

Article

Progressive Optimization of Target Distribution and Effective Refinement of Hard Samples for Source-Free Domain Adaptation

Shumin Liang , Xiaorong Hou , Yajian Zeng and Xinrui Wang

School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; liangsm@std.uestc.edu.cn (S.L.); 202111060901@std.uestc.edu.cn (Y.Z.); 202311060938@std.uestc.edu.cn (X.W.)

* Correspondence: houxr@uestc.edu.cn

Abstract

In recent years, the research on source-free domain adaptation has received increasing attention and has achieved considerable progress. It can overcome the dependence on source domain data and obtain a target domain model with robust performance on the target domain only by using the source domain model and unlabeled target domain data. However, existing studies typically handle the target domain distribution in a relatively coarse manner and are consistently susceptible to model noise interference. Therefore, we propose a progressive optimization strategy for the target domain distribution, including two parts: inter-category and intra-category. Regarding inter-category, we decide whether to separate category pairs based on the degree of discrepancy between them. Regarding intra-category, we consider whether to aggregate sample pairs based on whether their pairwise similarity—among samples assigned to the same predicted category—is sufficiently high. And as the training progresses, all data will be optimized. Additionally, for some difficult-to-distinguish categories, we propose a screening strategy that fuses source domain and target domain knowledge. We also further optimized the samples belonging to these categories. Our results on three image datasets demonstrate the effectiveness of our method.

Keywords: source-free domain adaptation; target distribution; hard sample

1. Introduction

Deep neural networks have achieved remarkable success in various artificial intelligence tasks, including object recognition, object detection, and image segmentation. However, their performance heavily depends on large amounts of labeled training data. When deploying a trained model to a new domain—even one with the same category set—performance often degrades noticeably. To address this challenge, unsupervised domain adaptation (UDA) aims to transfer knowledge from a labeled source domain to an unlabeled target domain. A key limitation of UDA is its reliance on access to source data during the adaptation process, which may be restricted due to privacy policies or storage constraints. Source-free domain adaptation (SFDA) overcomes this obstacle by performing adaptation using only a pre-trained source model and unlabeled target data, eliminating the need for original source data.

Substantial advancements have been achieved in source-free domain adaptation research, which can broadly be categorized into three primary research directions. The first one is the pseudo-labeling strategy, which mainly assigns labels to target domain data



Academic Editors: Shaolin Liao, Jinxin Li and Tiaojie Xiao

Received: 30 January 2026

Revised: 12 March 2026

Accepted: 13 March 2026

Published: 15 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

via cluster centers or neighboring samples, thereby facilitating the model training process [1–8]. The second one is the contrastive learning strategy, which aims to minimize the discrepancy between positive sample pairs while maximizing that between negative sample pairs [9–12]. The third one is based on diversity and discriminability, with its objective being the optimization of target domain data distribution [3,4,11,13–16].

Existing studies on the optimization of the target domain distribution remain relatively coarse-grained and are often susceptible to noise in the model outputs. With the intention of alleviating this problem, we optimize the inter-category and intra-category distributions simultaneously. Specifically, we aim to separate different categories while compressing samples belonging to the same category. Moreover, taking into account the presence of model prediction noise, it will have a negative impact on the process of optimizing the distribution. That is, it will erroneously bring samples of different categories closer and samples of the same category further apart. Therefore, we adopt a progressive strategy for distribution optimization. We prioritize the separation of category pairs with significant differences and the aggregation of sample pairs of the same category with high similarity. This is carried out to prevent the inappropriate treatment of certain easily confused categories during the early stage of training. As the training progresses and the accuracy of the model output improves, we incorporate an increasing amount of data for processing. This helps to mitigate the interference of model prediction noise on domain adaptation.

In addition, in the target domain, for some categories with similar appearances, samples belonging to these categories are prone to being classified into incorrect but similar categories. This poses a certain level of difficulty for the model to achieve accurate classification. Therefore, we conduct additional optimization on these samples. Herein, we first screen out these indistinguishable similar categories, terming them *hard categories*; subsequently, we perform a further analysis and refinement on the samples affiliated with these categories, designating them as *hard samples*. To identify these hard categories, it is necessary to first count the category pairs with high similarity in the target domain. However, the category relationships in the target domain obtained from the target model during the early stage of training are prone to noise. To this end, we draw on the category relationships in the source domain, which can be obtained through the source classifier. Since the source model has been trained with labeled data, its category relationships can serve as a highly valuable reference for those in the target domain. Consequently, in order to alleviate the noise of the target model and the shift between the source domain and the target domain, we integrate the accurate category relationships of the source domain with the knowledge of the target domain. This approach aims to construct an improved category relationship that matches the actual data style of the target domain. Based on this, we can identify similar category pairs and then select the corresponding hard samples. We perform data augmentation on these samples to obtain more category information about the hard samples. A novel Gini confidence-based weighting strategy is adopted for these augmented samples to guide the training of the target model.

In conclusion, our contribution is mainly divided into the following three points. (1) We propose a progressive optimization framework for refining the data distribution in the target domain. (2) We introduce a novel strategy for screening hard categories in the target domain. (3) We present a novel weighting strategy for augmented samples to help correct the target model's predictions on hard samples.

The remainder of this paper is organized as follows. Section 2 provides a comprehensive review of the existing literature in the field of domain adaptation. Section 3 elaborates on the implementation details of the proposed method, including the optimization of inter-category and intra-category distributions, as well as the refinement of hard samples. Section 4 presents the experimental results of the method on three datasets, along with

in-depth discussion and analysis. Section 5 summarizes the key contributions of this study and outlines potential directions for future improvements.

2. Related Work

Domain adaptation aims to transfer knowledge from a source domain to a target domain with distribution discrepancy. Depending on whether the source data are accessible during adaptation, existing methods can be broadly categorized into unsupervised domain adaptation (UDA) and source-free domain adaptation (SFDA). UDA assumes that both labeled source data and unlabeled target data are available during training, while SFDA only relies on a pre-trained source model and unlabeled target data due to privacy or storage constraints. In this section, we first review representative UDA methods and then summarize recent advances in SFDA.

2.1. Unsupervised Domain Adaptation

Recent progress in unsupervised domain adaptation (UDA) has been significantly driven by models that leverage source-domain information. These methodologies can be broadly categorized into three distinct strategic frameworks.

The first approach centers on pseudo-labeling techniques, where core strategies encompass feature clustering, model-based prediction, and generative adversarial networks (GANs). PRPL [17] generates reliable target labels by selecting high-quality pre-trained features and adopting recurrent pseudo-labeling. PFAN [18] effectively screens reliable pseudo-labeled samples and mitigates the bias induced by erroneous labels via the alternating iterations of EHTS and APA. Representative studies also encompass [19–22]. Despite their effectiveness, their performance is highly sensitive to noisy pseudo labels and error accumulation.

The second paradigm focuses on distance-based alignments, aiming to bridge the domain gap by minimizing metrics that measure distributional divergence. Ref. [23] minimizes the mean and covariance discrepancies of features between the source domain and the target domain. The main works are [24–29]. However, most discrepancy-based methods focus on marginal alignment and may overlook class-conditional mismatches, leading to suboptimal semantic alignment.

The third strategy leverages adversarial learning to foster domain-invariant representations, typically achieved via feature-level alignment or data generation. DDAT [30] reduces the domain gap through dual-layer adversarial domain adaptation. Prominent studies within this direction also include [31–35]. Nevertheless, adversarial training is often unstable and may suffer from mode collapse or insufficient category-level alignment.

Despite their impressive results, these approaches rely heavily on the availability of source data—a requirement that is often impractical due to data privacy or storage constraints. Consequently, source-free domain adaptation has emerged as a more viable alternative for real-world deployment.

2.2. Source-Free Domain Adaptation

Due to the unavailability of source domain data, source-free domain adaptation (SFDA) poses a more significant challenge than conventional unsupervised domain adaptation, as it relies solely on a pre-trained model from the source domain and unlabeled data from the target domain. To tackle this challenge, three primary routes have been proposed.

The first one is the strategy based on pseudo-labels. It mainly assigns labels to target domain data through cluster centers or neighbor samples. Specifically, SHOT [3] updates the pseudo-labels of samples by iteratively refining the cluster centers, whereas NRC [4] facilitates more stable learning by analyzing distinct types of neighbors associated with

each sample. The relevant studies also include [1,2,5–8]. However, its performance largely depends on the initial prediction quality and is sensitive to noisy labels.

The second one uses contrastive learning strategy. It primarily seeks to minimize the discrepancy between positive sample pairs while maximizing that between negative sample pairs. HCL [12] proposes a historical contrastive learning framework that encompasses both instance-level and category-level. The main works are [9–11]. Nevertheless, its effectiveness relies on reliable pair construction, which may be compromised under severe domain shifts.

The third one is the strategy based on diversity and discriminability. Its aim is to optimize the distribution of the target domain data. For instance, AaD [13] posits that locally similar features should yield consistent predictions (attraction), whereas dissimilar features should produce divergent predictions (dispersion). Certain methods are also considered to have employed this route, including [3,4,11,14–16]. However, these methods typically rely on global distribution regularization and may overlook the negative impact of prediction noise.

Currently, there is still relatively limited and rather rudimentary research on the distribution optimization of the target domain in the area of source-free domain adaptation. Considering that the process of domain adaptation is often disrupted by prediction noise, we propose a progressive optimization approach for the distribution of the target domain to mitigate the negative impacts brought about by prediction noise. Specifically, we conduct optimizations from both inter-category and intra-category perspectives. Secondly, there are always some hard samples in the target domain, which are difficult to be accurately classified because their characteristics align closely with those of multiple categories. This is often due to the small differences between several categories. Therefore, we consider adopting a novel strategy to filter out similar categories and optimize the samples belonging to these categories to help them be accurately classified into the correct categories.

3. Method

3.1. Preliminaries

In the source-free domain adaptation setting, a pre-trained source model—consisting of a feature extractor and a classifier—is initially trained using labeled data from the source domain. The objective is to adapt this model to the target domain by leveraging the unlabeled target data $\{x_t^i\}_{i=1}^{N_t}$, where N_t denotes the number of samples in the target domain, through a knowledge transfer mechanism. Our approach adopts the mean-teacher framework, which includes a teacher model and a student model, both initialized using the parameters of the source model. The feature extractor and classifier of the teacher model are represented as f_{tc} and g_{tc} , respectively, while those of the student model are denoted as f_{st} and g_{st} . For an input sample x^i , the student model first generates a feature representation $z_{st}^i = f_{st}(x^i) \in \mathbb{R}^d$, with d being the dimensionality of the feature space. This feature is then passed through the classifier to produce a probability distribution, $p_{st}^i = \delta(g_{st}(z_{st}^i)) \in \mathbb{R}^K$, where δ represents the softmax function, and K is the number of categories. The final predicted label is subsequently obtained as $\hat{y}_{st}^i = \text{argmax}(p_{st}^i)$, where $\text{argmax}(\cdot)$ returns the index corresponding to the highest probability in the output vector. *st* is the abbreviation for “student”, and *tc* is the abbreviation for “teacher”. Ultimately, the student model serves as the final adapted target model.

3.2. Overview

The proposed methodology consists of two components: the progressive distribution optimization module and the hard sample optimization module.

To alleviate the interference caused by prediction noise in the early training phase, we introduce a progressive strategy to refine the target domain distribution by dispersing the

inter-category distribution and compressing the intra-category distribution. Specifically, inter-category separation is enhanced by prioritizing category pairs with high disparity, while intra-category compactness is improved by focusing on sample pairs of the same category that exhibit high similarity. The purpose of this phased optimization mechanism is to enable robust adaptation under conditions of unreliable predictions. As illustrated in Figure 1.

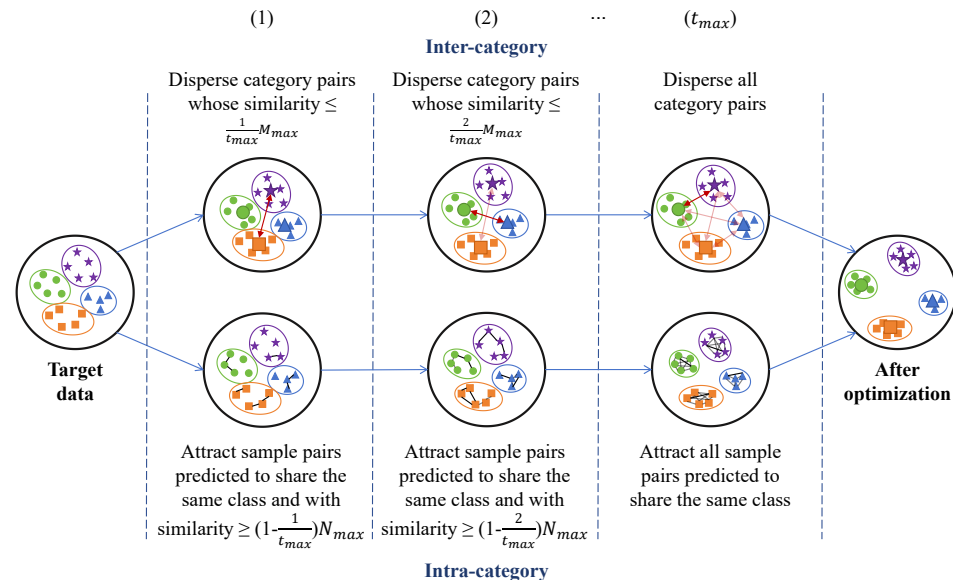


Figure 1. The framework of progressive optimization of inter-category and intra-category distributions in the target domain. The red double arrows denote an increase in distance, whereas the black lines denote a decrease in distance. The dark arrows denote newly added optimizations in the current iteration, while the light arrows represent optimizations implemented in previous iterations.

Furthermore, due to high visual similarity among certain categories, samples from these categories are prone to misclassification. To mitigate this issue, we identify easily confused categories through knowledge fusion between the source and target domains. We then apply data augmentation to samples within these categories. For augmented samples, a novel weighting strategy is employed to guide and rectify the target model's predictions. As presented in Figure 2.

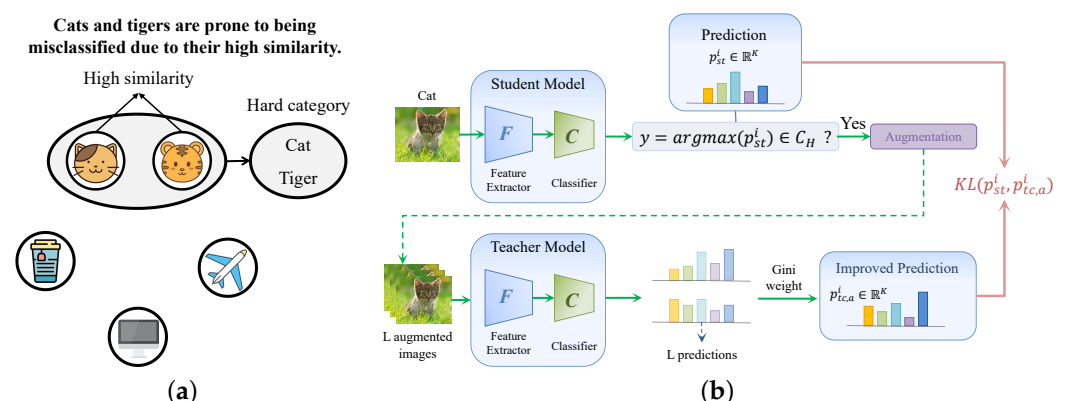


Figure 2. The framework for screening hard categories (a) and optimizing hard samples (b). C_H denotes the set of hard categories. p_{st}^i represents the prediction result of the student model. $p_{tc,a}^i$ is the improved prediction generated by the teacher model.

3.3. Progressively Optimize Distributions

In the target domain, the data distribution directly affects the performance of the target model. It can be argued that, when inter-category distributions exhibit high distin-

guishability and intra-category samples demonstrate strong compactness, the classifier is better able to derive clear and accurate decision boundaries for target domain data. Consequently, optimizing the data distribution of the target domain is of great necessity. Nevertheless, owing to the presence of domain shift, particularly during the early phase of training, the predictions of the target model regarding the target data frequently exhibit noise and lack accuracy. Relying solely on these predictions to partition the target domain data may weaken the effectiveness of domain adaptation. Consequently, we implement a progressive strategy to optimize the data distribution within the target domain. This strategy encompasses two key aspects: enhancing inter-category separation and promoting intra-category compactness.

3.3.1. Inter-Category Optimization

To optimize across categories, it is first necessary to obtain the representation of each category. Suppose that each category within a batch is represented by its average prediction, and the representation for category k is

$$p^k = \frac{\sum_{i=1}^{N_b} I(\hat{y}_{st}^i = k) \cdot p_{st}^i}{\sum_{i=1}^{N_b} I(\hat{y}_{st}^i = k)} \in \mathbb{R}^K, \quad (1)$$

where $p_{st}^i \in \mathbb{R}^K$ and $\hat{y}_{st}^i$ denote the prediction vector and predicted label of the student model for sample i , respectively; st is the abbreviation for “student”, K represents the number of categories in the target domain, N_b refers to the mini-batch size, and $I(\cdot)$ is the indicator function.

In the early stage of training, the presence of prediction noise may introduce biases into the derived category representations—particularly for easily confused categories. Categories with substantial disparities can be separated with greater accuracy, whereas accurately distinguishing between similar categories presents a more pronounced challenge. Specifically, the classification boundaries for the former are better defined, while those for the latter remain relatively ambiguous. The undifferentiated dispersion of categories can give rise to negative transfer. Therefore, we initially focus on processing category pairs with more distinct differences. As training progresses, category representations gradually become more accurate, and then more categories are processed step by step, until all categories are optimized.

Specifically, the similarity between categories is represented as $\text{sim}(a, b) = a \cdot b^T$, where both a and b denote category representations.

In each training iteration, we set a threshold, h_1 , to determine whether to process a particular category. Specifically,

$$h_1 = \frac{t}{t_{max}} \cdot M_{max}, \quad (2)$$

where t denotes the current number of training iterations, t_{max} represents the maximum number of training iterations, and $\frac{t}{t_{max}} \in (0, 1]$. M_{max} stands for the maximum similarity among all category pairs, serving to achieve a correspondence between training time and similarity, and it is updated throughout the training process.

Remark: The rationale for adopting a linear threshold schedule is twofold: (1) Training stability: As training progresses, the target model’s accuracy improves relatively steadily. A linear schedule smoothly and evenly relaxes the threshold, allowing for a consistent, gradual inclusion of category pairs into the optimization process. This avoids sudden, drastic shifts in the loss landscape that could destabilize training. (2) Simplicity and robustness:

While non-linear schedules could theoretically be applied, a linear schedule is intuitive and effectively achieves progressive optimization without introducing additional complex hyperparameters that would require heavy tuning for different datasets.

When the divergence between two categories is relatively large, that is, the similarity does not exceed the threshold h_1 , we increase the distance between them. Category pairs with a similarity greater than the threshold will not be processed. This is to avoid mistakenly moving certain samples away from their own categories.

$$U(A, B) = \begin{cases} 1 & \text{if } \text{sim}(P^A, P^B) \leq h_1 \\ 0 & \text{if } \text{sim}(P^A, P^B) > h_1 \end{cases} \quad (3)$$

when $U(A, B) = 1$, we perform optimization on categories A and B .

In summary, we optimize the relationship between categories by reducing the statistics of similarity among the categories that meet the conditions, i.e.,

$$Z_{inter} = \frac{\sum_{A=1}^K \sum_{B=1, B \neq A}^K U(A, B) \text{sim}(P^A, P^B)}{\sum_{A=1}^K \sum_{B=1, B \neq A}^K U(A, B)}. \quad (4)$$

In a progressive manner, we increase the distances between different categories; that is, reducing their similarity and thereby enhancing their distinguishability.

3.3.2. Intra-Category Optimization

Our objective is to minimize the distance among samples of the same category in the target domain, thereby enhancing the compactness of their distribution. However, certain samples—despite being predicted as belonging to the same category—may in fact originate from different categories. Minimizing their distances without careful consideration could also induce negative transfer. To reduce this risk, we prioritize optimizing those sample pairs that are not only predicted to be of the same category but also have a high degree of similarity, thereby mitigating the impact of incorrect predictions. As the classifier's accuracy improves during training, an increasing number of same-category samples are progressively incorporated into the optimization process.

Analogous to the inter-category training, in this case, we establish a threshold, h_2 , for the optimization of intra-category samples. Specifically,

$$h_2 = (1 - \frac{t}{t_{max}}) \cdot N_{max}, \quad (5)$$

where N_{max} denotes the maximum value of the similarities of all sample pairs and is updated iteratively throughout the training process. The rationale for employing linear threshold scheduling herein aligns with that for threshold h_1 .

When two samples are predicted to be of the same category and their similarity is not less than the threshold h_2 , we optimize their relationship. When a sample pair is predicted to be of different categories or their similarity is less than the threshold, no processing is carried out on them to prevent negative optimization.

$$V(a, b) = \begin{cases} 1 & \text{if } \text{sim}(p^a, p^b) \geq h_2 \text{ and } \hat{y}_{st}^a = \hat{y}_{st}^b \\ 0 & \text{if } \text{sim}(p^a, p^b) < h_2 \text{ or } \hat{y}_{st}^a \neq \hat{y}_{st}^b \end{cases} \quad (6)$$

when $V(a, b) = 1$, we perform optimization on samples a and b .

In conclusion, we enhance the similarity of intra-category sample pairs that satisfy the conditions to optimize their relationships.

$$Z_{intra} = \frac{\sum_{a=1}^{N_b} \sum_{b=1, b \neq a}^{N_b} V(a, b) \text{sim}(p_{st}^a, p_{st}^b)}{\sum_{a=1}^{N_b} \sum_{b=1, b \neq a}^{N_b} V(a, b)}. \quad (7)$$

This is implemented in a progressive fashion to decrease the distances between intra-category samples, that is, increasing their similarity and thereby rendering their distribution more compact.

Consequently, the overall optimization function with respect to the inter-category and intra-category distributions of the target domain is

$$L_{ii} = \frac{Z_{inter} + e}{Z_{intra} + e}, \quad (8)$$

where $e = 10^{-8}$, which is employed to address the cases where the numerator or the denominator is zero.

3.4. Optimize Hard Samples

In the target domain, certain samples are misclassified by the classifier due to the high similarity between their true categories and the assigned incorrect categories. This inherent similarity introduces ambiguity in classification, making it challenging for the model to produce accurate predictions. Thus, addressing this issue is imperative, as it also contributes to the aforementioned optimization of the target domain distribution.

3.4.1. Screening Hard Categories and Samples

It is necessary to first identify category pairs with high similarity by systematically computing the pairwise similarities across all categories. However, in the early stages of training, the category relationships of the target domain obtained are subject to noise. In contrast, the source domain model is trained on the labeled source domain, during which it acquires accurate prior category information about the source domain. This information serves as a crucial clue for the target domain. Despite the existence of domain shift between the source domain and the target domain, the category relationships still exhibit relevance. For example, cats and tigers share a relatively high degree of similarity in both domains, while cats and tables maintain a relatively low similarity. Therefore, the category relationships in the source domain hold significant reference value for the unlabeled target domain. Consequently, after deriving the noisy category relationships of the target domain from the student model, their confidence levels are analyzed. We retain parts that are certain; for uncertain parts, we use the category relationships of the source domain for compensation. As the training proceeds, the accuracy of the target model improves, and the confidence levels of the category relationships in the target domain also increase steadily.

Here, the average Gini confidence of the student model's prediction results for the target data is employed to represent the confidence level of the target domain category relationships. The Gini confidence of the prediction results of sample i obtained from the student model is denoted as

$$C(p_{st}^i) = 1 - G(p_{st}^i), \quad (9)$$

where $G(p_{st}^i)$ is the Gini impurity of p_{st}^i , which is specifically represented as $G(p_{st}^i) = 1 - \sum_{j=1}^K (p_{st,j}^i)^2$, and j represents the index within the p_{st}^i .

Thus, the confidence of the category relationships in the target domain is denoted as

$$\bar{C}(p_{st}) = \frac{1}{N_t} \sum_{i=1}^{N_t} C(p_{st}^i). \quad (10)$$

Suppose that R_{st} denotes the category relationship derived by the student model, which encapsulates domain-specific characteristics of the target distribution, i.e.,

$$R_{st} = \cos(P_{st}, P_{st}^T) \in \mathbb{R}^{K \times K}, \quad (11)$$

where $\cos(\cdot, \cdot)$ denotes the cosine similarity function, and $P_{st} \in \mathbb{R}^{K \times K}$ represents the average predictions of the K categories in the target domain output by the student model. Suppose R_s denotes the category relationship of the source domain, and its calculation method is

$$R_s = \cos(W, W^T) \in \mathbb{R}^{K \times K}, \quad (12)$$

where $W \in \mathbb{R}^{K \times d}$ denotes the weight of the source classifier.

By integrating R_s with R_{st} , an improved category relationship is obtained, which effectively preserves the prior knowledge of the source domain while achieving better alignment with the data style of the target domain. And the parts of the student model that are determined are retained, while the uncertain parts are supplemented through the category relationships in the source domain. The improved category relationship is specifically represented as

$$R_I = (1 - \bar{C}(p_{st})) \cdot R_s + \bar{C}(p_{st}) \cdot R_{st}. \quad (13)$$

Remark: Despite the presence of a domain shift between the source and target domains, the category relationship of the source domain—serving as a supplement to the uncertain component of the target domain category relationship—typically provides valuable reference information for the latter. In extreme scenarios where the source and target domains differ drastically, the source domain category relationship only replaces the uncertain part of the target domain category relationship. Even without such supplementation, this uncertain component would inherently be noise. Thus, even under significant domain shift, the negative impact it introduces to the improved category relationship and overall training remains relatively minor. On the whole, the source domain category relationship primarily plays a facilitative role.

After obtaining the improved category relationship R_I , we identify the s pairs of categories with the highest similarity from the category relationship matrix, that is, $\text{top}_s(R_I)$. These categories are considered the most difficult to distinguish in the target domain, namely *hard categories*. Then, we select the samples assigned to these categories by the student classifier and further optimize these samples, which are identified as *hard samples*.

3.4.2. Optimize Hard Samples

We conduct multiple data augmentation on hard samples to capture more abundant category information. Specifically, a simple yet effective data augmentation strategy is employed: L perturbations are sampled from the Gaussian distribution $\mathcal{N}(0, 0.1)$, and each is added to the original data, generating L augmented samples that are structurally distinct from yet semantically consistent with the original data. Subsequently, the teacher model is employed to predict on these L new data. After calculating the weighted average, we obtain $p_{tc,a}$, which is used to align the results of the student model. The detailed operation is as follows.

Suppose x_h is a hard sample belonging to the hard category, and x_a^l ($l \in (1, L)$) is the l -th augmented sample generated for x_h . The prediction result of the teacher model for x_a^l is $p_{tc,a}^l = \delta(g_{tc}(f_{tc}(x_a^l))) \in \mathbb{R}^K$, where tc is the abbreviation for “teacher”.

The improved prediction $p_{tc,a}$ for sample x_h is derived by applying weighted aggregation to the prediction results of L augmented samples, i.e.,

$$p_{tc,a} = \sum_{l=1}^L w^l p_{tc,a}^l \in \mathbb{R}^K, \quad (14)$$

where w^l denotes the weight assigned to the l -th augmented sample. And

$$w^l = \frac{C(p_{tc,a}^l)}{\sum_{i=1}^L C(p_{tc,a}^i)}, \quad (15)$$

and $C(\cdot)$ denotes the Gini confidence of the vector inside the parentheses.

The improved prediction $p_{tc,a}$ is employed to calibrate the outcomes of the student model. Suppose that the number of hard samples in the target domain is n , and the corresponding loss for these hard samples is

$$L_{hard} = \sum_{i=1}^n KL(p_{st}^i \parallel p_{tc,a}^i), \quad (16)$$

where $KL(\cdot \parallel \cdot)$ denotes the Kullback–Leibler (KL) divergence function.

3.5. Overall Loss

Our approach comprises two distinct components: (1) **Progressive Optimization Distribution**; (2) **Optimization of Hard Samples**. Therefore, it is denominated as **POSTAL**. The overall loss function of the target model is

$$L_{POSTAL} = L_{baseline} + \alpha L_{ii} + \beta L_{hard}, \quad (17)$$

where α and β are trade-off hyperparameters. $L_{baseline}$ is the loss of baseline (i.e., SHOT or AaD).

Updating the teacher model parameters. Upon the completion of each iteration, the student model θ_{st} is updated via backpropagation. In contrast, the teacher model θ_{tc} performs a momentum-based update from the student model using the Exponential Moving Average (EMA) method, i.e.,

$$\theta_{tc}^t = \sigma \theta_{tc}^{t-1} + (1 - \sigma) \theta_{st}^t, \quad (18)$$

where θ_{tc} and θ_{st} are the parameters of the teacher and student models, respectively, and t denotes the training time. σ is a hyperparameter set to 0.9.

3.6. Algorithm

Algorithm 1 describes the process of our method. The input comprises unlabeled target domain data $\mathcal{X}_t = \{x_t\}_{i=1}^{N_t}$ and the maximum number of training iterations t_{max} . Line 1 initializes the teacher and student models. Lines 2–11 outline the training iteration process of the algorithm: Lines 3–5 detail the procedure for computing the loss L_{ii} ; Lines 6–9 elaborate on the process of deriving the loss L_{hard} ; Line 10 specifies the training and update mechanism for the student model; Line 11 describes the parameter update strategy for the teacher model. The final output is the target domain model (i.e., the student model), which consists of a feature extractor, f_{st} , and a classifier, g_{st} .

Algorithm 1: POSTAL algorithm for closed-set SFDA

Input: Unlabeled target domain $\mathcal{X}_t = \{x_t\}_{i=1}^{N_t}$, maximum iterations t_{max} .
Output: Target feature extractor f_{st} , target classifier g_{st} .

- 1 Initialize the teacher and student models with the source domain model;
- 2 **for** $t = 1 \rightarrow t_{max}$ **do**
- 3 To mitigate the interference of prediction noise, optimization thresholds h_1 (Equation (2)) and h_2 (Equation (5)) are respectively introduced, corresponding to the optimization conditions $U(A, B)$ (Equation (3)) and $V(a, b)$ (Equation (6));
- 4 Calculate the inter-category similarity Z_{inter} (Equation (4)) and the intra-category sample similarity Z_{intra} (Equation (7)) that meet the optimization conditions;
- 5 Calculate the loss function L_{ii} for progressively optimizing the inter-category and intra-category distributions via Equation (8);
- 6 Calculate the improved category relationship R_I based on the average prediction confidence $\bar{C}(p_{st})$ from the source domain category relationship R_s and the target domain category relationship R_{st} via Equation (13);
- 7 Screening the hard categories based on $\text{top}_s(R_I)$;
- 8 For hard samples in the hard categories, data augmentation is applied, and the teacher model generates optimized prediction, $p_{tc,a}$, via Equation (14) for the augmented data;
- 9 Obtain the loss L_{hard} via Equation (16) to supervise the training of the student model with $p_{tc,a}$;
- 10 The student model is trained with the total loss L_{POSTAL} (Equation (17)), and its parameters are updated via gradient backpropagation;
- 11 Update teacher model parameters through EMA via Equation (18).
- 12 **return** f_{st}, g_{st}

4. Experiments

4.1. Settings

Datasets. We conducted image classification experiments on three benchmark datasets: Office-31 [36], ImageCLEF [37] and DomainNet [38]. **Office-31** contains three domains—Amazon (A), DSLR (D) and Webcam (W)—with 2817, 498, and 795 images, respectively, totaling 31 categories. **ImageCLEF** contains three domains: Caltech-256 (C), ImageNet ILSVRC 2012 (I) and Pascal VOC 2012 (P). Each domain consists of 12 categories, with 50 images contained in each categories. **DomainNet** contains images from six different domains, amounting to roughly 600,000 images in total. To evaluate the effectiveness of the proposed method, we conduct a series of experiments on seven domain adaptation tasks.

Implementation Details. To ensure a fair comparison with existing approaches, we employ a ResNet-50 as the backbone network across all datasets. Following the setup of SHOT [3] and AaD [13], the model adopts an identical architecture featuring a fully connected layer, followed by Batch Normalization [39], and another fully connected layer with weight normalization [40]. For optimization, SGD is used with a momentum value of 0.9 and a batch size of 64 on all datasets. The initial learning rate is set to 1×10^{-3} for all layers, while the last two fully connected layers use a higher learning rate of 1×10^{-2} . For the Office-31 and ImageCLEF datasets, we conducted training for 50 epochs; for the DomainNet dataset, the training was performed for 30 epochs.

The learning rate ζ is adjusted dynamically according to the formula $\zeta = \zeta_0 \cdot (1 + 10 \cdot \frac{t}{t_{max}})^{-\tau}$, consistent with the methodologies in [41,42]. Here, the hyperparameter τ governs the decay rate and is fixed at 1. For all our experiments, we set $\alpha = 1$, $\beta = 0.1$ and $L = 12$.

All experiments were conducted on a workstation equipped with an NVIDIA RTX 3090 GPU. The proposed model was implemented using the PyTorch (torch 2.4.1) deep learning framework. The training process was performed with CUDA (12.2) acceleration, and common scientific computing libraries including NumPy (1.23.2) and SciPy (1.10.1) were used for data processing.

Competitors. To prove the effectiveness of our method, we compare POSTAL with the state-of-the-art SFDA methods, including SHOT [3], NRC [4], G-SFDA [43], TPDS [44], C-SFDA [11], UI-SFDA [14], CADRN [45], NVC-ELR [8], DPL [7], OCRCI [5], PFC [6] and AaD [13].

4.2. Results of Object Recognition

Results on Office-31. The comparison result on Office-31 dataset between our method and other SFDA methods is shown in Table 1. Our method achieves an accuracy of 91.5%. A further boost in average accuracy is observed with our approach, which surpasses AaD and SHOT by 1.6% and 1.4%, respectively. The accuracy across all six tasks are not lower than the original ones. Particularly on the A→W task, POSTAL surpasses AaD with a significant 4.3% increase in accuracy.

Table 1. Accuracy (%) on Office-31 for source-free domain adaptation. “SF” stands for source-free. The best results are highlighted in **bold** and the second-best results are highlighted in underline.

Method	SF	A→D	A→W	D→A	D→W	W→A	W→D	Avg.
C-SFDA [11]	✓	96.2	93.9	77.3	98.8	<u>77.9</u>	99.7	90.5
UI-SFDA [14]	✓	95.3	94.2	76.4	98.3	77.5	<u>99.9</u>	90.3
CADRN [45]	✓	95.6	<u>94.7</u>	76.9	98.6	77.7	100.0	90.6
DPL [7]	✓	<u>97.0</u>	93.5	77.0	98.6	78.0	100.0	<u>90.7</u>
NVC-ELR [8]	✓	95.8	91.7	<u>77.2</u>	98.9	77.2	100.0	90.1
SHOT [3]	✓	94.0	90.1	74.7	98.4	74.3	<u>99.9</u>	88.6
POSTAL + SHOT	✓	95.8	93.2	75.6	99.2	76.0	100.0	90.0 (+1.4)
AaD [13]	✓	96.4	92.1	75.0	<u>99.1</u>	76.5	100.0	89.9
POSTAL + AaD	✓	98.2	96.4	77.1	<u>99.1</u>	78.0	100.0	91.5 (+1.6)

Results on ImageCLEF. The result on ImageCLEF dataset is shown in Table 2. We achieve an accuracy of 90.9%. Our method outperforms AaD and SHOT by margins of 1.1% and 1.0% in average accuracy, respectively, demonstrating a clear performance gain over these baselines. Notably, our method maintains performance consistency, with accuracies across all six tasks exceeding those of the original. Specifically, for the task of I→P, our accuracy is 1.5% higher than that of AaD.

Table 2. Accuracy (%) on ImageCLEF for source-free domain adaptation. “SF” stands for source-free. The best results are highlighted in **bold** and the second-best results are highlighted in underline.

Method	SF	C→I	C→P	I→C	I→P	P→C	P→I	Avg.
G-SFDA [43]	✓	92.0	79.5	96.7	79.2	95.5	92.5	89.2
NRC [4]	✓	93.3	78.0	97.2	79.2	96.7	92.3	89.4
OCRCI [5]	✓	<u>94.3</u>	<u>80.0</u>	<u>97.5</u>	<u>80.8</u>	<u>97.7</u>	<u>93.7</u>	<u>90.7</u>
SHOT [3]	✓	92.3	79.5	95.8	78.7	95.3	92.7	89.1
POSTAL + SHOT	✓	93.2	80.3	<u>97.5</u>	79.2	96.7	93.5	90.1 (+1.0)
AaD [13]	✓	93.8	78.7	96.8	79.5	96.7	93.2	89.8
POSTAL + AaD	✓	94.7	79.7	97.8	81.0	97.8	94.3	90.9 (+1.1)

Results on DomainNet. Our methods yields competitive results on DomainNet dataset compared with other methods, which is shown in Table 3. Our method achieves an

accuracy of 71.0%. Relative to AaD and SHOT, our proposed technique yields an additional 1.4% improvement in terms of average classification accuracy. And the accuracy on all seven tasks is no less than the original. In the specific P→C task, our model significantly outperforms SHOT, delivering a substantial 4.3% margin in terms of accuracy.

Table 3. Accuracy (%) on DomainNet for source-free domain adaptation. “SF” stands for source-free. The best results are highlighted in **bold** and the second-best results are highlighted in underline.

Method	SF	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Avg.
C-SFDA [11]	✓	70.8	<u>71.1</u>	68.5	62.1	67.4	62.7	80.4	69.0
TPDS [44]	✓	66.4	67.0	65.6	59.8	64.3	61.5	79.0	66.2
CADRN [45]	✓	69.2	68.8	70.3	61.6	66.7	60.7	81.2	68.4
PFC [6]	✓	<u>71.9</u>	70.3	<u>72.7</u>	62.3	68.5	61.0	<u>83.1</u>	<u>70.0</u>
SHOT [3]	✓	67.7	68.4	66.9	60.1	66.1	59.9	80.8	67.1
POSTAL + SHOT	✓	69.7	68.7	71.2	62.2	66.3	59.9	81.4	68.5 (+1.4)
AaD [13]	✓	71.0	70.3	71.8	<u>62.5</u>	<u>69.0</u>	60.4	82.3	69.6
POSTAL + AaD	✓	73.2	71.2	73.0	63.9	69.8	<u>62.3</u>	83.3	71.0 (+1.4)

From Tables 1–3, the following three conclusions can be drawn. First, when our method is integrated with the baseline methods, it yields a significant improvement in model accuracy over the original performance. This is attributed to two key aspects: our progressive optimization of inter-category and intra-category distributions in the target domain mitigates the interference of noise on distribution optimization, while our hard sample refinement strategy reduces the negative impact of such samples on model accuracy. Additionally, this integration validates the plug-and-play nature of our method. Second, our method exhibits favorable performance across the three diverse datasets, indicating its compatibility and generalization capability to different datasets. Third, despite the fact that the source-free domain adaptation process does not access the source domain data, promising results are still achieved on the target domain—this holds considerable application value for scenarios where access to source domain data is constrained.

4.3. Model Analysis

Parameter sensitivity. We conduct a sensitivity analysis to evaluate the impact of hyperparameters α and β within our framework. Specifically, α and β regulate the trade-off weights for the losses L_{ii} and L_{hard} , respectively. The resulting performance on the Office-31 D→A task is illustrated in Figure 3.

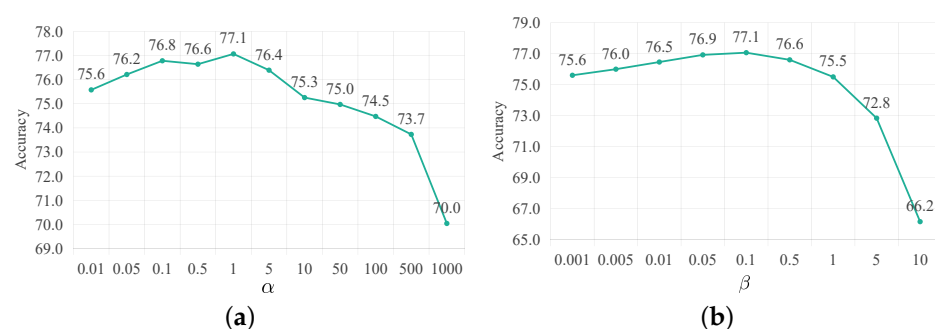


Figure 3. Parametric sensitivity analysis of α and β .

We vary α from 0.01 to 1000 while keeping β fixed at 0.1 (Figure 3a) and similarly range β from 0.001 to 10 with α fixed at 1 (Figure 3b). As illustrated, the model attains its peak accuracy at the configuration of $\alpha = 1$ and $\beta = 0.1$. We employed identical hyperparameter configurations across three distinct datasets, thereby demonstrating robust cross-dataset generalizability. This allows the default hyperparameters to be reliably transferred to novel target domains, obviating the need for validation data.

Ablation study. To gain deeper insights into the contribution of various modules, we evaluate two architectural variants of our framework on the DomainNet dataset. The results, conducted under the SFDA setting, are summarized in Table 4. Specifically, we remove L_{ii} and L_{hard} from the full model, respectively. L_{AaD} is the loss term of the baseline method AaD. Upon removing the loss L_{ii} , the accuracy of the model stands at 70.6%. After removing the loss L_{hard} , the model's accuracy is 70.7%. The accuracy reaches its zenith at 71.0% upon the fusion of both components. This evidence confirms that each constituent contributes positively to the adaptation process, with their joint implementation being indispensable for optimal performance.

Table 4. Ablation study on DomainNet dataset. The w/o is short for without.

Method	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Avg.
L_{AaD}	71.0	70.3	71.8	62.5	69.0	60.4	82.3	69.6
w/o L_{ii}	72.7	70.9	72.9	63.8	69.3	61.7	83.0	70.6
w/o L_{Hard}	73.1	71.1	72.8	63.4	69.5	61.8	82.9	70.7
Full Model	73.2	71.2	73.0	63.9	69.8	62.3	83.3	71.0

Accuracy analysis of hard categories. Figure 4 presents our analysis of the accuracy for hard categories in the $W \rightarrow A$ task on the Office-31 dataset. The orange line denotes the variation of model accuracy during training. The green line represents the accuracy of samples from hard categories identified via our strategy. The blue line indicates the prediction accuracy of samples from remaining categories after excluding hard categories from the full dataset. Key observations are as follows: The accuracy of samples in hard categories is substantially lower than the overall dataset accuracy, with a gap of approximately 20%. After removing samples of hard categories, the overall dataset accuracy improves significantly by around 6%. Thus, our method effectively identifies hard categories in the dataset, and optimizing these categories is critical for enhancing the overall prediction accuracy.

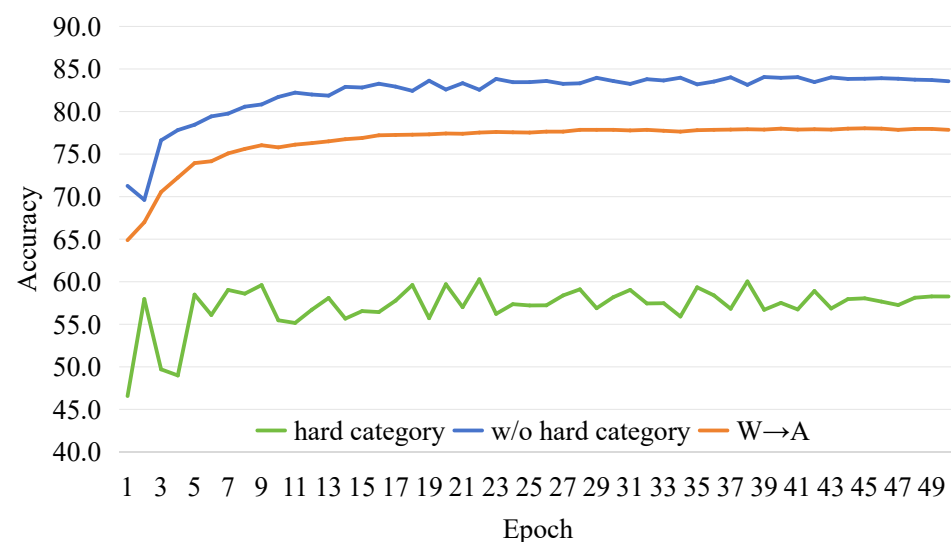
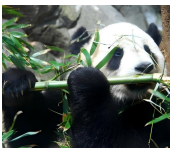


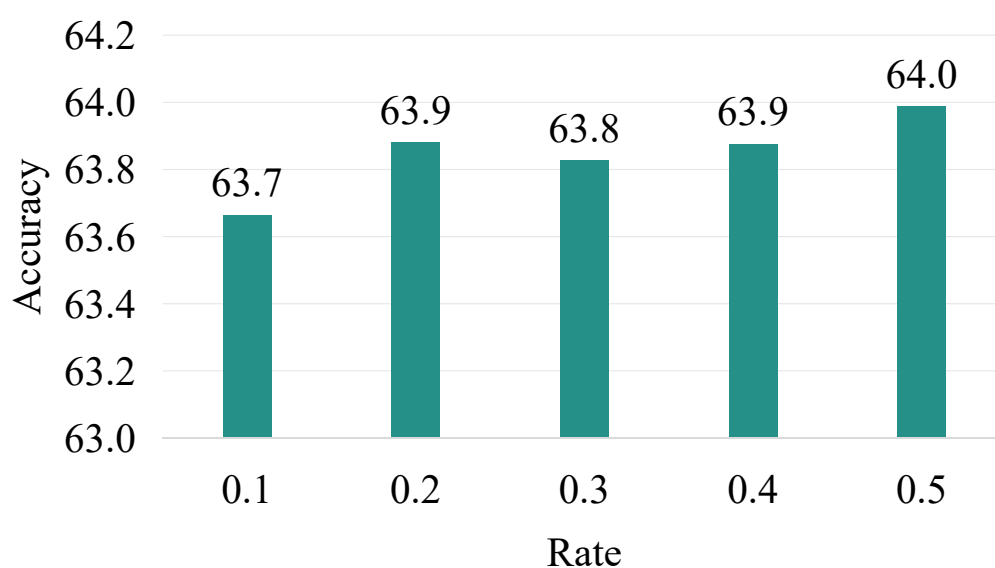
Figure 4. Accuracy analysis of hard categories.

Correcting predictions of hard samples. Table 5 shows examples of our method correcting the prediction results of hard samples. Empirical observation reveals that the original predicted categories exhibit high similarity to the corrected ones for these instances. Our method can effectively correct the original wrong predictions of the target model. This indicates the effectiveness of our method for correcting the prediction results of hard samples.

Table 5. Analysis of corrections for hard samples.

	1	2	3	4	5	6
Hard Sample						
Original Prediction (Wrong)	flower	bear	swan	peas	dolphin	laptop
After Optimization (Correct)	onion	panda	bird	asparagus	penguin	computer

Analysis of the number of hard categories. Figure 5 presents an analysis of the impact of the number of hard category pairs on model performance in the C→S task of the DomainNet dataset. The number of hard category pairs is selected proportionally based on the total number of categories in the dataset. Results indicate that the model's accuracy consistently remains above 63.6%. Specifically, when the number of hard category pairs is set to 25 (calculated as 126×0.2), the model achieves an accuracy of 63.9%, corresponding to a range of [26, 50] individual hard categories (approximately 20–40% of the total category number). When the number of hard category pairs is 63 (calculated as 126×0.5), the model accuracy reaches 64.0%, corresponding to a range of [64, 126] individual hard categories (approximately 51–100% of the total category number). Upon comprehensive consideration of computational cost and accuracy, 25 is selected as the number of hard category pairs. For all datasets, the number of hard category pairs is determined using the same protocol: multiplying the total number of categories in the dataset by 0.2 and rounding down. Specifically, Office-31 yields 6 (31×0.2), ImageCLEF yields 2 (12×0.2), and DomainNet yields 25 (126×0.2).

**Figure 5.** Analysis of the number of hard categories.

Analysis of data augmentation frequency. Figure 6 shows an analysis of the number of data augmentations for hard samples on the W→A task of the office-31 dataset. As illustrated in the results, the model's accuracy plateaus above 77.8% once the number of

data augmentations L reaches 10. The peak performance of 78.0% is attained at $L = 12$. Consequently, we fix $L = 12$ as the default augmentation frequency for hard samples in our experiments.

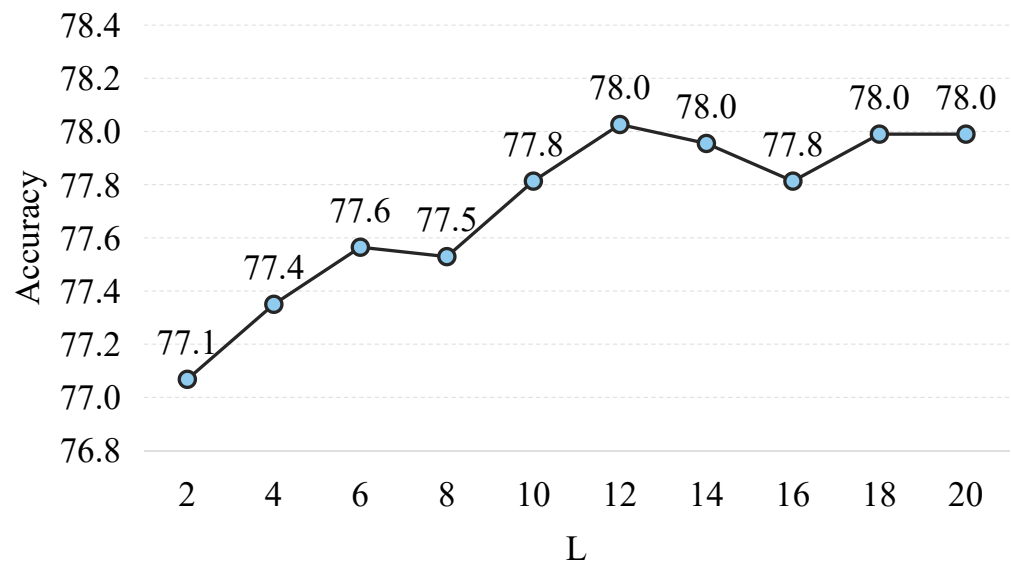


Figure 6. Analysis of the number of data augmentations for hard samples.

Comparison of running time and storage space. Table 6 presents a comparative analysis of the runtime and storage consumption between our proposed method POSTAL and other baseline methods. Our method POSTAL leverages AaD as the embedding baseline. It introduces only marginal increases in overhead. However, its resource consumption remains substantially lower than that of C-SFDA. Furthermore, POSTAL achieves superior accuracy compared to both AaD and C-SFDA. These results confirm that our method strikes a favorable balance between performance and resource consumption, making it suitable for practical use.

Table 6. Comparison of running time (s/epoch) and storage space (MB).

Method	A→D	A→W	D→A	D→W	W→A	W→D	Memory	Acc
C-SFDA	598.4	770.0	2492.3	776.3	2473.8	591.9	28,689	90.5
AaD	4.7	3.5	9.3	3.5	9.5	4.7	9458	89.9
Ours	7.2	9.3	34.4	9.5	32.4	7.5	9931	91.5

Training stability. Figure 7 illustrates the evolution of accuracy on the Office-31 dataset throughout the training process. It is evident that the performance exhibits a steady upward trend before reaching an asymptotic plateau, thereby substantiating the stability and convergence of our proposed method.

Feature visualization. To qualitatively assess the learned representations, we utilize t-SNE [46] to visualize the feature distributions for the P→C task on DomainNet. Figure 8 compares the source model, SHOT, AaD, and our proposed method. As illustrated, our approach yields significantly more compact clusters with superior inter-category separability compared to the baselines, thereby validating its effectiveness.

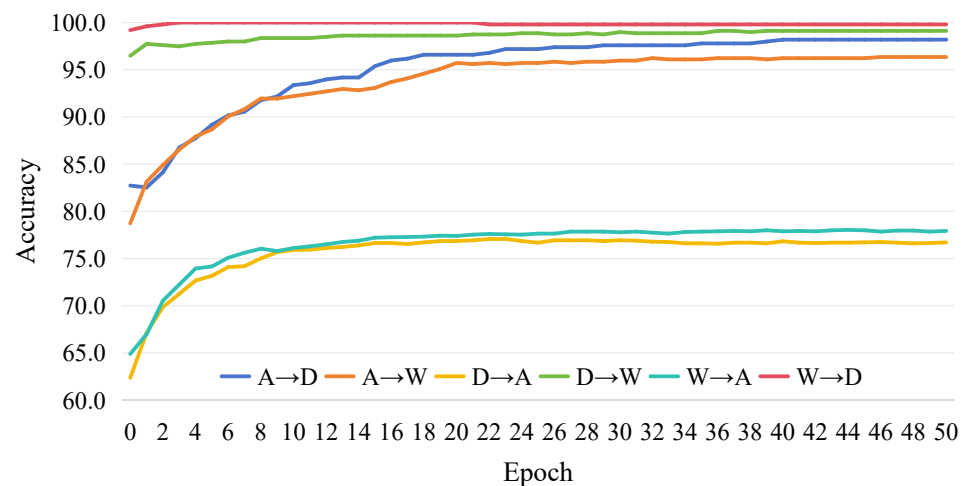


Figure 7. Training stability analysis on Office-31.

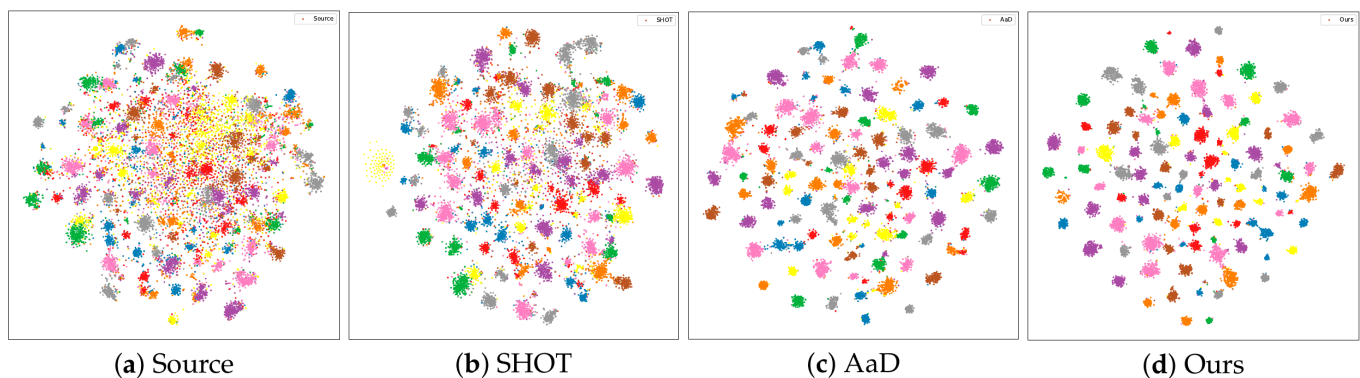


Figure 8. Feature visualizations of task $P \rightarrow C$ on DomainNet dataset by t-SNE [46].

4.4. Discussion

Our method enhances the performance of source-free domain adaptation by progressively optimizing inter-category and intra-category distributions, as well as refining predictions for hard samples. The experimental results presented above validate the effectiveness of our approach. However, the optimization of hard samples in our method still has limitations: while data augmentation can capture richer category information for hard samples, it remains insufficient for mining category information from samples with extremely ambiguous semantic cues, potentially failing to accurately capture their true category attributes. The work by [47]—a study focusing on few-shot learning—offers a promising solution: by analyzing feature maps across multiple scales, it synchronously captures high-level semantic information and low-level texture details, thereby enabling more accurate and comprehensive extraction of category information for hard samples. This strategy could also facilitate the optimization of both inter-category and intra-category distributions. This constitutes a key direction for our future work.

5. Conclusions

In this paper, we propose to tackle the source-free domain adaptation problem by leveraging progressive distribution optimization and novel correction of hard samples. Specifically, we propose a progressive strategy to optimize both inter-category and intra-category data distributions in the target domain, aiming to mitigate the adverse effects of model prediction noise. This enables more efficacious separation of distinct categories and tighter clustering of samples within the same category. Furthermore, we identify hard categories by incorporating knowledge from both source and target domains, and enhance

the optimization of hard samples within these hard categories via data augmentation and Gini confidence weighting. Experimental results demonstrate the effectiveness of the proposed approach. However, there remains substantial room for improvement in the optimization of hard samples, which constitutes a key direction for our future research.

Author Contributions: Conceptualization, S.L.; methodology, S.L.; software, Y.Z. and X.W.; validation, S.L.; formal analysis, S.L.; investigation, S.L.; resources, X.H., Y.Z. and X.W.; data curation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, X.H.; visualization, S.L.; supervision, X.H.; project administration, X.H.; funding acquisition, X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 12171073). The APC was funded by the National Natural Science Foundation of China (Grant No. 12171073).

Data Availability Statement: The data presented in this study are available at https://link.springer.com/chapter/10.1007/978-3-642-15561-1_16 (Office-31 [36]); <https://www.imageclef.org/2014/adaptation> (ImageCLEF [37]); <http://ai.bu.edu/M3SDA/> (DomainNet [38]), accessed on 12 March 2026.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lu, Y.; Lan, Y.; Yang, H.; Lai, Z.; Li, X. Exploring Generic Knowledge and Reactivating Source Model for Source-free Universal Domain Adaptation. *IEEE Trans. Multimed.* **2026**. [CrossRef]
2. Luo, J.; Tao, H.; Luo, X.; Zhao, Y.; Xiao, Z.; He, D.; Ju, W.; Chen, C.; Hua, X.S.; Zhang, M. Robust Cross Supervision with Target Mining for Source-free Graph Domain Adaptation. *IEEE Trans. Knowl. Data Eng.* **2026**, *38*, 2394–2407. [CrossRef]
3. Liang, J.; Hu, D.; Feng, J. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the International Conference on Machine Learning*. PMLR; ICML: San Diego, CA, USA, 2020; pp. 6028–6039.
4. Yang, S.; van de Weijer, J.; Herranz, L.; Jui, S. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 29393–29405.
5. Liang, S.; Hou, X.; Zeng, Y. Optimized category relation and category individuation for source-free domain adaptation. *Neurocomputing* **2025**, *646*, 130465. [CrossRef]
6. Pan, Z.; Yu, X.; Zhang, W.; Gao, Y. Overcoming learning bias via Prototypical Feature Compensation for source-free domain adaptation. *Pattern Recognit.* **2025**, *158*, 111025. [CrossRef]
7. Liu, P.; Li, J.; Zhao, M.; Xue, W.; Hu, Q.; Chen, S. Domain-Division based Progressive Learning for Source-Free Domain Adaptation. *IEEE Trans. Multimed.* **2025**, *27*, 7081–7092. [CrossRef]
8. Xu, G.; Yi, L.; Xu, P.; Li, J.; Pu, R.; Shui, C.; Ling, C.; McLeod, A.I.; Wang, B. Unraveling the mysteries of label noise in source-free domain adaptation: Theory and practice. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 4007–4023. [CrossRef]
9. Zhang, Y.; Wang, Z.; He, W. Class Relationship Embedded Learning for Source-Free Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 7619–7629.
10. Chen, X.; Yang, H.; Zhang, H.; Yao, H.; Zhu, X. Uncertainty-aware pseudo-label filtering for source-free unsupervised domain adaptation. *Neurocomputing* **2024**, *575*, 127190. [CrossRef]
11. Karim, N.; Mithun, N.C.; Rajvanshi, A.; Chiu, H.P.; Samarasekera, S.; Rahnavard, N. C-sfda: A curriculum learning aided self-training framework for efficient source free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 24120–24131.
12. Huang, J.; Guan, D.; Xiao, A.; Lu, S. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 3635–3649.
13. Yang, S.; Wang, Y.; Wang, K.; Jui, S.; Weijer, J.V. Attracting and dispersing: A simple approach for source-free domain adaptation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 5802–5815.
14. Mitsuzumi, Y.; Kimura, A.; Kashima, H. Understanding and improving source-free domain adaptation from a theoretical perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2024; pp. 28515–28524.
15. Wang, N.; Jin, W.; Bi, H.; Xu, C.; Li, F. Complex-Valued Source-Free Domain Adaptation for PolSAR Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2026**, *64*, 5203216. [CrossRef]

16. Liang, S.; Hou, X.; Zeng, Y. Mean-teacher-based relationship learning with inter-and intra-category optimization for source-free domain adaptation. In *Proceedings of the International Conference on Computer Vision and Image Computing (CVIC 2025)*; SPIE: Bellingham, WA, USA, 2026; Volume 14070, pp. 128–139.
17. Zhang, Y.; Davison, B.D. Efficient pre-trained features and recurrent pseudo-labeling in unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2021; pp. 2719–2728.
18. Chen, C.; Xie, W.; Huang, W.; Rong, Y.; Ding, X.; Huang, Y.; Xu, T.; Huang, J. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2019; pp. 627–636.
19. Zhang, Y.; Hao, X.; Li, F.; Wang, Z.; Li, D.; Li, M.; Mao, R. Unsupervised domain adaptation semantic segmentation method for wheat disease detection based on UAV multispectral images. *Comput. Electron. Agric.* **2025**, *236*, 110473. [[CrossRef](#)]
20. Wang, Y.; Meng, H.; Zhou, Z.; Ding, M.; Zeng, W. Deep Structure Alignment Network for Scalable Unsupervised Domain Adaptation. *Knowl. Based Syst.* **2026**, *343*, 115641. [[CrossRef](#)]
21. Zhang, Y.; Davison, B.D. Adversarial continuous learning in unsupervised domain adaptation. In *Proceedings of the Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event*; Springer: Berlin/Heidelberg, Germany, 2021; Part II, pp. 672–687.
22. Zhang, Y.; Deng, B.; Jia, K.; Zhang, L. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 781–797.
23. Yang, J.; Wen, Z.; Li, Q.; Shao, H. FATransformer: Feature Alignment Transformer for Unsupervised Domain Adaptation in Radio Frequency Fingerprinting Identification. *IEEE Internet Things J.* **2026**, *340*, 115641. [[CrossRef](#)]
24. Zhang, L.; Yu, Z.; Yang, K.; Wang, B.; Chen, C.P. Transferable and discriminative broad network for unsupervised domain adaptation. *Knowl. Based Syst.* **2025**, *315*, 113297. [[CrossRef](#)]
25. Lee, H.J.; Bi, Y.; Lee, S.; Park, G.M.; Kim, J.U.; Kim, S.T.; Jiang, Z.; Navab, N. Unsupervised Domain Adaptation for Medical Image Segmentation Using Adaptogen-Perturbation. *Med. Image Anal.* **2026**, *110*, 104002. [[CrossRef](#)] [[PubMed](#)]
26. Zhu, X.; Zhu, B.; Tan, Y.; Wang, S.; Hao, Y.; Zhang, H. Enhancing zero-shot vision models by label-free prompt distribution learning and bias correcting. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 2001–2025.
27. Kang, G.; Jiang, L.; Yang, Y.; Hauptmann, A.G. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2019; pp. 4893–4902.
28. Cai, Z.; Jing, X.Y.; Shao, L. Visual-depth matching network: Deep rgb-d domain adaptation with unequal categories. *IEEE Trans. Cybern.* **2020**, *52*, 4623–4635. [[CrossRef](#)]
29. Zhang, Z.; Wang, M.; Huang, Y.; Nehorai, A. Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2018; pp. 3437–3445.
30. Liu, Q.; Li, Q.; Li, X.; Pi, J.; Yuan, D.; Liu, Y. Unsupervised Domain Adaptive Thermal Infrared Tracking. *IEEE Trans. Multimed.* **2026**. [[CrossRef](#)]
31. Yoon, H.; Lee, J.; Yang, D. Transferring domain-invariant and adversarially robust feature learning for unsupervised domain adaptation. *Appl. Soft Comput.* **2026**, *193*, 114859. [[CrossRef](#)]
32. Cai, Z.; Huang, Y.; Zhang, T.; Jing, X.Y.; Zheng, Y.; Shao, L. Attention cycle-consistent universal network for more universal domain adaptation. *Pattern Recognit.* **2024**, *147*, 110109. [[CrossRef](#)]
33. Wang, Z.; Hao, Y.; Mu, T.; Li, O.; Wang, S.; He, X. Bi-directional distribution alignment for transductive zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 19893–19902.
34. Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Zhang, K.; Tao, D. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2019; pp. 2427–2436.
35. Wang, Z.; Chen, L.; Wang, F.Y. Fuzzy inference attention module for unsupervised domain adaptation. *IEEE Trans. Fuzzy Syst.* **2023**, *32*, 1706–1718. [[CrossRef](#)]
36. Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Adapting visual category models to new domains. In *Proceedings of the Computer Vision–ECCV 2010: 11th European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2010; Part IV 11, pp. 213–226.
37. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep transfer learning with joint adaptation networks. In *Proceedings of the International Conference on Machine Learning*. PMLR; ICML: San Diego, CA, USA, 2017; pp. 2208–2217.
38. Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2019; pp. 1406–1415.

39. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*. PMLR; ICML: San Diego, CA, USA, 2015; pp. 448–456.
40. Salimans, T.; Kingma, D.P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 901–909.
41. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *Proceedings of the International Conference on Machine Learning*. PMLR; ICML: San Diego, CA, USA, 2015; pp. 1180–1189.
42. Long, M.; Cao, Z.; Wang, J.; Jordan, M.I. Conditional adversarial domain adaptation. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1647–1657.
43. Yang, S.; Wang, Y.; Van De Weijer, J.; Herranz, L.; Jui, S. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2021; pp. 8978–8987.
44. Tang, S.; Chang, A.; Zhang, F.; Zhu, X.; Ye, M.; Zhang, C. Source-free domain adaptation via target prediction distribution searching. *Int. J. Comput. Vis.* **2024**, *132*, 654–672. [[CrossRef](#)]
45. Shao, R.; Zhang, W.; Luo, K.; Li, Q.; Wang, J. Consistent Assistant Domains Transformer for Source-free Domain Adaptation. *IEEE Trans. Image Process.* **2025**, *34*, 6589–6602. [[CrossRef](#)] [[PubMed](#)]
46. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
47. Tang, H.; Yuan, C.; Li, Z.; Tang, J. Learning attention-guided pyramidal features for few-shot fine-grained recognition. *Pattern Recognit.* **2022**, *130*, 108792. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.