54.7 | 53.8 | 64.0 | 76.4 | 64.7 | 80.0 | 74.3 | 74.0 | 50.4 | 42.7 | 49.7 | 60.6 |
| ISRA+BA$^3$US [50] | 43.3 | 55.1 | 56.9 | 59.4 | 75.8 | 66.3 | 76.3 | 76.1 | 77.3 | 44.2 | 50.6 | 50.6 | 61.0 |
| STCPDA [93] | 65.1 | 69.6 | **69.6** | 72.7 | 77.6 | 78.7 | 78.1 | 72.9 | **80.0** | 64.4 | 60.7 | 67.8 | 71.4 |
| *AR [33] | 52.7 | 68.2 | 58.3 | 66.8 | 77.5 | 74.4 | 76.7 | 71.8 | 70.5 | 53.7 | 53.6 | 61.6 | 65.5 |
| **ARPM** | **67.9** | **79.8** | 66.3 | **78.4** | **84.1** | **81.9** | **86.5** | **78.0** | 78.6 | **62.5** | **64.8** | **71.7** | **75.0** |

average results over these three different runs in Tables 1, 4, 5, 2, and 3 for Office-Home, Office-31, DomainNet, ImageNet-Caltech, and VisDA-2017 datasets, respectively. The results of compared methods on Office-Home, Office-31, ImageNet-Caltech, and VisDA-2017 datasets are quoted from their papers. The results of compared methods on DomainNet dataset are produced using their official codes (except the results of STCPDA [93], which are quoted from its paper).

In Table 1, our approach ARPM achieves the best average accuracy of 81.8% on Office-Home dataset, outperforming the second-best approach of MOT by 1.2%. Tables 2 and 3 imply that our proposed ARPM achieves the best results of 85.9% and 93.2% on ImageNet-Caltech and ViSDA-2017, respectively. ARPM outperforms the second-best method on ImageNet-Caltech by 1.2% and on VisDA-2017 by 0.8%, respectively. In Table 4, on Office-31 dataset, our proposed ARPM achieves the best result of 98.7%, outperforming the recent

**Table 6** H-score (%) on Office-Home of different open-set DA methods. The best results are bolded.

| Method | A→C | A→P | A→R | C→A | C→P | C→R | P→A | P→C | P→R | R→A | R→C | R→P | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STA [52] | 55.8 | 54.0 | 68.3 | 57.4 | 60.4 | 66.8 | 61.9 | 53.2 | 69.5 | 67.1 | 54.5 | 64.5 | 61.1 |
| OSBP [19] | 55.1 | 65.2 | 72.9 | 64.3 | 64.7 | 70.6 | 63.2 | 53.2 | 73.9 | 66.7 | 54.5 | 72.3 | 64.7 |
| ROS [55] | 60.1 | 69.3 | 76.5 | 58.9 | 65.2 | 68.6 | 60.6 | 56.3 | 74.4 | 68.8 | 60.4 | 75.7 | 66.2 |
| DCC [64] | 56.1 | 67.5 | 66.7 | 49.6 | 66.5 | 64.0 | 55.8 | 53.0 | 70.5 | 61.6 | 57.2 | 71.9 | 61.7 |
| OVANet [63] | 58.6 | 66.3 | 69.9 | 62.0 | 65.2 | 68.6 | 59.8 | 53.4 | 69.3 | 68.7 | 59.6 | 66.7 | 64.0 |
| UMAD [97] | 59.2 | 71.8 | 76.6 | 63.5 | 69.0 | 71.9 | 62.5 | 54.6 | 72.8 | 66.5 | 57.9 | 70.7 | 66.4 |
| GATE [65] | 63.8 | 70.5 | 75.8 | 66.4 | 67.9 | 71.7 | 67.3 | 61.5 | 76.0 | 70.4 | 61.8 | 75.1 | 69.1 |
| GLC [98] | **65.3** | 74.2 | 79.0 | 60.4 | 71.6 | **74.7** | 63.7 | **63.2** | 75.8 | 67.1 | 64.3 | 77.8 | 69.8 |
| PPOT [68] | 60.7 | 75.2 | 79.5 | **67.3** | 70.1 | 73.8 | **70.6** | 57.2 | 76.1 | 71.8 | 61.4 | 75.8 | 70.0 |
| ANNA [61] | 69.9 | 73.7 | 76.8 | 64.7 | 68.6 | 73.0 | 66.5 | 63.1 | 76.6 | 71.6 | **65.7** | 78.7 | 70.7 |
| **ARPM** | 63.8 | **76.0** | **80.6** | **67.3** | **72.1** | 74.6 | 67.7 | 61.7 | **76.7** | **73.8** | 65.4 | **81.2** | **71.7** |

PDA methods CLA [91] (the second-best method) by 0.2%. ARPM outperforms CLA by 2.3% on Office-Home dataset. Table 5 shows that our approach ARPM achieves the best accuracy on DomainNet dataset, outperforming the second-best method by 3.6%.

Among the reported methods in Tables 1, 2, 3, 4, and 5, DANN [13], DAN [10], ADDA [15], and CDAN+E [37] are devised for closed-set DA which do not consider the challenge of label space mismatch in PDA and achieve worse results than the other PDA approaches. In PDA approaches, IWAN [23], PADA [17], ETN [26], DRCN [27], TSCDA [29], and BA$^3$US [28] align reweighted source domain feature distribution and unweighted target domain feature distribution by minimizing maximum mean discrepancy or adversarial training, performing worse than MOT [51]. MOT [51] aligns reweighted source and unweighted target feature distributions using robust optimal transport that does not require exact matching of the distributions. This implies that aligning reweighted distribution without requiring exact matching may be better than that with requiring exact matching for PDA. Our proposed ARPM boosts the performance of MOT on all the evaluated datasets. ARPM adversarially learns to reweight source domain data by minimizing the cross-domain distribution distance, and enforcing robustness and reducing prediction uncertainty of the recognition model, which more effectively tackles PDA from a novel perspective with theoretical analysis.

Compared with the conference version, the results on Office, Office-Home, ImageNet-Caltech, VisDA-2017, and DomainNet datasets are improved by 1.8%, 3.5%, 1.2%, 4.4%, and 9.5% in this journal version, respectively. In this journal version, we extend the work by introducing more methodological techniques, *e.g.*, the $\alpha$-power maximization for substituting entropy minimization, the neighborhood reciprocity clustering [32], and the spectral normalization to enforce the Lipschitz constraint and the PCA-based initialization strategy to improve stability. The performance improvements demonstrate the effectiveness of the methodological extensions for PDA. Note that the usefulness of each of these techniques will be verified in Sect. 5.5.

## 5.3 Results for Open-set and Universal DA

We follow the protocol of [68] to conduct experiments on Office-Home dataset for open-set and universal DA. In the test phase, the samples with prediction confidence lower than 0.65 are classified as "unknown" class. $\tau$ is set to 0.25 and $\lambda'$ is set to 0.05. The other experimental details are the same as those for PDA. We report the open-set evaluation metric, H-score, in Tables 6 and 7. It can be observed from Tables 6 and 7 that our method performs better than the recent open-set DA methods and universal DA methods on Office-Home dataset. Our proposed ARPM outperforms the previous state-of-the-art method PPOT [68] by 1.7% in open-set DA task as in Table 6 and by 0.2% in universal DA task. These results imply

**Table 7** H-score (%) on Office-Home of different universal DA methods. The best results are bolded.

| Method | A→C | A→P | A→R | C→A | C→P | C→R | P→A | P→C | P→R | R→A | R→C | R→P | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UAN [20] | 51.6 | 51.7 | 54.3 | 61.7 | 57.6 | 61.9 | 50.4 | 47.6 | 61.5 | 62.9 | 52.6 | 65.2 | 56.6 |
| CMU [21] | 56.0 | 56.9 | 59.2 | 67.0 | 64.3 | 67.8 | 54.7 | 51.1 | 66.4 | 68.2 | 57.9 | 69.7 | 61.6 |
| DANCE [62] | 61.0 | 60.4 | 64.9 | 65.7 | 58.8 | 61.8 | 73.1 | 61.2 | 66.6 | 67.7 | 62.4 | 63.7 | 63.9 |
| USFDA [99] | 60.3 | 79.7 | 80.7 | 64.2 | 67.9 | 79.3 | 74.1 | 65.3 | 80.1 | 68.6 | 59.8 | 77.9 | 71.4 |
| UMAD [97] | 61.1 | 76.3 | 82.7 | 70.7 | 67.7 | 75.7 | 64.4 | 55.7 | 76.3 | 73.2 | 60.4 | 77.2 | 70.1 |
| DCC [64] | 58.0 | 54.1 | 58.0 | 74.6 | 70.6 | 77.5 | 64.3 | 73.6 | 74.9 | 81.0 | **75.1** | 80.4 | 70.2 |
| OVANet [63] | 63.4 | 77.8 | 79.7 | 69.5 | 70.6 | 76.4 | 73.5 | 61.4 | 80.6 | 76.5 | 64.3 | 78.9 | 72.7 |
| GATE [65] | 63.8 | 75.9 | 81.4 | 74.0 | 72.1 | 79.8 | 74.7 | **70.3** | 82.7 | 79.1 | 71.5 | 81.7 | 75.6 |
| UniOT [100] | **67.3** | 80.5 | 86.0 | 73.5 | 77.3 | 84.3 | 75.5 | 63.3 | 86.0 | 77.8 | 65.4 | 81.9 | 76.6 |
| GLC [98] | 64.3 | 78.2 | **89.6** | 63.1 | **81.7** | **89.1** | 77.6 | 54.2 | **88.9** | 80.7 | 54.2 | **85.9** | 75.6 |
| PPOT [68] | 66.0 | 79.3 | 84.8 | **78.8** | 78.0 | 80.4 | **82.0** | 62.0 | 86.0 | **82.3** | 65.0 | 80.8 | 77.1 |
| SAKA [69] | 64.3 | 80.4 | 86.1 | 72.0 | 71.1 | 77.8 | 71.5 | 61.7 | 83.8 | 79.1 | 64.8 | 82.4 | 74.6 |
| **ARPM** | 65.2 | **81.2** | 89.4 | 73.2 | 73.4 | 83.9 | 74.9 | 67.3 | 84.8 | 78.9 | 70.3 | 85.1 | **77.3** |

**Table 8** Accuracy (%) on ImageNet-R of different TTA methods. The best result is bolded.

| Method | Source-trained model | TTT [101] | NORM [102] | TENT [22] | DUA [103] | **TPM (ours)** |
|---|---|---|---|---|---|---|
| Accuracy | 33.0 | 33.5 | 34.7 | 36.8 | 33.2 | **38.3** |

**Table 9** Ablation study on VisDA-2017 dataset. "SO" means training the model using the source domain data only. "R" represents our adversarial reweighting model. "P" is the $\alpha$-power loss. "N" is the neighborhood reciprocity clustering loss. "SO+R+N+P" is our full method ARPM. We also report the results for entropy minimization (E) as a substitute of $\alpha$-power loss (P).

| Method | S→R |
|---|---|
| SO | 46.4 |
| SO+R | 50.4 |
| SO+P | 90.7 |
| SO+N | 89.2 |
| SO+R+P | 91.7 |
| SO+R+N | 90.8 |
| SO+N+P | 92.4 |
| SO+R+N+P (ARPM) | **93.2** |
| SO+E | 85.2 |
| SO+R+E | 89.2 |
| SO+E+R+N | 90.7 |

that our method can tackle the tasks of open-set DA and universal DA with open-class data.

## 5.4 Results for TTA

For TTA, we take the RestNet-18 pre-trained on ImageNet as the source-trained model and evaluate it on ImageNet-R dataset [104]. The experimental setups are the same as those in [22]. The experimental results are reported in Table 8. We can see that our proposed method TPM (test $\alpha$-power maximization) achieves better results than the other TTA approaches. Especially, TPM outperforms TENT. The model adaptation is performed by $\alpha$-power maximization in TPM and by entropy minimization in TENT. The performance improvement of TPM over TENT indicates that our $\alpha$-power maximization could be more effective than entropy minimization for TTA.

## 5.5 Analysis

In this section, for convenience of description, we utilize "SO" to denote the baseline approach that trains the recognition model using the source domain data only. "R" represents our adversarial reweighting model. "P" is the $\alpha$-power loss. "N" is the neighborhood reciprocity clustering loss.

**Effectiveness of components in ARPM.** We study the effectiveness of each component in our

**Table 10** Ablation study on Office-Home dataset. "SO" means training the model using the source domain data only. "R" represents our adversarial reweighting model. "P" is the $\alpha$-power loss. "N" is the neighborhood reciprocity clustering loss. "SO+R+N+P" is our full method ARPM. We also report the results for entropy minimization (E) as a substitute of $\alpha$-power loss (P).

| Method | A→C | A→P | A→R | C→A | C→P | C→R | P→A | P→C | P→R | R→A | R→C | R→P | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SO | 50.2 | 69.5 | 79.8 | 60.2 | 61.1 | 67.5 | 60.3 | 43.7 | 74.5 | 71.3 | 50.6 | 78.0 | 63.9 |
| SO+R | 50.3 | 71.4 | 83.3 | 61.9 | 64.3 | 72.6 | 63.8 | 43.9 | 78.5 | 73.3 | 52.6 | 80.5 | 66.4 |
| SO+P | 64.3 | 84.4 | 88.8 | 76.6 | 80.0 | 81.9 | 78.2 | 65.4 | 88.4 | 81.0 | 63.1 | 86.3 | 78.2 |
| SO+N | 63.1 | 83.8 | 88.3 | 75.1 | 76.6 | 77.4 | 77.4 | 60.8 | 85.1 | 81.3 | 62.9 | 86.1 | 76.5 |
| SO+R+P | 67.5 | 87.6 | 91.7 | 79.4 | 78.9 | 85.0 | 78.9 | 65.3 | **90.8** | 82.8 | 63.7 | 87.7 | 79.9 |
| SO+R+N | 65.3 | 86.1 | 92.0 | **80.3** | 77.2 | 85.4 | **81.2** | 64.2 | 87.8 | 84.5 | 65.3 | 89.0 | 79.9 |
| SO+N+P | 65.6 | 86.5 | 91.3 | 78.6 | 82.1 | 82.3 | 79.8 | 65.6 | 90.7 | 82.2 | 65.0 | 86.9 | 79.7 |
| SO+R+N+P (ARPM) | **68.3** | **87.8** | **92.3** | 77.8 | **84.6** | **86.3** | 81.1 | **69.2** | 89.5 | **86.2** | **70.0** | **89.1** | **81.8** |
| SO+E | 58.6 | 84.1 | 88.0 | 74.1 | 76.8 | 77.4 | 76.1 | 61.4 | 86.2 | 78.5 | 61.5 | 84.3 | 75.6 |
| SO+R+E | 63.3 | 86.7 | 91.5 | 78.0 | 77.6 | 81.9 | 77.5 | 64.6 | 89.7 | 81.0 | 63.6 | 86.8 | 78.4 |
| SO+E+R+N | 66.6 | 86.8 | 91.9 | 78.3 | 81.9 | 85.3 | 77.9 | 65.9 | 89.6 | 83.7 | 67.9 | 89.6 | 80.5 |

method on VisDA-2017 and Office-Home datasets, of which the results are reported in Tables 9 and 10. Note that the differences between SO and "ResNet-50" in Tables 1 and 9 are that SO utilizes label smoothing and feature normalization. These two techniques enable SO to perform slightly better than ResNet-50. We can observe from Tables 9 and 10 that on both datasets, SO+R, SO+R+P, SO+R+N, and SO+R+N+P (*i.e.*, our full method ARPM) respectively outperforms SO, SO+P, SO+N, and SP+N+P, demonstrating the effectiveness of our adversarial reweighting model for learning to reweight source domain data. The results of SO+P, SO+R+P, SO+N+P, SO+R+N+P are better than those of SO, SO+R, SO+N, SO+R+N, respectively. This confirms the usefulness of our $\alpha$-power maximization for reducing the prediction uncertainty of the recognition model. Tables 9 and 10 show that SO+N+P outperforms both SO+N and SO+P, which implies that $\alpha$-power maximization and neighborhood reciprocity clustering are complementary to improve the performance. On all the combinations of SO, P, N, and R, SO+R+P+N, *i.e.*, ARPM, achieves the best results on both datasets. Note that SO and SO+R do not utilize target domain data to update the recognition model (SO+R utilizes target domain data to learn the source data weight). The results of SO and SO+R seem to be largely lower than those of the other approaches in Tables 9 and 10 since the $\alpha$-power and neighborhood reciprocity clustering losses are implemented on target domain data.

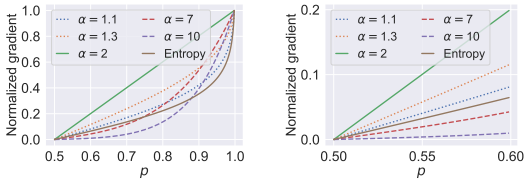
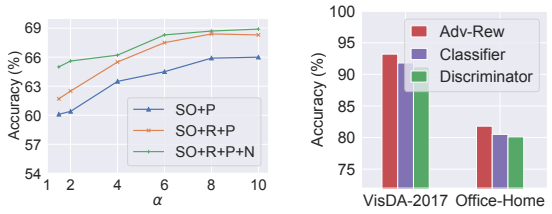



**Fig. 6** (a) The normalized gradient of $\alpha$-power loss with varying $\alpha$ and entropy loss. (b) Amplified part of Fig. 6(a) with $p$ ranging in $[0.5, 0.6]$.

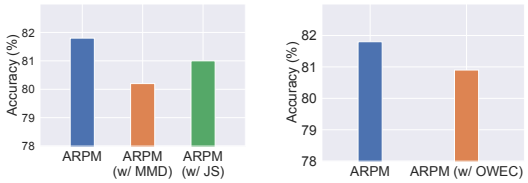
**Comparison of $\alpha$-power maximization and entropy minimization.** We report the results of entropy minimization (E) as a substitute of $\alpha$-power maximization (P) in our framework. Tables 9 and 10 show that SO+E, SO+R+E, and SO+R+E+N respectively degrade the results of SO+P, SO+R+P, and SO+R+P+N by more than 1.0% on both VisDA-2017 and Office-Home datasets. This demonstrates that our $\alpha$-power maximization is more effective than entropy minimization for PDA.

We provide more analysis on the $\alpha$-power maximization and entropy minimization in this paragraph. We take the two-way classification task as an example to compare the gradients of the $\alpha$-power loss (*i.e.*, $\mathcal{H}(p) = p^\alpha + (1-p)^\alpha$) and entropy loss (*i.e.*, $\mathcal{H}(p) = p\log p + (1-p)\log(1-p)$) *w.r.t.* the probability $p$ in Fig. 6. Note we only




(a) (b)

**Fig. 7** (a) Results with varying $\alpha$ in task A→C on Office-Home dataset. (b) Results of different reweighting strategies on Office-Home and VisDA-2017 datasets.




(a) (b)

**Fig. 8** (a) Results for MMD and JS-divergence for learning source data weights in our framework on Office-Home dataset. (b) Results for learning one weight for each class (OWEC) on Office-Home dataset.

care about the absolute value of the gradients in this experiment. The gradients are normalized by dividing by their maximum value when $p$ ranges in $[0.5, 0.99]$. The samples with larger normalized gradients are more important in training. We can see (in Fig. 6(a)) that the curves of $\alpha$-power losses approach that of entropy as $\alpha$ goes to 1. Larger $\alpha$ more possibly pushes the gradients of uncertain samples (with $p$ near 0.5) to zero as shown in Fig. 6(b) (also in Fig. 5 for three-way classification), and hence neglects their importance in network training. In Fig. 7(a), we show that in range $[1, 10]$, $\alpha$ larger than 2 achieves better performance than $\alpha$ smaller than 2. This may be because, for $\alpha$ (in $[1, 10]$) larger than 2, the uncertain samples that are more likely incorrectly predicted are less important (compared with $\alpha$ smaller than 2) when reducing the prediction uncertainty.

**Comparison of different reweighting strategies.** We compare different reweighting strategies for obtaining the weights used in our loss of Eq. (4) for PDA, including our adversarial reweighting (Adv-Rew), reweighting based on the classifier in the PDA methods [16, 17, 27, 28], and reweighting by the output of discriminator on source data as in [23]. For the classifier-based strategy, the source data weight of the $k$-th class is defined by $\frac{1}{n}\sum_{j=1}^{n} C(F(\mathbf{x}_j^t))_k$. For the discriminator-based strategy, we introduce a discriminator $\tilde{D}$ that aims to predict 1 (*resp.* 0) on the target (*resp.* source) domain data. The weight of source domain data $\mathbf{x}_i^s$ is $\tilde{D}(\mathbf{x}_i^s)$. The results in Fig. 7(b) show that our adversarial reweighting outperforms the other two reweighting strategies on VisDA-2017 and Office-Home datasets, confirming the effectiveness of our adversarial reweighting strategy.

**Comparison with MMD and JS-divergence to learn the weights.** As discussed in Sect. 3.2, we minimize the Wasserstein distance to learn the data weights deducing our adversarial reweighting model. We compare ARPM with the approaches that minimize the JS-divergence and Maximum Mean Discrepancy (MMD) to learn the data weights in our framework (denoted as ARPM (w/ JS) and ARPM (w/ MMD), respectively), on Office-Home dataset for PDA. Note that the JS-divergence also induced an adversarial reweighting model where the discriminator is trained with loss in [105]. In Fig. 8(a), we can see that ARPM using the Wasserstein distance for learning the data weights outperforms ARPM (w/ JS) and ARPM (w/ MMD). The reasons could be as follows. When the supports of source and target distributions are disjoint, the Wasserstein distance may be more suitable to measure their distance than JS-divergence [72]. The MMD with widely used kernels may be unable to capture very complex distances in high dimensional spaces [72, 106], possibly making it less effective than the Wasserstein distance in our framework.

**Comparison with learning one weight for each class (OWEC).** The previous PDA methods [16, 17, 27, 28] assign one weight for each class. As comparisons, we conduct experiments for learning one weight shared by samples of each class in our adversarial reweighting model (denoted as ARPM (w/ OWEC)) on Office-Home dataset for PDA. The results in Fig. 8(b) show that ARPM with individual weight for each sample outperforms ARPM (w/ OWEC), implying that learning individual weight for each sample is more effective. If the weight is learned for each sample, it is possible to assign higher weights to

**Table 11** Results (under varying random seeds) of gradient penalty (GP) and spectral normalization (SP) to impose Lipschitz constraint on Office-Home dataset.

| Seed | 2019 | 2021 | 2023 | Avg | Std |
|---|---|---|---|---|---|
| SP | 81.8 | 81.6 | 82.1 | 81.8 | 0.3 |
| GP | 81.1 | 80.6 | 81.9 | 81.2 | 0.7 |

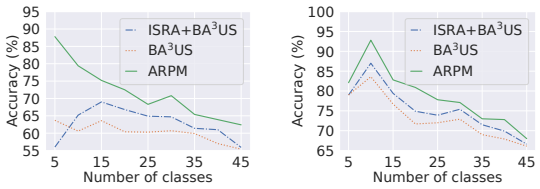
**Table 12** Results (under varying random seeds) of ARPM with and without PCA for initializing classifier on VisDA-2017 dataset.

| Seed | 2015 | 2017 | 2019 | 2021 | 2023 |
|---|---|---|---|---|---|
| w/ PCA | 92.4 | 93.5 | 92.2 | 93.9 | 93.6 |
| w/o PCA | 93.8 | 76.8 | 94.1 | 78.6 | 93.9 |

samples (even in the same class) closer to the target domain. The model trained in such a case may be more transferable, because samples (even in the source domain common classes) less relevant to the target domain become less important.

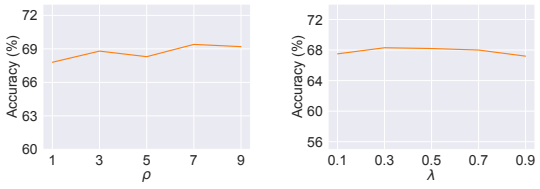
**Comparison of gradient penalty (GP) and spectral normalization (SP) to impose Lipschitz constraint.** We compare the gradient penalty (GP) and spectral normalization (SP) to impose Lipschitz constraint in Eq. (3). The results in Table 11 indicate that spectral normalization achieves better average accuracy and lower standard variation. As discussed in Sect. 3.2, GP introduces additional hyper-parameter and randomness, which may degrade the reproducibility.

**PCA initialization of classifier improving reproducibility.** Table 12 shows that initializing the weight of the classifier by PCA as discussed in Sect. 3.5 does improve the reproducibility of our method (*i.e.*, our method with PCA performs well over different random seeds). Note that $\nabla_{\mathbf{p}}\mathcal{H}_\alpha(\mathbf{p}) \propto (p_1^{\alpha-1}, p_2^{\alpha-1}, \cdots, p_{|\mathcal{Y}|}^{\alpha-1})$, which implies that $\mathbf{p}$ is updated with high possibility towards the class corresponding to the largest element of $\mathbf{p}$. Therefore, good initialization with "correct" $\nabla_{\mathbf{p}}\mathcal{H}_\alpha(\mathbf{p})$ on target domain may yield better results. Our PCA-based initialization specifies the variance preservation idea of commonly used random initialization strategies [79, 80] for normalized features and reduces the randomness, which may yield better reproducibility.



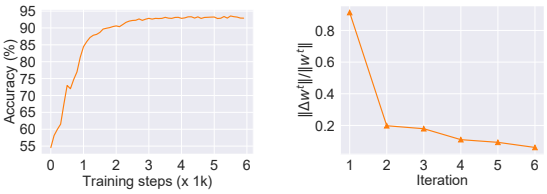

(a) (b)

**Fig. 9** Accuracy with varying numbers of target classes in tasks (a) A→C and (b) C→A on Office-Home dataset.




(a) (b)

**Fig. 10** Results for varying magnitudes of (a) $\rho$ in the constraints of adversarial reweighting model and (b) $\lambda$ in Eq. (9), in task A→C on Office-Home dataset.

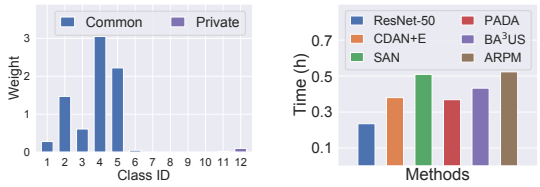



(a) (b)

**Fig. 11** (a) Accuracy in training in task S→R on VisDA-2017 dataset. (b) Relative difference of source data weights in alternate iteration in task S→R on VisDA-2017 dataset.

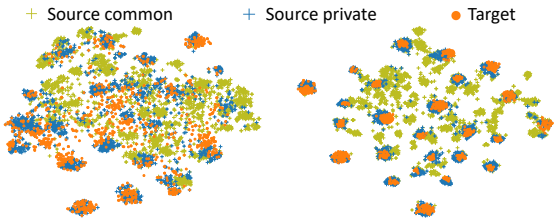
**Accuracy with varying numbers of target classes.** We evaluate our method with different numbers of target classes in Fig. 9. Our method of ARPM outperforms recent PDA methods BA$^3$US [28] and ISRA+BA$^3$US [50] when the number of target classes is smaller than 45 (the number of source classes is 65). This indicates that our method is effective for PDA with different degrees of label space mismatch.

**Sensitivity to hyper-parameters.** We investigate the effect of hyper-parameters $\lambda$ and $\rho$ in Fig. 10. Our method is relatively stable to $\rho$ in range [3, 9] as shown in Fig. 10(a) and to $\lambda$ in [0.3, 0.7] as shown in Fig. 10(b).




(a) (b)

**Fig. 12** (a) Average weights for each class on source domain in task S→R on VisDA-2017. (b) Computational cost of PDA methods in task A→C on Office-Home.




(a) Before adaptation (b) After adaptation

**Fig. 13** T-SNE visualization of features in task R→A on Office-Home dataset. (a) Features before adaptation. (b) Features after adaptation of ARPM.

**Convergence of training algorithm.** In Fig. 11, we take the PDA task S→R on VisDA-2017 as an example to study the convergence of our method. Figure 11(a) indicates that the accuracy of our approach stably increases and converges in the training process. We also show the relative difference of weights in Fig. 11(b). The relative difference is $\frac{\|\Delta \mathbf{w}^t\|}{\|\mathbf{w}^t\|}$, where $\Delta \mathbf{w}^t = \mathbf{w}^{t+1} - \mathbf{w}^t$, and $\mathbf{w}^t$ is the value of the weights in the $t$-th iteration of the alternate training algorithm. We can see that $\frac{\|\Delta \mathbf{w}^t\|}{\|\mathbf{w}^t\|}$ stably decreases.

**Visualization of learned weights.** We visualize the learned average weights of source domain data of each class in task S→R on VisDA-2017 dataset, as shown in Fig. 12(a). We can see that the source domain common class (the first six classes) data are assigned with larger weights in general (except the 6-th class). Even for the 6-th class, its weight is larger than the weights of five among the total of six source-private classes.

**Computational cost.** We compare the computational cost of different methods with the total training time in the same training steps (5000 steps), as in Fig. 12(b). Figure 12(b) shows that our approach (ARPM) is comparable to other methods in terms of computational time cost. Note that the test time cost is similar for all the methods because all the methods only need one forward process.

**Feature visualization.** We visualize the features before and after the adaptation of ARPM in Fig. 13 by T-SNE [107]. We can see that from Fig. 13(b), the source domain common class features are more discriminative/separable than the source-private class features, and the target domain data features are aligned with the source domain common class features.

# 6 Conclusion

In this paper, we propose a novel ARPM approach for PDA, in which we propose an adversarial reweighting model to learn to reweight source domain data, propose $\alpha$-power maximization to reduce prediction uncertainty, and utilize the neighborhood reciprocity clustering to enforce robustness. Extensive experiments on five benchmark datasets demonstrate the effectiveness of the proposed ARPM for PDA. We also present the theoretical analysis of the proposed method. Additionally, we extend our approach to more "open-world" recognition tasks, including open-set DA, universal DA, and TTA. Since both the adversarial reweighting model and the $\alpha$-power maximization in our approach require accessing the target domain data, it is non-trivial to extend our approach to the adaptation tasks without target domain data, *e.g.*, domain generalization. We will explore more applications of our approach in the future.

# 7 Acknowledgments

The work was supported by National Key R&D Program 2021YFA1003002, NSFC (12125104, U20B2075, 12326615), Postdoctoral Fellowship Program of CPSF GZB20230582, and Key Laboratory of Biomedical Imaging Science and System, Chinese Academy of Sciences.

# 8 Data Availability Statement

The data that support the findings of this study are available from the authors upon request.

# 9 Statements and Declarations

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.

# Appendix

We first give some lemmas and then provide the proof of Theorem 1.

**Lemma 1.** *Divide $\mathcal{Y}_{\text{com}}$ into $S_1$ and $S_2$ such that $S_1 = \{i \in \mathcal{Y}_{\text{com}} : \mathbb{E}_{(\mathbf{x},y)\sim\frac{P_i^{\text{c}}+Q_i}{2}}\mathbb{I}(\exists \mathbf{x}' \in \mathcal{N}(\mathbf{x}), \tilde{f}(\mathbf{x}) \neq \tilde{f}(\mathbf{x}')) < \min\{\epsilon, q\}\}$ and $S_2 = \{i \in \mathcal{Y}_{\text{com}} : \mathbb{E}_{(\mathbf{x},y)\sim\frac{P_i^{\text{c}}+Q_i}{2}}\mathbb{I}(\exists \mathbf{x}' \in \mathcal{N}(\mathbf{x}), \tilde{f}(\mathbf{x}) \neq \tilde{f}(\mathbf{x}')) \geq \min\{\epsilon, q\}\}$. Under the condition of Theorem 1, we have*

$$\sum_{i\in S_2}\frac{P^{\text{c}}+Q}{2}(y=i) \leq \frac{R_{\frac{P^{\text{c}}+Q}{2}}(f)}{\min\{\epsilon,q\}}. \tag{16}$$

*Proof.* Suppose $\sum_{i\in S_2}\frac{P^{\text{c}}+Q}{2}(y=i) > \frac{R(f)}{\min\{\epsilon,q\}}$, which implies

$$\begin{aligned}
&R_{\frac{P^{\text{c}}+Q}{2}}(f)\\
=&\frac{P^{\text{c}}+Q}{2}(\{\mathbf{x}, \exists \mathbf{x}' \in \mathcal{N}(\mathbf{x}), \tilde{f}(\mathbf{x}) \neq \tilde{f}(\mathbf{x}')\})\\
=&\mathbb{E}_{(\mathbf{x},y)\sim\frac{P^{\text{c}}+Q}{2}}\mathbb{I}(\exists \mathbf{x}' \in \mathcal{N}(\mathbf{x}), \tilde{f}(\mathbf{x}) \neq \tilde{f}(\mathbf{x}'))\\
=&\sum_{i\in\mathcal{Y}_{\text{com}}}\Big\{\mathbb{E}_{\mathbf{x}\sim\frac{P_i^{\text{c}}+Q_i}{2}}\mathbb{I}(\exists \mathbf{x}' \in \mathcal{N}(\mathbf{x}), \tilde{f}(\mathbf{x}) \neq \\
&\qquad \tilde{f}(\mathbf{x}'))\frac{P^{\text{c}}+Q}{2}(y=i)\Big\}\\
\geq&\sum_{i\in S_2}\Big\{\mathbb{E}_{\mathbf{x}\sim\frac{P_i^{\text{c}}+Q_i}{2}}\mathbb{I}(\exists \mathbf{x}' \in \mathcal{N}(\mathbf{x}), \tilde{f}(\mathbf{x}) \neq \\
&\qquad \tilde{f}(\mathbf{x}'))\frac{P^{\text{c}}+Q}{2}(y=i)\Big\}\\
\geq&\min\{\epsilon,q\}\sum_{i\in S_2}\frac{P^{\text{c}}+Q}{2}(y=i) > R_{\frac{P^{\text{c}}+Q}{2}}(f).
\end{aligned} \tag{17}$$

$R_{\frac{P^{\text{c}}+Q}{2}}(f) > R_{\frac{P^{\text{c}}+Q}{2}}(f)$ forms a contradiction. □

**Lemma 2** (Lemma 2 in [6])**.** *Under the condition of Theorem 1, if sub-populations $P_i^{\text{c}}$ and $Q_i$ satisfy $\mathbb{E}_{(\mathbf{x},y)\sim\frac{P_i^{\text{c}}+Q_i}{2}}\mathbb{I}(\exists \mathbf{x}' \in \mathcal{N}(\mathbf{x}), \tilde{f}(\mathbf{x}) \neq \tilde{f}(\mathbf{x}')) < \min\{\epsilon, q\}$, we have*

$$|\varepsilon_{P_i^{\text{c}}}(f) - \varepsilon_{Q_i}(f)| \leq 2q. \tag{18}$$

**Lemma 3** (Lemma 3 in [6])**.** *For any distribution $P$, if $f$ is $L$-Lipschiz w.r.t. $d(\cdot,\cdot)$, we have*

$$R_P(f) \leq \frac{1}{(1-2L\xi)}(1 - M_P(f)). \tag{19}$$

**Proof of Theorem 1.** From the definition of $\varepsilon_Q(f)$ in PDA, we have

$$\begin{aligned}
\varepsilon_Q(f) =& \sum_{i\in\mathcal{Y}_{\text{com}}}\varepsilon_{Q_i}(f)Q(y=i)\\
\leq& \sum_{i\in S_1}\varepsilon_{Q_i}(f)Q(y=i) + \sum_{i\in S_2}Q(y=i)\\
\leq& \sum_{i\in S_1}(\varepsilon_{P_i}(f)+2q)rP^{\text{c}}(y=i)\\
&+\sum_{i\in S_2}Q(y=i)\\
\leq& \sum_{i\in\mathcal{Y}_{\text{com}}}(\varepsilon_{P_i}(f)+2q)rP^{\text{c}}(y=i)\\
&+\sum_{i\in S_2}Q(y=i)\\
=&r\varepsilon_{P^{\text{c}}}(f) + 2qr + \sum_{i\in S_2}Q(y=i).
\end{aligned} \tag{20}$$

The second inequality uses Lemma 2 and $\frac{Q(y=i)}{P^{\text{c}}(y=i)} \leq r$ for $i \in \mathcal{Y}_{\text{com}}$. Since

$$\begin{aligned}
\frac{P^{\text{c}}+Q}{2}(y=i) =& \frac{1}{2}(P^{\text{c}}(y=i) + Q(y=i))\\
\geq& \frac{1}{2}(\frac{1}{r}Q(y=i) + Q(y=i))\\
=& \frac{1+r}{2r}Q(y=i),
\end{aligned} \tag{21}$$

we have

$$\sum_{i\in S_2}Q(y=i) \leq \frac{2r}{1+r}\sum_{i\in S_2}\frac{P^{\text{c}}+Q}{2}(y=i). \tag{22}$$

Using Lemma 1, we have

$$\sum_{i\in S_2}Q(y=i) \leq \frac{2r}{\min\{\epsilon,q\}(1+r)}R_{\frac{P^{\text{c}}+Q}{2}}(f). \tag{23}$$

Applying Lemma 3, for any $\eta \in [0,1]$, we have

$$\begin{aligned}&\sum_{i\in S_2} Q(y=i)\\ \leq&\frac{2r\eta}{\min\{\epsilon,q\}(1+r)} R_{\frac{P^c+Q}{2}}(f)\\ +&\frac{2r(1-\eta)}{\min\{\epsilon,q\}(1+r)(1-2L\xi)}(1-\mathcal{M}_{\frac{P^c+Q}{2}}(f)).\end{aligned} \tag{24}$$

Combining Eqs. (20) and (24), we have

$$\begin{aligned}\varepsilon_Q(f) \leq& r\varepsilon_{P^c}(f) + c_1 R_{\frac{P^c+Q}{2}}(f)\\ &+ c_2(1 - M_{\frac{P^c+Q}{2}}(f)) + 2rq,\end{aligned} \tag{25}$$

where the coeffcients $c_1 = \frac{2\eta r}{\min\{\epsilon,q\}(1+r)}$ and $c_2 = \frac{2r(1-\eta)}{\min\{\epsilon,q\}(1-2L\xi)(1+r)}$ are constants to $f$. □

# References


[1] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)

[2] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS (2012)

[3] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)

[4] Gu, X., Sun, J., Xu, Z.: Unsupervised and semi-supervised robust spherical space domain adaptation. IEEE Trans. PAMI (In press, 2022)

[5] Liang, J., Hu, D., Wang, Y., He, R., Feng, J.: Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. IEEE Trans. PAMI (In press, 2021)

[6] Liu, H., Wang, J., Long, M.: Cycle self-training for domain adaptation. In: NeurIPS (2021)

[7] Xu, T., Chen, W., WANG, P., Wang, F., Li, H., Jin, R.: CDTrans: Cross-domain transformer for unsupervised domain adaptation. In: ICLR (2022)

[8] Kang, G., Jiang, L., Wei, Y., Yang, Y., Hauptmann, A.: Contrastive adaptation network for single- and multi-source domain adaptation. IEEE Trans. PAMI **44**(4), 1793–1804 (2022)

[9] Koniusz, P., Tas, Y., Porikli, F.: Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In: CVPR (2017)

[10] Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: ICML (2015)

[11] Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV (2016)

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

[13] Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML (2015)

[14] Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: ICCV (2015)

[15] Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)

[16] Cao, Z., Long, M., Wang, J., Jordan, M.I.: Partial transfer learning with selective adversarial networks. In: CVPR (2018)

[17] Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: ECCV (2018)

[18] Panareda Busto, P., Gall, J.: Open set domain adaptation. In: ICCV (2017)

[19] Saito, K., Yamamoto, S., Ushiku, Y., Harada, T.: Open set domain adaptation by backpropagation. In: ECCV (2018)

[20] You, K., Long, M., Cao, Z., Wang, J., Jordan, M.I.: Universal domain adaptation. In: CVPR (2019)

[21] Fu, B., Cao, Z., Long, M., Wang, J.: Learning to detect open classes for universal domain adaptation. In: ECCV (2020)

[22] Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: ICLR (2021)

[23] Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: CVPR (2018)

[24] Feng, Q., Kang, G., Fan, H., Yang, Y.: Attract or distract: Exploit the margin of open set. In: ICCV (2019)

[25] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE TKDE **22**(10), 1345–1359 (2010)

[26] Cao, Z., You, K., Long, M., Wang, J., Yang, Q.: Learning to transfer examples for partial domain adaptation. In: CVPR (2019)

[27] Li, S., Liu, C.H., Lin, Q., Wen, Q., Su, L., Huang, G., Ding, Z.: Deep residual correction network for partial domain adaptation. IEEE Trans. PAMI **43**(7), 2329–2344 (2020)

[28] Liang, J., Wang, Y., Hu, D., He, R., Feng, J.: A balanced and uncertainty-aware approach for partial domain adaptation. In: ECCV (2020)

[29] Ren, C.-X., Ge, P., Yang, P., Yan, S.: Learning target-domain-specific classifier for partial domain adaptation. IEEE TNNLS **32**(5), 1989–2001 (2020)

[30] Yan, J., Jing, Z., Leung, H.: Discriminative partial domain adversarial network. In: ECCV (2020)

[31] Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: NeurIPS (2005)

[32] Yang, S., Wang, Y., weijer, J., Herranz, L., JUI, S.: Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In: NeurIPS (2021)

[33] Gu, X., Yu, X., Yang, Y., Sun, J., Xu, Z.: Adversarial reweighting for partial domain adaptation. In: NeurIPS (2021)

[34] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: A kernel method for the two-sample-problem. In: NeurIPS (2006)

[35] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)

[36] Tang, S., Chang, A., Zhang, F., Zhu, X., Ye, M., Zhang, C.: Source-free domain adaptation via target prediction distribution searching. IJCV (In press, 2023)

[37] Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: NeurIPS (2018)

[38] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. JMLR **17**(1), 2096–2030 (2016)

[39] Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: ICML (2018)

[40] Liu, X., Guo, Z., Li, S., Xing, F., You, J., Kuo, C.-C.J., El Fakhri, G., Woo, J.: Adversarial unsupervised domain adaptation with conditional and label shift: Infer, align and iterate. In: ICCV (2021)

[41] Du, Z., Li, J., Su, H., Zhu, L., Lu, K.: Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In: CVPR (2021)

[42] Li, H., Wan, R., Wang, S., Kot, A.C.: Unsupervised domain adaptation in the wild via disentangling representation learning. IJCV **129**, 267–283 (2021)

[43] Gu, X., Sun, J., Xu, Z.: Spherical space domain adaptation with robust pseudo-label loss. In: CVPR (2020)

[44] Wang, X., Li, L., Ye, W., Long, M., Wang, J.: Transferable attention for domain adaptation. In: AAAI (2019)

[45] Yang, S., Wang, Y., Van De Weijer, J., Herranz, L., Jui, S.: Generalized source-free domain adaptation. In: ICCV (2021)

[46] Li, W., Chen, S.: Unsupervised domain adaptation with progressive adaptation of subspaces. PR **132**, 108918 (2022)

[47] Li, W., Chen, S.: Partial domain adaptation without domain alignment. IEEE Trans. PAMI **45**(7), 8787–8797 (2023)

[48] Balgi, S., Dukkipati, A.: Contradistinguisher: A vapnik's imperative to unsupervised domain adaptation. IEEE Trans. PAMI **44**(9), 4730–4747 (2022)

[49] Sahoo, A., Panda, R., Feris, R., Saenko, K., Das, A.: Select, label, and mix: Learning discriminative invariant feature representations for partial domain adaptation. In: WACV (2023)

[50] Xiao, W., Ding, Z., Liu, H.: Implicit semantic response alignment for partial domain adaptation. In: NeurIPS (2021)

[51] Luo, Y.-W., Ren, C.-X.: Mot: Masked optimal transport for partial domain adaptation. In: CPVR (2023)

[52] Liu, H., Cao, Z., Long, M., Wang, J., Yang, Q.: Separate to adapt: Open set domain adaptation via progressive separation. In: CVPR (2019)

[53] Baktashmotlagh, M., Faraki, M., Drummond, T., Salzmann, M.: Learning factorized representations for open-set domain adaptation. In: ICLR (2019)

[54] Kundu, J.N., Venkat, N., Revanur, A., Babu, R.V., *et al.*: Towards inheritable models for open-set domain adaptation. In: CVPR (2020)

[55] Bucci, S., Loghmani, M.R., Tommasi, T.: On the effectiveness of image rotation for open set domain adaptation. In: ECCV (2020)

[56] Rakshit, S., Tamboli, D., Meshram, P.S., Banerjee, B., Roig, G., Chaudhuri, S.: Multi-source open-set deep adversarial domain adaptation. In: ECCV (2020)

[57] Luo, Y., Wang, Z., Huang, Z., Baktashmotlagh, M.: Progressive graph learning for open-set domain adaptation. In: ICML (2020)

[58] Pan, Y., Yao, T., Li, Y., Ngo, C.-W., Mei, T.: Exploring category-agnostic clusters for open-set domain adaptation. In: CVPR (2020)

[59] Jing, M., Li, J., Zhu, L., Ding, Z., Lu, K., Yang, Y.: Balanced open set domain adaptation via centroid alignment. In: AAAI (2021)

[60] Jing, T., Liu, H., Ding, Z.: Towards novel target discovery through open-set domain adaptation. In: ICCV (2021)

[61] Li, W., Liu, J., Han, B., Yuan, Y.: Adjustment and alignment for unbiased open set domain adaptation. In: CVPR (2023)

[62] Saito, K., Kim, D., Sclaroff, S., Saenko, K.: Universal domain adaptation through self supervision. In: NeurIPS (2020)

[63] Saito, K., Saenko, K.: Ovanet: One-vs-all network for universal domain adaptation. In: ICCV (2021)

[64] Li, G., Kang, G., Zhu, Y., Wei, Y., Yang, Y.: Domain consensus clustering for universal domain adaptation. In: CVPR (2021)

[65] Chen, L., Lou, Y., He, J., Bai, T., Deng,

M.: Geometric anchor correspondence mining with uncertainty modeling for universal domain adaptation. In: CVPR (2022)

[66] Kundu, J.N., Bhambri, S., Kulkarni, A.R., Sarkar, H., Jampani, V., Radhakrishnan, V.B.: Subsidiary prototype alignment for universal domain adaptation. In: NeurIPS (2022)

[67] Qu, S., Zou, T., Röhrbein, F., Lu, C., Chen, G., Tao, D., Jiang, C.: Upcycling models under domain and category shift. In: CVPR (2023)

[68] Yang, Y., Gu, X., Sun, J.: Prototypical partial optimal transport for universal domain adaptation. In: AAAI (2023)

[69] Wang, Y., Zhang, L., Song, R., Li, H., Rosin, P.L., Zhang, W.: Exploiting inter-sample affinity for knowability-aware universal domain adaptation. IJCV (In press, 2023)

[70] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NeurIPS (2017)

[71] Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: ICLR (2018)

[72] Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML (2017)

[73] Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: ICML (2020)

[74] Cui, S., Wang, S., Zhuo, J., Li, L., Huang, Q., Tian, Q.: Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In: CVPR (2020)

[75] Chen, M., Xue, H., Cai, D.: Domain adaptation for semantic segmentation with maximum squares loss. In: ICCV (2019)

[76] Zhong, Z., Zheng, L., Cao, D., Li, S.: Re-ranking person re-identification with k-reciprocal encoding. In: CVPR (2017)

[77] Diamond, S., Boyd, S.: Cvxpy: A python-embedded modeling language for convex optimization. JMLR **17**(1), 2909–2913 (2016)

[78] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.*: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)

[79] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)

[80] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV (2015)

[81] Wei, C., Shen, K., Chen, Y., Ma, T.: Theoretical analysis of self-training with deep networks on unlabeled data. In: ICLR (2021)

[82] Cai, T., Gao, R., Lee, J., Lei, Q.: A theory of label propagation for subpopulation shift. In: ICML (2021)

[83] Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: ICLR (2019)

[84] Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV (2010)

[85] Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, Pasadena (2007)

[86] Venkateswara, H., Eusebio, J., Chakraborty,

S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: CVPR (2017)

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

[88] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV (2019)

[89] Guo, P., Zhu, J., Zhang, Y.: Selective partial domain adaptation. In: BMVC (2022)

[90] Lin, K.-Y., Zhou, J., Qiu, Y., Zheng, W.-S.: Adversarial partial domain adaptation by cycle inconsistency. In: ECCV (2022)

[91] Yang, C., Cheung, Y.-M., Ding, J., Tan, K.C., Xue, B., Zhang, M.: Contrastive learning assisted-alignment for partial domain adaptation. IEEE Trans. NNLS **34**(10), 7621–7634 (2023)

[92] Wu, K., Wu, M., Chen, Z., Jin, R., Cui, W., Cao, Z., Li, X.: Reinforced adaptation network for partial domain adaptation. IEEE Trans. CSVT **33**(5), 2370–2380 (2023)

[93] He, C., Li, X., Xia, Y., Tang, J., Yang, J., Ye, Z.: Addressing the overfitting in partial domain adaptation with self-training and contrastive learning. IEEE Trans. CSVT (In press, 2023)

[94] Cao, Z., You, K., Zhang, Z., Wang, J., Long, M.: From big to small: Adaptive learning to partial-set domains. IEEE Trans. PAMI **45**(2), 1766–1780 (2023)

[95] Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: ICCV (2019)

[96] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)

[97] Liang, J., Hu, D., Feng, J., He, R.: Umad: Universal model adaptation under domain and category shift. arXiv preprint arXiv:2112.08553 (2021)

[98] Qu, S., Zou, T., Röhrbein, F., Lu, C., Chen, G., Tao, D., Jiang, C.: Upcycling models under domain and category shift. In: CVPR (2023)

[99] Kundu, J.N., Venkat, N., Babu, R.V., *et al.*: Universal source-free domain adaptation. In: CVPR (2020)

[100] Chang, W., Shi, Y., Tuan, H., Wang, J.: Unified optimal transport framework for universal domain adaptation. In: NeurIPS (2022)

[101] Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: ICML (2020)

[102] Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., Bethge, M.: Improving robustness against common corruptions by covariate shift adaptation. In: NeurIPS (2020)

[103] Mirza, M.J., Micorek, J., Possegger, H., Bischof, H.: The norm must go on: Dynamic unsupervised domain adaptation by normalization. In: CVPR (2022)

[104] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: ICCV (2021)

[105] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. arXiv preprint

arXiv:1406.2661 (2014)

[106] Reddi, S., Ramdas, A., Póczos, B., Singh, A., Wasserman, L.: On the high dimensional power of a linear-time two sample test under mean-shift alternatives. In: AISTATS (2015)

[107] Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR **9**(86), 2579–2605 (2008)