# Reviewing the Forgotten Classes for Domain Adaptation of Black-Box Predictors

**Shaojie Zhang**[1,2], **Chun Shen**[1,2,3], **Shuai Lü**[1,2,3*], **Zeyu Zhang**[1,3]

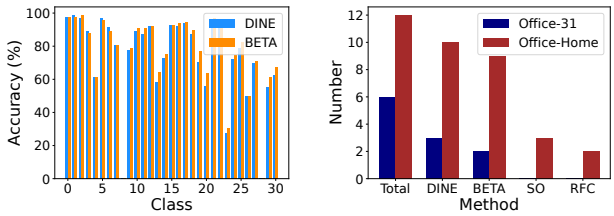
[1]Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, China
[2]College of Software, Jilin University, China
[3]College of Computer Science and Technology, Jilin University, China
{lus, shenchun}@jlu.edu.cn, {sjzhang22, zeyuz22}@mails.jlu.edu.cn

## Abstract

For addressing the data privacy and portability issues of domain adaptation, Domain Adaptation of Black-Box Predictors (DABP) aims to adapt a black-box source model to an unlabeled target domain without accessing both the source-domain data and details of the source model. Although existing DABP approaches based on knowledge distillation (KD) have achieved promising results, we experimentally find that these methods all have the minority class forgetting issue, which refers that the trained model completely forgets some minority classes. To address this issue, we propose a method called Reviewing the Forgotten Classes (RFC), which including two main modules. Firstly, we propose a simple but effective component called selection training (ST). ST selects classes that the model tends to forget according to the learning status of the model and obtains clean samples of the selected classes with the small-loss criterion for enhanced training. ST is orthogonal to previous methods and can effectively alleviate their minority class forgetting issue. Secondly, we find that neighborhood clustering (NC) can help the model learn more balanced than KD so that further alleviate the minority class forgetting issue. However, NC is based on the fact that target features from the source model already form some semantic structure, while DABP is unable to obtain the source model. Thus, we use KD and ST to warm up the target model to form a certain semantic structure. Overall, our method inherits the merits of both ST and NC, and achieves state of the art on three DABP benchmarks.

## Introduction

Deep learning has achieved significant performance on visual tasks in multiple fields, such as image classification (Qian, Hang, and Liu 2022) and semantic segmentation (Huang et al. 2022). However, it is limited by the need for large-scale labeling. Therefore, unsupervised domain adaptation (UDA) is proposed, which aims to transfer the knowledge from the labeled source domain to the unlabeled target domain with access to both source data and models. In view of the importance of data privacy, source-free domain adaptation (SFDA) is proposed, which can adapt a pre-trained source model to the unlabeled target domain without accessing the private source data. Even so, SFDA still suffers from the data privacy issue due to generation techniques like generative adversarial learning, and also requires the target domain to employ the same network as the source domain, which is unfriendly to low-resource target users. Therefore, Domain Adaptation of Black-Box Predictors (DABP) (Liang et al. 2022) is proposed to learn a model with only the unlabeled target-domain data and a black-box source predictor.




(a) Accuracy (b) Forgetting frequency

Figure 1: (a) The per-class accuracies of existing methods on Office-31(W→A); (b) The forgetting frequency on Office-31 and Office-Home; "Total" is the total number of tasks on each dataset and "SO" means the source-only model.

Existing DABP methods have achieved promising performance, but we experimentally find that these methods all suffer from the minority class forgetting issue, which refers that the model trained with existing methods completely forget some classes (the accuracies of these classes are 0%) during the learning process, like the classes 8 and 28 on Office-31 (W→A) shown in Fig. 1(a). The main reason is that these methods utilize the imbalanced predictions of target domain samples from the source model due to the domain shift and teach the model by knowledge distillation throughout the entire training stage, which results that samples actually belonging to minority classes are prone to be pushed in majority classes. Fig. 1(b) shows that current state-of-the-art methods, DINE (Liang et al. 2022) and BETA (Yang et al. 2023), suffer from the minority class forgetting issue severely, especially over 9 (totally 12) transfer tasks on Office-Home for both methods. However, to the best of our knowledge, there is currently no research on this issue. In addition, previous methods use a two-stage knowledge distillation framework, including the distillation stage and the fine-tuning stage,

*Corresponding author.

which is complex and inconvenient for practical scenarios.

In this paper, we propose a method called Reviewing the Forgotten Classes (RFC) that introduces two key modules to solve the minority class forgetting issue. Firstly, we are inspired by an observation that there are still a few reliable predictions for the minority classes from the source model in most tasks as shown in Fig 1(b). Thus, we propose a simple but effective component called selection training (ST) to utilize these clean samples to enhance the memory of the model to the minority classes. Secondly, we experimentally find a property of the neighborhood clustering (NC) method in SFDA that it can help the model learn more balanced than knowledge distillation (KD), which further alleviate the minority class forgetting issue. Specifically, NC encourages local neighbors in the feature space to have more similar predictions than other features.

To implement the above method, ST firstly selects classes that are easily forgotten by the model based on the prediction number and entropy of each class, and then obtains clean samples belonging to the selected classes with the small-loss criterion. We augment the selected samples with weak-strong augmentation and train the model with the noise-robust loss function.

Moreover, current NC methods are not applicable in our setting. The main reason is that these methods are based on the fact that the target features from the source model already form some semantic structures (Yang et al. 2022), while DABP cannot meet the requirement without accessing the source model. To address this issue, we use KD to warm up the model to ensure that it can extract target features with a certain semantic structure, and combine ST to prevent the model from forgetting the minority classes during the warm-up phase. Then, we adopt NC to replace KD for training. However, minority classes sometimes draw the wrong neighboring samples closer, because the semantic structure of the features from the warmed-up model is not always good enough. Therefore, we also combine NC with ST to ensure the stability of minority classes clustering. Finally, as shown in Fig. 1(b), our method can effectively solve the minority class forgetting issue except two extremely hard tasks on Office-Home which the source model already forgets the minority classes. Besides, due to the property of NC, unlike previous methods, our method no longer requires the fine-tuning stage to achieve further performance improvement.

We summary the main contributions as follows:

- We propose a simple but effective component called selection training (ST), which adaptively reviewing classes that tend to be forgetten according to the learning status of the model and is orthogonal to existing DABP methods.
- We experimentally demonstrate that neighborhood clustering (NC) can help models learn more balanced than knowledge distillation (KD), and combine KD and ST to warm up the model so that meet the prerequisite of NC. It provides a new research perspective for DABP.
- We propose a method called Reviewing the Forgotten Classes (RFC), which inherits the merits of ST and NC. This is the first work that addresses the minority class forgetting issue for DABP.
- Extensive experiments demonstrate that RFC achieves state-of-the-art performance consistently on three DABP benchmarks and works well even if the source predictor only provides few labels for the minority classes.

## Related Works

**Source-free domain adaptation.** SFDA (Liang, Hu, and Feng 2020) transfers the knowledge from source domain to unlabeled target domain without accessing the private source data. Yi et al. (2023) focus on addressing the label noise in SFDA. Yang et al. (2021a,b, 2022) propose neighborhood clustering, which enforces prediction consistency between local neighbors. G-SFDA (Yang et al. 2021b) forces the network to activate different channels for different domains and focuses on the neighborhood structure of data. NRC (Yang et al. 2021a) encourages label consistency among data with high local affinity and utilizes a self-regularization term to reduce the negative impact of noisy neighbors. AaD (Yang et al. 2022) obtains two simple terms by optimizing an upper-bound of the neighborhood clustering objective. However, the above neighborhood clustering methods are all based on the fact that target features from the source model already form some semantic structure, which is not satisfied in DABP. For this, we warm up the model with knowledge distillation and ST, and then combines ST with AaD (Yang et al. 2022) to ensure stable neighborhood clustering.

**Domain Adaptation of Black-Box Predictors.** IterLNL (Zhang et al. 2021) first states the setting of DABP and tackles the label noise during adaptation via iterative sample selection. Thus, there are few works conducted in this field. DINE (Liang et al. 2022) proposes a two-step knowledge adaptation framework, which uses knowledge distillation with information maximization (Krause, Perona, and Gomes 2010) and MixUp (Zhang et al. 2017) to solve the DABP problem. BETA (Yang et al. 2023) divides the target domain into two subdomains, then utilizes mutually-distilled twin networks to address the confirmation bias. DINE and BETA utilize KD throughout the training, while our method focus on the minority class forgetting issue from the perspective of NC and works well even if the source predictor only provides few labels for the minority classes.

## Preliminaries

For domain adaptation of black-box predictors, we are only given an unlabeled target domain $\mathcal{D}_t = \{x_i^t\}_{i=1}^{N_t}$ with $N_t$ samples where $x_i^t \in \mathcal{X}_t$, and a black-box predictor $h_s$ trained on a labeled source domain $\mathcal{D}_s = \{x_i^s, y_i^s\}_{i=1}^{N_s}$ where $x_i^s \in \mathcal{X}_s, y_i^s \in \mathcal{Y}_s$. There exists a domain shift between $\mathcal{D}_s$ and $\mathcal{D}_t$. Without accessing details about the source model, e.g., backbone type and network parameters, we can only utilize the predictions of the target samples $\mathcal{X}_t$ from $h_s$ for single-source adaptation in the target domain. Specifically, for the target sample $x_i^t$, we can only obtain its prediction $\tilde{y}_i^t = h_s(x_i^t) \in \mathbb{R}^C$. In this paper, the target model is divided into two parts: the feature encoder $f$, and the classifier $g$. We denote the feature output by the feature en-

coder as $z_i = f(x_i^t) \in \mathbb{R}^d$, where $d$ is the dimension of the feature space, and the prediction of the network as $p_i = \delta(g(z_i)) \in \mathbb{R}^C$ where $\delta$ is the softmax function. We study the single-source closed-set setting where both domains have the same $C$ classes, i.e., $\mathcal{Y}_s = \mathcal{Y}_t$.

DINE (Liang et al. 2022) proposes a two-step knowledge adaptation framework with information maximization and MixUp for DABP. The objective of DINE is formulated as:

$$\mathcal{L}_{DINE} = \mathcal{L}_{kl} - \mathcal{L}_{im} + \beta\mathcal{L}_{mix} \tag{1}$$

$$\mathcal{L}_{kl} = \mathbb{E}_{x_i^t \in \mathcal{X}_t} \mathcal{D}_{kl}(p_i^T || p_i) \tag{2}$$

$$\mathcal{L}_{im} = H(\mathbb{E}_{x_i^t \in \mathcal{X}_t} p_i) - \mathbb{E}_{x_i^t \in \mathcal{X}_t} H(p_i) \tag{3}$$

$$\mathcal{L}_{mix} = \mathbb{E}_{x_i^t, x_j^t \in \mathcal{X}_t} \mathbb{E}_{\lambda \in \mathrm{Beta}(\alpha,\alpha)} l_{ce}(Mix_\lambda(p_i', p_j'), p^{mix}) \tag{4}$$

where $\mathcal{D}_{kl}(\cdot||\cdot)$ denotes the KL-divergence, and $p_i^T$ is the exponential moving average (EMA) prediction of the sample $x_i^t$, i.e., $p_i^T = \gamma p_i^T + (1-\gamma)p_i$. $H(\cdot)$ denotes the information entropy, and $l_{ce}$ is the cross-entropy loss. $Mix_\lambda(a,b) = \lambda a + (1-\lambda)b$ denotes the MixUp (Zhang et al. 2017) operation, and $\lambda$ is sampled from a Beta distribution. $p'$ is the no-gradient value of $p$ and $p^{mix}$ is the prediction of the sample $Mix_\lambda(x_i^t, x_j^t)$. $\beta$, $\gamma$ and $\alpha$ are hyper-parameters.

AaD (Yang et al. 2022) optimizes an upper-bound of the neighborhood clustering objective and obtains two simple terms, formulated as:

$$\mathcal{L}_{AaD} = \mathbb{E}_{x_i^t \in \mathcal{X}_t}[-\sum_{j \in \mathcal{C}_i} p_i^{\mathrm{T}} p_j + \lambda' \sum_{m \in \mathcal{B}_i} p_i^{\mathrm{T}} p_m] \tag{5}$$

where $\mathcal{C}_i$ includes $K$ nearest samples of $x_i^t$ and $\mathcal{B}_i$ includes all other samples except $x_i^t$ in the current mini-batch as the potential dissimilar samples. Note that, in order to efficiently retrieve nearest neighbors for training, AaD (Yang et al. 2022) builds two memory banks to store all target features along with their predictions, and only the features along with their predictions computed in each mini-batch are used to update the memory bank. Yang et al. (2022) empirically find that using a hyper-parameter $\lambda'$ to decay the second term works better than a constant.

## Method

**Overview.** Our method is proposed to solve the minority class forgetting issue, which introduces two main modules. Firstly, we propose selection training to enhance the model to remember the minority classes. This module is orthogonal to the training process of existing DABP methods, and we only need to plug into selection training after their training step. Secondly, we experimentally find that neighborhood clustering can help the model learn more balanced than knowledge distillation, which can further alleviate the minority class forgetting issue. However, unlike SFDA, we can not access the source model. Thus, in order to ensure that the target model can form a certain semantic structure, we warm up it with knowledge distillation. Due to the simplicity and effectiveness of DINE (Liang et al. 2022), we directly combined it with selection training to warm up the model. Then, we replace knowledge distillation with neighborhood clustering for balanced training. Specifically, we adopt AaD (Yang et al. 2022), which only needs to optimize two simple terms, to represent neighborhood clustering methods. The full algorithm of RFC is shown in Algorithm 1.

Algorithm 1: RFC (Reviewing the Forgotten Classes)

**Input**: Target Data $\mathcal{X}_t$, Training Epochs $T_{max}$, Warm-up Epochs $T_w$

Build memory banks storing all target features $M_f$ and predictions $M_p$.
**for** $epoch = 1$ to $T_{max}$ **do**
  **if** $epoch \leq T_w$ **then**
    Sample batch from $\mathcal{X}_t$ and update model by minimizing $\mathcal{L}_{DINE}$.
  **else**
    Sample batch from $\mathcal{X}_t$ and update the memory banks $M_f$ and $M_p$.
    Update model by minimizing $\mathcal{L}_{AaD}$.
  **end if**
  Obtain the selected class set $\mathcal{S}_c$ based on the learning status of the model.
  Sample based on the EMA small-loss criterion for each class in $\mathcal{S}_c$ and obtain the selected sample set $\mathcal{S}_s$.
  Sample batch from $\mathcal{S}_s$ and update model by minimizing $\mathcal{L}_{sel}$ and $\mathcal{L}_{rce}$.
  Update the EMA prediction $p_i^T$.
**end for**
**return** Target Adapted Model

### Selection Training

For every epoch, we firstly select the classes that the model tends to forget according to the learning status of the model, and then obtain clean samples belonging to the selected classes with the small-loss criterion. Finally, we augment the selected samples with weak-strong augmentation and train the model with the noise-robust loss function.

**Adaptive class selection.** Intuitively, the class with fewer samples predicted by the model is considered more likely to be forgotten, formulated as:

$$\sigma(c) = \sum_{i=1}^{N_t} \mathbb{1}(argmax(p_i) = c), c \in \{1, ..., C\} \tag{6}$$

where $\sigma(c)$ is the number of samples belonging to the class $c$ predicted by the model, and $\mathbb{1}(\cdot)$ is the indicator function. Then, we sort all classes in ascending order based on $\sigma(\cdot)$ and obtain $rank^{num} = [rank^{num}(1), ..., rank^{num}(C)]$. $rank^{num}(i)$ represents the prediction number ranking of the $i$-th class.

In fact, the target domain itself is class-imbalanced so that there are a few classes whose ground truth sample numbers are small. For these classes, their high rankings of prediction number are not necessary to mean that the model cannot learn them well. We follow the intuition that the model can make high confidence predictions for the classes that it learns well and the high confidence prediction is equal to

low prediction entropy. Thus, except the prediction number of classes, we also calculate the prediction entropy among all classes for every epoch, formulated as:

$$ent(c) = -\frac{1}{N_c}\sum_{i=1}^{N_t} \mathbb{1}(\hat{y}_i^t = c) \cdot p_i log p_i \tag{7}$$

where $ent(c)$ indicates the prediction entropy of the class $c$ at the current epoch. $N_c = \sum_{i=1}^{N_t} \mathbb{1}(\hat{y}_i^t = c)$, where $\hat{y}_i^t$ is the pseudo label of the sample $x_i^t$ based on the EMA prediction $p_i^T$ rather than the current prediction $p_i$. The main reason is that the current predictions of the model fluctuate greatly in the early training stage, which would make the estimated class entropy become unstable and inaccurate. Then, we sort all classes in descending order based on $ent(\cdot)$ and obtain $rank^{ent} = [rank^{ent}(1), ..., rank^{ent}(C)]$, because the prediction confidence of the classes with high entropy is low, which means the model learns these classes not well.

We comprehensively consider the above two rankings for each class to select the classes that the model tends to forget, i.e., few predictions and high entropy. Specifically, we obtain the total ranking for each class by directly summing the two rankings, formulated as:

$$rank(c) = rank^{num}(c) + rank^{ent}(c) \tag{8}$$

where $rank(c)$ is the total ranking of the class $c$. The higher the $rank(c)$, the larger the possibility of the class $c$ being forgotten. Thus, we select $N_{cls}$ classes with the highest ranking to obtain a selected class set $\mathcal{S}_c$, where $N_{cls} = \lfloor \tau \cdot C \rfloor$.

**Sample selection.** Due to the domain shift, the predictions from the source model are highly possible to be inaccurate and noisy. Thus, we further try to select clean samples among the selected classes with the small-loss criterion (Gui, Wang, and Tian 2021) for selection training. Different from using the mean loss in RSL (Gui, Wang, and Tian 2021), we use EMA to update the losses of samples for every epoch so that no longer need to store the historical losses for all samples:

$$l^T(x_i^t) = \gamma l^T(x_i^t) + (1-\gamma) l(x_i^t) \tag{9}$$

where $l^T(x_i^t)$ is the EMA loss of the sample $x_i^t$, and $l(x_i^t)$ is calculated by the cross-entropy loss at the current epoch. For each selected class, we adopt samples with the $r$ ratio smallest EMA losses, where $r \in [0, 1]$.

However, when facing the extreme situation where some classes in $\mathcal{S}_c$ have extremely few samples predicted by the source model in the beginning, e.g., $N_c \leq 5$, such a sample strategy can not obtain sufficient samples for selection training. For example, if $N_c = 4$ and $r = 0.4$, we can only obtain $\lfloor N_c \cdot r \rfloor = 1$ sample of the class $c$, which is not enough to make a significant positive impact on the model. To ensure our method can work in the extreme situation, we improve the sample strategy with a non-linear mapping function:

$$r_c = min\{1, r\sqrt{\frac{N_{median}}{N_c}}\}, c \in \mathcal{S}_c \tag{10}$$

where $r_c$ is the sampling ratio for the class $c$. $N_{median}$ is the median of the prediction number in $\mathcal{S}_c$. By using the improved strategy, we can enable almost all samples of the classes that have extremely few predicted samples to participate in training. Moreover, we can obtain a more balanced sample set $\mathcal{S}_s = \{x_i^t \mid \hat{y}_i^t = c \wedge l^T(x_i^t) \leq l^T(x_c'), c \in \mathcal{S}_c\}$, where $l^T(x_c')$ is the $k_c^{th}$ smallest EMA loss in the selected class $c$ and $k_c = \lfloor N_c \cdot r_c \rfloor$. We show the difference between non-linear mapping function and linear mapping function in the Fig. 2, and set $r = 0.4$, $N_{median} = 20$. But in practical, $r$ is a hyper-parameter and $N_{median}$ is a variable. Non-linear mapping function can sample as many as possible for the extreme minority classes to make them are adequately trained and sample less than linear mapping function for the majority classes in $\mathcal{S}_c$ to train more balanced.

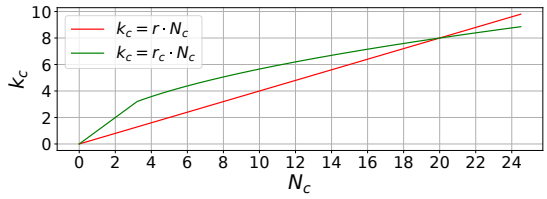



Figure 2: Visualization of the proposed sample strategy.

**Noise-robust training.** We utilize the samples in $\mathcal{S}_s$ to enhance the model to learn the classes in $\mathcal{S}_c$. Specifically, we adopt the weak augmentation (i.e., random cropping and flipping) and the strong augmentation (i.e., RandAugment (Cubuk et al. 2020) and AutoAugment (Cubuk et al. 2019)) to enlarge the sample set. Formally, the selection training objective for the selected samples is:

$$\mathcal{L}_{sel} = -\frac{1}{2N_{sa}}\sum_{i=1}^{N_{sa}} \hat{y}_i^t log p_i^{we} + \hat{y}_i^t log p_i^{st} \tag{11}$$

where $N_{sa}$ is the size of $\mathcal{S}_s$. We use $p_i^{we}$ and $p_i^{st}$ to denote the output of the sample $x_i^t$ augmented by weak and strong augmentation, respectively.

Although we filter out most noise with the EMA small-loss criterion, there may be still few noisy samples in $\mathcal{S}_s$ due to the high noise ratio in DABP. Thus, we combine $\mathcal{L}_{sel}$ with the reverse cross-entropy loss $\mathcal{L}_{rce}$ (Wang et al. 2019) to further reduce the negative impact from noisy samples:

$$\mathcal{L}_{rce} = -\frac{1}{N_{sa}}\sum_{i=1}^{N_{sa}} \bar{p}_i log q_i \tag{12}$$

$$\bar{p}_i = \frac{1}{2}(p_i^{we} + p_i^{st}) \tag{13}$$

where $q_i$ is the one-hot coding of $\hat{y}_i^t$. Finally, we update the EMA predictions of the selected samples to avoid them being classified into other wrong classes.

## Experiment

**Datasets. Office-31** is a small-scale benchmark with 3 domains, **A**mazon, **D**SLR and **W**ebcam. This dataset contains 31 classes. **Office-Home** is a challenging benchmark which contains totally 15500 images from 65 classes and

| Method | MF | A→D | A→W | D→A | D→W | W→A | W→D | Avg. |
|---|---|---|---|---|---|---|---|---|
| Source Only | × | 78.9 | 74.8 | 57.0 | 91.8 | 62.0 | 98.8 | 77.2 |
| NRC (Yang et al. 2021a) | × | 96.0 | 90.8 | 75.3 | 99.0 | 75.0 | 100.0 | 89.4 |
| AaD (Yang et al. 2022) | × | 96.4 | 92.1 | 75.0 | 99.1 | 76.5 | 100.0 | 89.9 |
| LNL-OT (Asano et al. 2020) | ✓ | 88.8 | 85.5 | 64.6 | 95.1 | 66.7 | 98.7 | 83.2 |
| LNL-KL (Zhang et al. 2021) | ✓ | 89.4 | 86.8 | 65.1 | 94.8 | 67.1 | 98.7 | 83.2 |
| HD-SHOT (Liang et al. 2022) | ✓ | 86.5 | 83.1 | 66.1 | 95.1 | 68.9 | 98.1 | 83.0 |
| SD-SHOT (Liang et al. 2022) | ✓ | 89.2 | 83.7 | 67.9 | 95.3 | 71.1 | 97.1 | 84.1 |
| DINE (Liang et al. 2022) | ✓ | 91.6 | 86.8 | 72.2 | **96.2** | 73.3 | 98.6 | 86.4 |
| BETA (Yang et al. 2023) | ✓ | 93.6 | 88.3 | 76.1 | 95.5 | 76.5 | **99.0** | 88.2 |
| RFC (Ours) | ✓ | **94.4** | **93.0** | **76.7** | 95.6 | **77.5** | 98.1 | **89.2** |

Table 1: Accuracies (%) on Office-31 (ResNet-50). Bold text indicates the best results.

| Method | MF | 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. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Only | × | 44.5 | 67.8 | 74.5 | 53.8 | 62.6 | 65.9 | 53.3 | 42.1 | 73.5 | 66.3 | 45.8 | 77.6 | 60.6 |
| G-SFDA (Yang et al. 2021b) | × | 57.9 | 78.6 | 81.0 | 66.7 | 77.2 | 77.2 | 65.6 | 56.0 | 82.2 | 72.0 | 57.8 | 83.4 | 71.3 |
| NRC (Yang et al. 2021a) | × | 57.7 | 80.3 | 82.0 | 68.1 | 79.8 | 78.6 | 65.3 | 56.4 | 83.0 | 71.0 | 58.6 | 85.6 | 72.2 |
| AaD (Yang et al. 2022) | × | 59.3 | 79.3 | 82.1 | 68.9 | 79.8 | 79.5 | 67.2 | 57.4 | 83.1 | 72.1 | 58.5 | 85.4 | 72.7 |
| LNL-OT (Asano et al. 2020) | ✓ | 49.1 | 71.7 | 77.3 | 60.2 | 68.7 | 73.1 | 57.0 | 46.5 | 76.8 | 67.1 | 52.3 | 79.5 | 64.9 |
| LNL-KL (Zhang et al. 2021) | ✓ | 49.0 | 71.5 | 77.1 | 59.0 | 68.7 | 72.9 | 56.4 | 46.9 | 76.6 | 66.2 | 52.3 | 79.1 | 64.6 |
| HD-SHOT(Liang et al. 2022) | ✓ | 48.6 | 72.8 | 77.0 | 60.7 | 70.0 | 73.2 | 56.6 | 47.0 | 76.7 | 67.5 | 52.6 | 80.2 | 65.3 |
| SD-SHOT(Liang et al. 2022) | ✓ | 50.1 | 75.0 | 78.8 | 63.2 | 72.9 | 76.4 | 60.0 | 48.0 | 79.4 | 69.2 | 54.2 | 81.6 | 67.4 |
| DINE (Liang et al. 2022) | ✓ | 52.2 | 78.4 | 81.3 | 65.3 | 76.6 | 78.7 | 62.7 | 49.6 | 82.2 | 69.8 | 55.8 | 84.2 | 69.7 |
| BETA (Yang et al. 2023) | ✓ | 57.2 | 78.5 | 82.1 | **68.0** | 78.6 | 79.7 | 67.5 | 56.0 | **83.0** | 71.9 | 58.9 | 84.2 | 72.1 |
| RFC (Ours) | ✓ | **57.4** | **80.0** | **82.8** | 67.0 | **80.6** | **80.2** | **68.3** | **57.8** | 82.8 | **72.8** | **59.3** | **85.9** | **72.9** |

Table 2: Accuracies (%) on Office-Home (ResNet-50). Bold text indicates the best results.

4 domains, **A**rt, **C**lipart, **P**roduct, **R**eal World. **VisDA-17** is a large-scale benchmark, with a 12-class synthetic-to-real transfer task. The source domain contains 152397 synthetic images and the target domain consists of 55388 real-world images.

**Implementation details.** Generally, we run our methods three times via PyTorch, and report the average accuracies. For a fair comparison, we use the same backbones as the previous methods, i.e., ResNet-50 for Office-31 and Office-Home, and ResNet-101 for VisDA-17. We utilize the ImageNet pre-trained model as initialization and employ mini-batch SGD to optimize the network with the learning rate of 1e-3 for the feature encoder and 1e-2 for the Multi-Layer Perception (MLP) classifier. Following the previous method (Yang et al. 2023), we use the suggested training strategies including the momentum (0.9), batch size (64), and weight decay (1e-3). We also set the same training epoch as BETA (Yang et al. 2023), i.e., $T_{max} = 50$ for Office-31 and Office-Home, and 10 for VisDA-17. The warm-up epoch $T_w$ is set 5 except 3 for VisDA-17. The hyper-parameters of DINE (Liang et al. 2022) and AaD (Yang et al. 2022) are mostly kept same as the original paper, which is attached in the Appendix. Moreover, for our two hyper-parameters, we set $\tau = 0.2$, $r = 0.4$ except $\tau = 0.1$, $r = 0.05$ for VisDA-17. Code is available at https://github.com/shaojiezhanglalala/RFC.

**Baselines.** "Source Only" is the model trained with the same training strategy as BETA (Yang et al. 2023). We compare our method with state-of-the-art DABP methods, including LNL-OT (Asano, Rupprecht, and Vedaldi 2020), LNL-KL (Zhang et al. 2021), HD-SHOT, SD-SHOT, DINE (Liang et al. 2022) and BETA (Yang et al. 2023). Note that, the results of HD-SHOT and SD-SHOT are both reported by DINE. Moreover, we also compare some related SFDA methods which use neighborhood clustering, including G-SFDA (Yang et al. 2021b), NRC (Yang et al. 2021a) and AaD (Yang et al. 2022).

## Performance Comparison

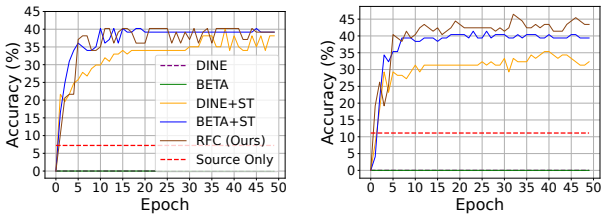
The results on Office-31, Office-Home and VisDA-17 are shown in Tables 1, 2 and 3 respectively. "MF" means model-free. ✓ indicates the DABP method, and × indicates the SFDA method. Overall, RFC surpasses the state-of-the-art DABP methods on all three benchmarks. For Office-31, simple but with the minority class forgetting issue, the performance of our method is also competitive compared to the SFDA methods. For Office-Home, the dataset with the most severe minority class forgetting issue, our method even outperforms than the SFDA methods. This demonstrates that addressing the minority class forgetting issue is crucial for achieving good performance in DABP. Although, the minority class forgetting issue does not exist in VisDA-17, our method can also outperform BETA, which is twin network

| Method | MF | plane | bcycl | bus | car | horse | knife | mcycl | person | plant | sktbrd | train | truck | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Only | ✗ | 64.3 | 24.6 | 47.9 | 75.3 | 69.6 | 8.5 | 79.0 | 31.6 | 64.4 | 31.0 | 81.4 | 9.2 | 48.9 |
| G-SFDA (Yang et al. 2021b) | ✗ | 96.1 | 88.3 | 85.5 | 74.1 | 97.1 | 95.4 | 89.5 | 79.4 | 95.4 | 92.9 | 89.1 | 42.6 | 85.4 |
| NRC (Yang et al. 2021a) | ✗ | 96.8 | 91.3 | 82.4 | 62.4 | 96.2 | 95.9 | 86.1 | 80.6 | 94.8 | 94.1 | 90.4 | 59.7 | 85.9 |
| AaD (Yang et al. 2022) | ✗ | 97.4 | 90.5 | 80.8 | 76.2 | 97.3 | 96.1 | 89.8 | 82.9 | 95.5 | 93.0 | 92.0 | 64.7 | 88.0 |
| LNL-OT (Asano et al. 2020) | ✓ | 82.6 | 84.1 | 76.2 | 44.8 | 90.8 | 39.1 | 76.7 | 72.0 | 82.6 | 81.2 | 82.7 | 50.6 | 72.0 |
| LNL-KL (Zhang et al. 2021) | ✓ | 82.7 | 83.4 | 76.7 | 44.9 | 90.9 | 38.5 | 78.4 | 71.6 | 82.4 | 80.3 | 82.9 | 50.4 | 71.9 |
| HD-SHOT (Liang et al. 2022) | ✓ | 75.8 | 85.8 | 78.0 | 43.1 | 92.0 | 41.0 | 79.9 | 78.1 | 84.2 | 86.4 | 81.0 | **65.5** | 74.2 |
| SD-SHOT (Liang et al. 2022) | ✓ | 79.1 | 85.8 | 77.2 | 43.4 | 91.6 | 41.0 | 80.0 | 78.3 | 84.7 | 86.8 | 81.1 | 65.1 | 74.5 |
| DINE (Liang et al. 2022) | ✓ | 81.4 | 86.7 | 77.9 | 55.1 | 92.2 | 34.6 | 80.8 | 79.9 | 87.3 | 87.9 | 84.3 | 58.7 | 75.6 |
| BETA (Yang et al. 2023) | ✓ | 94.9 | **90.2** | 85.4 | 61.1 | 95.5 | 93.1 | 85.0 | **83.8** | 92.9 | **91.9** | **91.1** | 55.0 | 85.1 |
| RFC (Ours) | ✓ | **95.6** | 89.7 | **87.8** | **75.8** | **96.5** | **96.5** | **90.4** | 82.8 | **96.0** | 70.0 | 85.7 | 55.1 | **85.2** |

Table 3: Accuracies (%) on VisDA-17 (ResNet-101). Bold text indicates the best results.

| Method | FT | Office-31 | Office-Home |
|---|---|---|---|
| DINE (Liang et al. 2022) | ✗ | 85.5 | 68.4 |
| | ✓ | 86.4 | 69.9 |
| DINE+ST | ✗ | 88.3 | 70.6 |
| | ✓ | 88.8 | 71.6 |
| BETA (Yang et al. 2023) | ✗ | 86.9 | 70.3 |
| | ✓ | 87.7 | 71.2 |
| BETA+ST | ✗ | 88.8 | 71.9 |
| | ✓ | 89.2 | 72.4 |

Table 4: Improvement by combining ST with existing DABP methods on Office-31 and Office-Home.




(a) Class 8 (b) Class 28

Figure 3: The accuracies of the forgotten classes for 50 epochs on Office-31(W→A).

architecture and time-consuming, by simply combining our selection training with neighborhood clustering. There is still a large gap between DABP and SFDA on VisDA-17, and the possible reason is that the target feature semantic structure from the model trained by distillation is not good enough as that from the source model, which needs to be investigated in the future.

## Improvement by Combining ST with Existing DABP Methods

Our proposed selection training module is orthogonal to the previous methods. Thus, we plug ST into DINE (Liang et al. 2022) and BETA (Yang et al. 2023) and train the models on Office-31 and Office-Home. DINE (Liang et al. 2022) and BETA (Yang et al. 2023) both introduce a secondary training phase, fine-tuning the distilled model, so we report the results in the training phase and the fine-tuning phase respectively. In Table 4, "FT" indicates the fine-tuning phase. ✗ indicates the results after the training phase, and ✓ indicates the results after the fine-tuning phase. As shown in Table 4, ST significantly boosts the performance of both two methods, especially for the training phase of DINE on Office-31 (+2.8%) and Office-Home (+2.2%). Moreover, as shown in Fig. 3, ST effectively improves the accuracies of the forgotten classes for both methods on Office-31 (the results on Office-Home are shown in Appendix), which demonstrates that our selection training does work again. Benefitting from neighborhood clustering, our method surpasses the improved methods, "DINE+ST" and "BETA+ST", and no longer requires the fine-tuning stage to achieve further performance improvement.

| Method | Office-31 | Office-Home | VisDA-17 |
|---|---|---|---|
| RFC (Ours) | 89.2 | 72.9 | 85.2 |
| w/o ST | 87.7 | 71.5 | 84.6 |
| w/o AaD | 88.3 | 70.6 | 80.7 |
| w/o entropy | 88.3 | 72.2 | 83.8 |
| w/o small-loss | 88.1 | 70.2 | 77.7 |
| w/o $\mathcal{L}_{rce}$ | 88.3 | 72.6 | 84.9 |

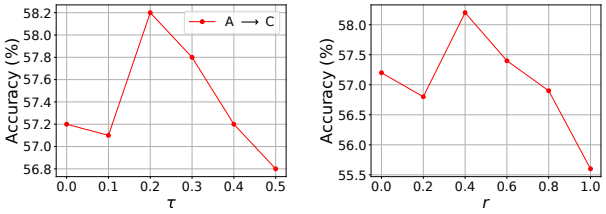
Table 5: Ablation study results on three datasets. "w/o entropy" indicates that we select the minority classes only based on the prediction number.

## Ablation Study

**Component-wise Analysis.** To demonstrate that the key components of our method is effective, we conduct experiments on all three benchmarks and show the results in Table 5. When the whole ST is dropped, the performance of our method decreases significantly, verifying its importance. "w/o AaD" is equal to training with DINE and ST, and the

| Task | Class | SO_Pred. | SO_Acc.(%) | w/o nl(%) | w/ nl(%) |
|---|---|---|---|---|---|
| C→P | 17 | 4 | 75.0 | 0.0 | 45.2 |
| C→R | 64 | 4 | 25.0 | 0.0 | 52.2 |
| R→A | 64 | 3 | 66.7 | 0.0 | 15.0 |
| P→A | 20 | 4 | 0.0 | 0.0 | 5.3 |

Table 6: Improvement by the non-linear mapping function for extremely hard tasks on Office-Home. "SO_Pred." and "SO_Acc." indicate the prediction numbers and accuracies of the classes from the black-box source model, respectively. "nl" means the non-linear mapping function.



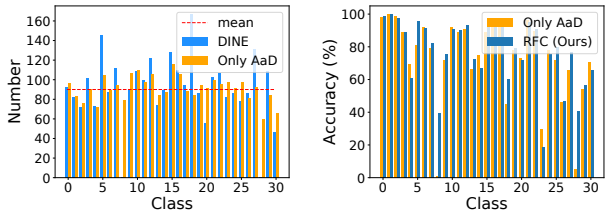

(a) Senseitivity to $\tau$ ($r$=0.4) (b) Senseitivity to $r$ ($\tau$=0.2)

Figure 4: The test accuracy with different values of $\tau$ and $r$ on Office-Home(A→C).

results show that neighborhood clustering can further improve the performance through balancing the training among classes. If we select the minority classes only based on the prediction number, the performance drops as shown in the row "w/o entropy", which shows that we should consider both the prediction number and the prediction entropy for more accurate selection. The performances of "w/o small-loss" and "w/o $\mathcal{L}_{rce}$" are both decreased, which demonstrate that training with less noise is crucial for better performance.

**Non-linear mapping function for hard tasks.** Due to the distant domain shift, the black-box source model predicts extremely imbalanced in the target domain so that some classes even only have few predictions less than 5 and there is still noise in these classes as shown in Table 6. Facing the extreme situation, we are more likely to select as many samples as possible for training compared with noise-robust training. We employ the non-linear mapping function to ensure more samples of the minority classes participating in training. As shown in Table 6, such a strategy significantly improves the accuracies of the minority classes in the hard tasks. Especially, for the P→A task, although the source model has already forgotten the class 20, the model trained with our method can recognize it again. The possible reason is that the source model makes a few wrong predictions for the class 20 and the model remembers these wrong samples by ST during the warm-up phase. Then, neighborhood clustering makes some true samples similar to the wrong samples be predicted truly.

**Hyper-parameter sensitivity.** We study $\tau$ and $r$ on Office-Home (A→C). We choose $\tau$ ranging from 0 to 0.5 and $r \in [0, 0.2, 0.4, 0.6, 0.8, 1]$. Fig. 4(a) shows that too small $\tau$ leads to insufficient training for some minority classes and too large $\tau$ results that some majority classes participant in training which indirectly decrease the enhanced training effect of the minority classes. Fig. 4(b) shows that too small $r$ results in less selected samples for the minority classes and too large $r$ leads to more noise in training. Thus, we set $\tau = 0.2$ and $r = 0.4$ for a reasonable trade-off between sufficient training and noise-robust training.




(a) Prediction distribution (b) Accuracy

Figure 5: (a) The prediction distribution of knowledge distillation and neighborhood clustering; (b) The per-class accuracies of our method with ST and without ST; Both experiments are conducted on Office-31(W→A).

## Difference between Knowledge Distillation and Neighborhood Clustering

We utilize DINE to represent methods based on knowledge distillation and "Only AaD" to represent methods based on neighborhood clustering. For "Only AaD", we firstly adopt DINE to warm up the model for 5 epochs and directly apply AaD for the rest 45 epochs to ensure the training epoch is same as DINE. "mean" is the uniform distribution. Fig. 5(a) shows that neighborhood clustering can make the model learn more balanced than knowledge distillation. However, as shown in Fig. 5(b), although neighborhood clustering can make a balanced prediction distribution, it cannot draw the true neighboring samples closer for the minority classes without a good target feature semantic structure from the model. Compared with "Only AaD", our method utilizes selection training during the entire training stage, and Fig. 5(b) shows that selection training is key to ensuring RFC works well. More extended experiments are shown in Appendix.

## Conclusion

In this work, we experimentally find that existing DABP methods suffer from the minority class forgetting issue and propose a simple Reviewing the Forgotten Classes (RFC) method to address it. Specifically, we introduce selection training, which adaptively selects the minority classes based on the learning status of the model and obtains the clean samples with the small-loss criterion for enhanced training. This key module is orthogonal to the previous methods and improved their performance by combining with it. Moreover, our method also adopt neighborhood clustering for training more balanced, which further prevents the model from forgetting the minority classes. Extensive experiments verify that our method is effective to solve the minority class forgetting issue and achieves state-of-the-art results on all three benchmarks.

## Acknowledgements

We sincerely thank the anonymous reviewers for their careful work and thoughtful suggestions, which have greatly improved this article. This work was supported by the Natural Science Research Foundation of Jilin Province of China under Grant Nos. 20220101106JC and YDZJ202201ZYTS423.

## References

Asano, Y. M.; Rupprecht, C.; and Vedaldi, A. 2020. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 113–123.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE Conference on Computer Vision and Pattern Recognition*, 702–703.

Gui, X.-J.; Wang, W.; and Tian, Z.-H. 2021. Towards understanding deep learning from noisy labels with small-loss criterion. In *International Joint Conferences on Artificial Intelligence*, 2469–2475.

Huang, Y.; Kang, D.; Jia, W.; Liu, L.; and He, X. 2022. Channelized axial attention–considering channel relation within spatial attention for semantic segmentation. In *AAAI Conference on Artificial Intelligence*, 1016–1025.

Krause, A.; Perona, P.; and Gomes, R. 2010. Discriminative clustering by regularized information maximization. In *Annual Conference on Neural Information Processing Systems*, 775–783.

Liang, J.; Hu, D.; and Feng, J. 2020. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, 6028–6039.

Liang, J.; Hu, D.; Feng, J.; and He, R. 2022. DINE: Domain adaptation from single and multiple black-box predictors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 8003–8013.

Qian, X.; Hang, R.; and Liu, Q. 2022. ReX: An efficient approach to reducing memory cost in image classification. In *AAAI Conference on Artificial Intelligence*, 2099–2107.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric cross entropy for robust learning with noisy labels. In *IEEE International Conference on Computer Vision*, 322–330.

Yang, J.; Peng, X.; Wang, K.; Zhu, Z.; Feng, J.; Xie, L.; and You, Y. 2023. Divide to adapt: Mitigating confirmation bias for domain adaptation of black-box predictors. In *International Conference on Learning Representations*.

Yang, S.; Wang, Y.; van de Weijer, J.; Herranz, L.; and Jui, S. 2021a. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *Annual Conference on Neural Information Processing Systems*, 29393–29405.

Yang, S.; Wang, Y.; van de Weijer, J.; Herranz, L.; and Jui, S. 2021b. Generalized source-free domain adaptation. In *IEEE International Conference on Computer Vision*, 8978–8987.

Yang, S.; Wang, Y.; Wang, K.; Jui, S.; and van de Weijer, J. 2022. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Annual Conference on Neural Information Processing Systems*, 5802–5815.

Yi, L.; Xu, G.; Xu, P.; Li, J.; Pu, R.; Ling, C.; McLeod, A. I.; and Wang, B. 2023. When source-free domain adaptation meets learning with noisy labels. In *International Conference on Learning Representations*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Zhang, H.; Zhang, Y.; Jia, K.; and Zhang, L. 2021. Unsupervised domain adaptation of black-box source models. In *British Machine Vision Conference*, 147.