Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 34(7): 4626–4647.
DOI: 10.3934/era.2026204
Received: 04 April 2026
Revised: 03 May 2026
Accepted: 20 May 2026
Published: 29 May 2026

Research article

Inverse distillation for source-free unsupervised domain adaptation

Di Wu[1,2,*], **Hui Jiang**[1,2], **Xing Wei**[3,*], **Junlong Xu**[3] **and Zhaoxin Ji**[3]

[1] Key Laboratory of Philosophy and Social Science of Anhui Province on Adolescent Mental Health and Crisis Intelligence Intervention, Hefei Normal University, Hefei 230601, China

[2] School of Computer and Artificial Intelligence, Hefei Normal University, Hefei 230601, China

[3] School of Computer and Information, Hefei University of Technology, Hefei 230601, China

* **Correspondence:** Email: diw@hfnu.edu.cn, weixing@hfut.edu.cn.

Abstract: Unsupervised domain adaptation (UDA) aims to leverage labeled source domain knowledge to improve the target domain's performance. Source-free domain adaptation (SFDA), a recent research focus, addresses challenges such as data privacy by relying solely on a pretrained source model and unlabeled target data, thus eliminating the need for direct access to source domain data. Although many studies have proposed methods such as generating a source domain, using a proxy source domain, or using pseudo-label training, these approaches directly fine-tune the source model, overlooking the excessive bias towards the source domain data in SFDA. The source domain model contains numerous source domain-specific features, and directly updating it to shift towards the target domain is hindered by these domain-specific features. To address this issue, we propose an inverse distillation-based SFDA method. By constructing an initial target domain model, the method extracts pure target domain features and distills them back into the source model, facilitating a smoother transition towards the target domain. Additionally, it identifies stable and active target samples from both structural and scoring perspectives, applying distinct matching strategies for pseudo-label selection. Extensive experiments and ablation studies on public datasets (Digits, Office-31, Office-Home and VisDA-2017) demonstrate the superior performance of our approach in SFDA tasks.

Keywords: source-free domain adaptation; knowledge distillation; image classification

Abbreviations: UDA: unsupervised domain adaptation; SFDA: source-Free domain adaptation; MMD: maximum mean discrepancy; ND: normalized distance; DBDA: dynamic balance-based domain adaptation; GANs: generative adversarial networks; MCD: maximum classifier discrepancy; CGDM: cross-gradient difference minimization; GRL: gradient reversal layer; LDA: linear discriminant analysis

1. Introduction

Deep learning networks have gradually become the mainstream method in various fields, such as image classification, object detection, semantic segmentation, and practical applications like remote sensing and pedestrian re-identification, owing to their end-to-end processing approach and excellent performance in computer vision tasks. However, the outstanding performance of deep learning relies on supervised training with a large amount of labeled data, which allows the network parameters to fit the task data required for training. In practical scenarios, on the one hand, collecting data requires a considerable amount of human effort; on the other hand, there is a high likelihood that the model will face data with distributions different from those in the training set during the application phase. Traditional vision models are designed to serve a single distribution of data and do not consider adaptation and generalization issues. Therefore, the distribution discrepancy between the source domain (training set) and the target domain (test set) can cause a sharp decline in a model's performance.

Thus, unsupervised domain adaptation (UDA) was proposed and has become the mainstream method for addressing the domain discrepancy between the source and target domains. Traditional UDA utilizes labeled source domain data and unlabeled target domain data, aiming to learn a model that performs well on both the source and target domains. With the development of traditional UDA, more domain adaptation tasks have been identified, such as multi-target domain adaptation (MTDA) and source-free domain adaptation (SFDA). SFDA, in particular, aims to generalize a model to both the source and target domains without relying on the source domain data, using only the pretrained source domain model and unlabeled target domain data. SFDA is widely applied in scenarios where the source domain data involve privacy and confidentiality issues, making it more practically meaningful.

The mainstream methods applied to SFDA can be broadly categorized into three types: Generating source domain data, constructing proxy source domains, and pseudo-label self-training. Generating source domain data involves introducing additional network structures, with a typical approach relying on generative adversarial networks (GANs) to generate source-like images for cross-domain adaptation [1, 2]. Constructing proxy source domains entails selecting representative samples from the target domain to replace the inaccessible source domain data, thereby addressing the issue of unavailable source domain data. Pseudo-label self-training typically uses denoising techniques to reduce noise in pseudo-labels and directly uses classification loss to learn the target domain's features. For instance, in [3], during model adaptation, within-domain and cross-domain mixed regularization was introduced, transferring label information from the proxy source domain to the target domain while mitigating the negative impact of noisy labels.

However, most mainstream methods focus on compensating for the lack of source domain data, with pseudo-label training often directly fine-tuning the source domain model. In contrast, this paper addresses another issue in SFDA: The excessive bias of the source model towards the source domain data, which hampers its ability to update due to the presence of source domain-specific features. In the SFDA problem setting, the source domain data are inaccessible, and only the pretrained model trained on the source domain can be utilized. This means that any strategy used to solve the SFDA problem will, directly or indirectly, rely on the complete source domain model, and thus be influenced by source domain-specific features. In traditional UDA tasks, methods such as adversarial alignment [4] and cyclic self-training [5] gradually shift the model towards a common feature space shared by both the source and target domains during training, in order to achieve domain adaptation. Recent progressive

enhancement strategies [6] have further demonstrated the effectiveness of feature-awareness contrastive loss in refining these labels for complex domains.

From this perspective, this paper introduces the concept of inverse distillation. Specifically, we construct a target domain model with randomly initialized parameters. By leveraging the idea of a proxy source domain, we select the most reliable and confident target samples from the target domain data. These samples, though fewer in number, exhibit the highest accuracy. Using supervised classification loss on the target domain model with these confident samples, we enable it to learn the domain-specific features of the target domain. Through the inverse distillation process from the target domain model to the source domain model, target domain-specific knowledge is transferred, directly guiding the source model to adapt to the target domain at the feature space level. Additionally, as the source domain model adapts to the target domain, the pseudo-labels it generates also gradually align with the target domain. We partition stable and active samples based on two metrics: prediction consistency scores and centroid distances, and fully leverage the update direction of the source model to assign pseudo-labels to active samples. This process enables the gradual learning of a target domain model that adapts to the target domain samples.

We summarize the contributions of this paper as follows.

- We identify an overlooked issue in SFDA: The trained source domain model contains numerous source domain-specific features, which introduce excessive bias towards the source domain's space. As a result, direct adaptation based on the source domain model struggles to shift towards the target domain. To address this, we propose the concept of inverse distillation to transfer target domain-specific knowledge to the source domain model.
- We divide the target samples into stable and active samples, and jointly determine pseudo-labels by considering both the feature structure and prediction scores. This approach fully leverages the update direction information of the source model.
- On the basis of the abovementioned proposals, we construct a SFDA framework based on inverse distillation. Extensive experiments and ablation studies are conducted on widely used datasets for SFDA tasks, and the results demonstrate that our method achieves excellent performance.

2. Related works

2.1. Unsupervised domain adaptation

UDA aims to transfer knowledge from the source domain, which is rich in labeled data, to the target domain, which lacks labels. Depending on the specific task setup, the UDA problem can be roughly divided into the following categories: One-to-one single-source domain adaptation (STDA), one-to-many single-source multitarget domain adaptation (MTDA), SFDA, and multisource domain adaptation (MSTDA). Domain adaptation, as one of the typical tasks in transfer learning, provides methods to bridge the domain gap for various visual tasks such as object recognition [4, 7–9], and semantic segmentation [10, 11]. In addition, domain adaptation has also been applied to practical scenarios, such as cross-domain remote sensing image classification [12, 13], domain adaptation in hyperspectral image classification [14], and pedestrian re-identification in multiple scenes [15]. Current mainstream UDA methods can be roughly divided into three categories. The first category is statistical moment matching methods, which use some metric to quantify the difference between the distributions of the source and target domains. By minimizing this difference, these methods achieve alignment between

the source and target domains [7, 16]. The second category includes adversarial learning methods, which borrow ideas from GANs. By constructing a gradient reversal layer (GRL), adversarial training between a generator and a domain discriminator is used to generate domain-invariant features [4]. The third category comprises pseudo-label self-training methods, which generate pseudo-labels for the target domain and iteratively train the model, explicitly learning the domain-specific features of the target domain. Methods like [17–20] filter and weigh the pseudo-labels to reduce the negative impact of incorrect pseudo-labels on the model. However, as domain adaptation continues to evolve, more task scenarios are emerging. Among them, SFDA, which is discussed in the following sections, is of greater practical significance. It does not require access to the source domain data during the transfer process, making it particularly useful in scenarios where the source domain data are difficult to access, such as those involving source domain privacy issues.

2.2. Source-free domain adaptation

The goal of SFDA is to address the domain adaptation problem without accessing the source data. In the past two years, SFDA has gained popularity, with most methods being either generation-based or self-training-based.

Generation-based methods [21–24] generate virtual high-level features for the source domain to connect the unknown source and target distributions. Self-training-based methods attempt to refine the source model using self-supervised techniques, with the most widely used being pseudo-labeling [25–28]. These methods learn from the target samples through different variants of contrastive learning. The methods in [26, 29] obtains pseudo-labels by mining hidden structural information, such as neighboring features. However, generating source samples often introduces additional modules, such as generators or discriminators. At the same time, the pseudo-labels may lead to incorrect labels due to a domain shift, both of which can negatively impact the adaptation process. Another approach [23, 24] is to select a portion of the target data as a pseudo-source domain to compensate for the unseen source domain. A typical method is the entropy criterion [30], which constructs a pseudo-source domain by using the average and maximum entropy of the target dataset to estimate the segmentation ratio, and then selects samples with lower entropy from the pseudo-label target domain for each class on the basis of the segmentation ratio.

In contrast, this paper starts from the bias of the source model towards the source domain and uses a target model inverse distillation strategy to guide the model to update towards the target domain, thereby producing higher-quality pseudo-labels, rather than directly fine-tuning the weights of the source model.

2.3. Self-training

Self-training [31, 32] has traditionally been a widely used method in the semi-supervised learning (SSL) domain. In recent years, self-training has gradually become an important strategy in UDA. In UDA, self-training generally involves two phases: Phase one, uses the model to generate pseudo-labels for the target domain data; and Phase two uses these pseudo-labels as the true labels for the target domain data to retrain the network. The major difference between UDA and semi-supervised learning is that the labeled and unlabeled data come from different distributions, which leads to the low reliability of the pseudo-labels. Since erroneous pseudo-labels propagate errors through backpropagation, directly using pseudo-labels for self-training carries significant risk [33]. In UDA methods, instance-adaptive

self-training (IAST) [17] uses a predefined threshold to filter pseudo-labels. Zheng and Yang [19] estimated uncertainty during the training process to resolve the difficulty and irrationality in threshold selection. Progressive feature alignment network (PFAN) [33] gradually selects reliable pseudo-labeled samples from easy to difficult ones and introduces an adaptive prototype alignment method to suppress the negative impact of erroneous pseudo-labels. In contrast, our method selects the pseudo-labels with the highest confidence in each class for training.

3. Proposed method

In this section, we first introduce the task setting of SFDA, followed by a detailed explanation of the proposed method and the specific operation of each module. Finally, we outline the overall training scheme of the model.

3.1. Task setting

Assume we have a classification model $M_s = C_s(G_s)$ trained on the source domain data $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where G_s is the feature generator and C_s is the classification head. Additionally, we have n_t unlabeled samples $D_t = \{x_i^t\}_{i=1}^{n_t}$ drawn from the target domain distribution $P_t(x_t)$, where the data distribution of the target domain is different from that of the source domain. At the same time, the source and target domains share the same classification targets, i.e., the classification labels $Y_s = Y_t$ for the source and target domain data.

It is important to note that SFDA does not have access to the source domain data (X_s, Y_s) or the target domain labels Y_t. The goal is to train a model that can accurately predict the class labels of the samples and generalize well to the target domain using only the pre-trained source domain model and the unlabeled target domain data.

3.2. Confidence-based target data selection

Due to the inability to access the source domain data in the SFDA task setting, using the target domain's outputs for self-training to shift the source domain model has become one of the mainstream methods. It is well known that the domain discrepancy introduces a significant amount of noise into the pseudo-labels of the target domain, making it difficult to complete the adaptation task. However, experimental results in domain adaptation studies show that even without any modification to the model, it does not exhibit completely non-functional performance on the target domain. In other words, although there are discrepancies between the source and target domains, they still share an overlapping region. Therefore, this paper selects a portion of high-confidence target domain data as the baseline for training in order to achieve the effect of shifting the model.

Other works [34,35] have proposed that the weights of the last fully connected layer of a classification model can serve as class prototype vectors, which effectively addresses the issue of not being able to access the source domain data in SFDA tasks. Inspired by these works, we extract the weight parameters $\phi_{sp} = \{\varphi_1, \varphi_2, ..., \varphi_k\}$ from the last layer of the source model's classification head C_s, where k represents the number of classes. As shown in Figure 1, we then infer the output vector $P_t = \{p_i\}_{i=1}^{n_t}$ for the target domain data on the basis of th source model M_s. By calculating the degree of alignment between each class prototype parameter and the target domain data $D = \phi_{sp} \times P_t$, we select the top N high-confidence target domain samples $\{x_{pt}, y_{pt}\} = \{\{x_{pt}^1, 1\}, \{x_{pt}^2, 2\}, ..., \{x_{pt}^N, N\}\}$ for each class, where N

is a hyperparameter. To avoid the problem of model bias caused by class imbalance, we ensure that an equal number of high-confidence target samples are selected for each class. In subsequent training, we assume that the predicted class labels for these selected high-confidence target domain samples are correct. Specifically, for these high-confidence target samples, we confidently apply cross-entropy loss for training.

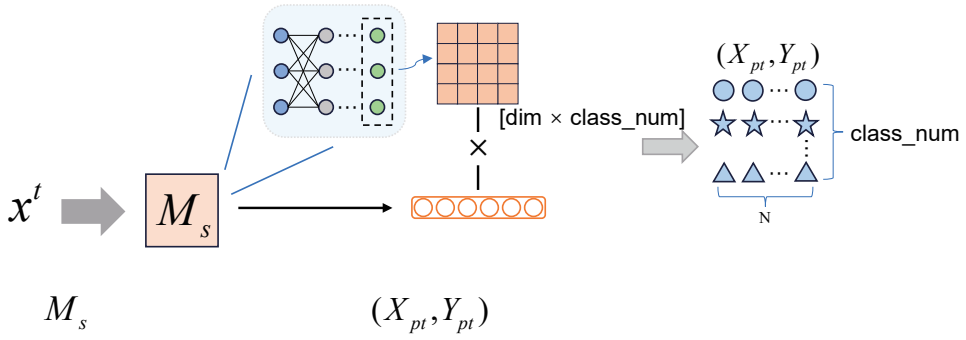

Figure 1. The process of selecting confident target samples.

3.3. Inverse distillation

The difficulty of the SFDA task lies in the inability to access the source domain data. Additionally, we argue that overfitting to the source domain, relative to the target domain, is another often overlooked issue. Since the source model is trained on source domain data using self-supervised learning, the feature space it forms is naturally biased toward the source domain and contains many domain-specific features that do not generalize well to other domains. More importantly, unlike traditional UDA, SFDA lacks a strategy during source domain training to guide the model toward a shared feature space between the source and target domains. Instead, what needs to be shifted is a pretrained, standalone source domain model.

Therefore, inspired by knowledge distillation methods and combined with the selected confident target data, we propose a inverse distillation strategy to indirectly guide the source model to shift and adapt to the target domain, as shown in Figure 2. Specifically, we construct a target model $M_t = C_t(G_t)$ with the same architecture as the source model, but with only initialized parameters. Using the confident target domain samples $\{x_{pt}, y_{pt}\}$, we train the feature extractor G_t and the classification head C_t on the target model M_t by minimizing the classification loss. Although the number of confident target samples is much smaller than the total amount of data in the target domain and may not fully represent the overall distribution of the target domain, for the selected confident target samples, the target model is supervised to predict the same correct class as the source model. More importantly, the target model has a feature distribution that is more aligned with the target domain. Therefore, we implement inverse distillation from the target model to the source model to transfer the target domain feature distribution of the confident target samples. Specifically, the mean squared error (MSE) loss $L_{mse} = \frac{1}{n}\sum_i^n (\hat{y}_i - y_i)^2$ is used as the distillation loss between both models, allowing the source model G_s to learn the feature distribution of the target model G_t. We perform inverse distillation only on the confident target samples.

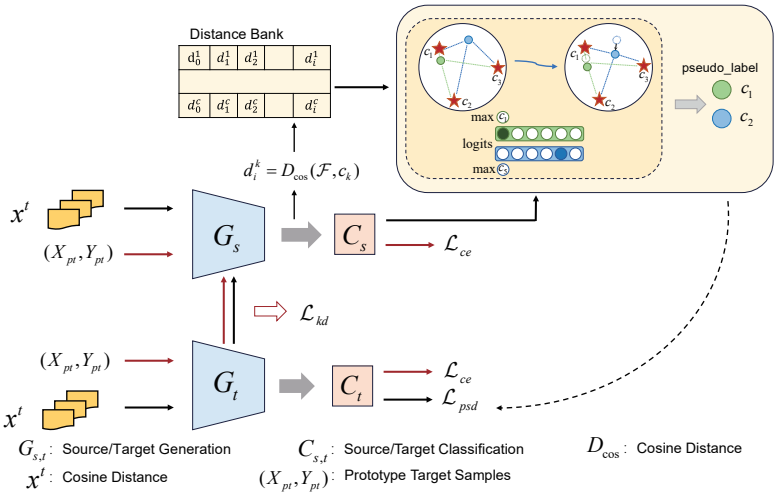

Figure 2. Diagram of the model framework. We briefly describe the working process of the model. (1) Supervised learning of target domain features on G_t using confidence target samples is inverse distilled to the source domain generator G_s to steer the source model toward the target domain. To ensure the correctness of the source model's prediction of the confidence target samples, the same classification loss L_{ce} is used on the source model, which is shown in the red flow line in diagram. (2) The output features of the target samples from the source domain generator are stored, along with the distance to the class centroids. The samples are then segmented according to the joint feature point structure and the predicted score information. Stable samples are incorporated into the inverse distillation loss calculation, while active samples are trained with pseudo-labels using the feature point update direction, as shown in the upper part of the diagram.

Meanwhile, to ensure that the source model's feature generator G_s maintains correct classification predictions during the inverse distillation process while shifting towards the target domain, we continue to apply the classification loss to the confident target samples in the source model during inverse distillation. Under the bidirectional constraints of both the distillation loss and classification loss, the source model maintains correct predictions while gradually shifting toward the target domain. The process above can be expressed by the following formulas:

$$L_{cls\_t} = L_{ce}(C_t(G_t(x_{pt})), y_{pt}), \tag{3.1}$$

$$L_{kd} = L_{mse} = \frac{1}{N}\sum_{i}^{N}(y_i^s - y_i^t)^2, \tag{3.2}$$

$$L_{cls\_s} = L_{ce}(C_s(G_s(x_{pt})), y_{pt}), \tag{3.3}$$

where y_i^s and y_i^t are the output feature vectors of the confident target samples from the source model generator and the target model generator, respectively.

3.4. Structured pseudo-label selection

As mentioned in 3.2, the SFDA task setting provides a model that is overfitted on the source domain. Compared with pseudo-label self-training methods in UDA and semi-supervised learning, SFDA cannot

filter pseudo-labels during the process to gradually correct them. Therefore, the pseudo-labels output by the source model carry a significant amount of domain-specific noise, i.e., incorrect samples that are suitable for the source domain but not for the target domain. Inspired by [25], centroids can robustly and more reliably represent the distribution of different classes within the target domain.

In this paper, on the basis of the update process of the source model's feature space, and combining both the structural and predicted score dimensions, we divide the target domain data into stable samples and active samples. Stable samples are considered more likely to be correct predictions, with the model's output scores used as pseudo-labels. Active samples are considered to be in an updating state, and their pseudo-labels are based on the trend of the feature structure update.

Under the assumption proposed in 3.2, confident target samples are considered to be correctly predicted target domain samples. Similarly, the feature centroids of confident target samples, generated by the source model generator G_s, are considered to be the target domain's class centroids. We use the average of the feature outputs for confident target samples in the source domain model as the class centroids as follows:

$$c_k = \frac{\sum_{x_{pt} \in X_{pt}} \mathbb{1}(y_{pt} = k) G_s(x_{pt})}{\sum_{x_{pt} \in X_{pt}} \mathbb{1}(y_{pt} = k)}, \tag{3.4}$$

where k denotes the corresponding class; $\mathbb{1}_{(y_{pt}=k)}$ is the indicator function, which takes the value 1 when $k = y_{pt}$ and 0 otherwise; and c_k represents the centroid of the k-th class.

Unlike conventional pseudo-label selection methods based on centroids, we consider the model's update process. First, we argue that simply using the distance between the sample features and class centroids as the selection criterion is unreasonable. Even with the correct update direction, the samples may move away from the correct centroid and approach the wrong one. Therefore, we construct a distance $Bank(N \times k)$ to record the current model's feature distances to each centroid, using the commonly used cosine distance as the metric for feature layers. After one batch of training, we calculate the difference between the new feature distance d_bank^e and the stored feature distance d_bank^{e-1}. The class with the smallest distance update difference is chosen as one of the factors for selecting the pseudo-label of the target sample, i.e., under the guidance of confident target sample inverse distillation, the sample moves towards the class with the largest update trend.

$$d\_bank_i^e = D_{\cos}(f_i^e, \{c_k\}_{k=0}^n), \tag{3.5}$$

where f_i^e denotes the features of the i-th sample in batch e from G_s.

Additionally, we incorporate both the predicted scores and the structural aspects. Samples that are correctly classified and tend to stabilize might still undergo some shift due to model updates. This update trend can amplify such shifts, leading to incorrect pseudo-labels. For example, consider a stable sample with Class 0, whose distances to the class centroids are $[0.1, 1.5, 1.6]$. After the model updates, the new distance vector becomes $[0.2, 1.6, 1.5]$, and the distance difference vector is $[0.1, 0.1, -0.1]$, which would incorrectly assign the pseudo-label to Class 2. For these stable samples, the irreversible effect of incorrect labels can cause the model to fail.

$$y_{score} = \arg\max\{C_s(G_s(x_{t\_w}))\}, \tag{3.6}$$

$$y_{dist} = \arg\min\{d\_bank^e\}, \tag{3.7}$$

$$\widehat{y}_t = \begin{cases} \arg\min\{d\_bank^e - d\_bank^{e-1}\}, & y_{score} \neq y_{dist} \\ y_{score}, & y_{score} = y_{dist} \end{cases}, \tag{3.8}$$

where x_{t_w} is the weak augmentation of the target domain sample, and we base the self-training process on [31]. The model gives better predictions for weakly augmented inputs, so the weakly augmented target samples are used as inputs to the source model to generate pseudo-labels. In contrast, the strongly augmented samples are used for training in the target model. This approach reduces the probability of error in pseudo-labels while leveraging strong augmentation to provide the target model with better learning performance. The category loss on the target model side is as follows:

$$L_{psd} = L_{a\_ce}(x^a_{t\_s}, \hat{y}^a_t) + \lambda L_{ia\_ce}(x^{ia}_{t\_s}, \hat{y}^{ia}_t), \tag{3.9}$$

where $x^a_{t\_s}$ denotes active samples and $x^{ia}_{t\_s}$ denotes stabilized samples. In addition, the stabilizing target sample is added to the loss calculation for inverse distillation.

3.5. Training process

In this work, given the idea of inverse distillation, we do not directly fine-tune the source model, but instead construct a target model to indirectly transfer pure target domain information.

Additionally, we categorize the target domain samples and jointly select target domain pseudo-labels from both the structural and score perspectives, allowing samples at different stages to play their corresponding roles. To clearly express the transfer process and how each module functions, we provide the detailed training process of the model here.

Step 1: Extract the parameters of the last fully connected layer of the source model's classification head as the class prototypes for the source domain. By calculating the compatibility between each class prototype and the target domain data, select the top N confident target samples (X_{pt}, Y_{pt}) for each class. Then, use these confident target samples to train the target model M_t.

$$\min_{\theta_1, \varphi_1} L_{cls\_t} \tag{3.10}$$

Step 2: Perform inverse distillation from the target model generator G_t to the source model generator G_s, guiding the source model's feature space to update towards the target domain. At the same time, to ensure that the updates do not affect the prediction output of the confident target samples, and jointly optimize the classification loss of the confident target samples in the source model.

$$\min_{\theta_s, \varphi_s} L_{src} = \lambda L_{kd} + L_{cls\_s} \tag{3.11}$$

Step 3: Update the category centroids of the confident target samples and record the distances between the output features of the source model generator G_s and each category centroid in the distance d_bank. Based on the consistency between the source model's predicted output class and the nearest centroid class, divide the target samples into stable target samples and active target samples. For stable samples, use the model's predicted class as the pseudo-label; for active target samples, use the update trend of the distance between the output features and the class centroids as the pseudo-label. (Since

the distance update trend occurs at least after the second batch, and to balance the impact of stable and active samples on the model, Step 3 starts participating in training from the second batch.)

$$\min_{\theta_t,\varphi_t} L_{psd} = L_{a\_ce}(x^a_{t\_s}, \hat{y}^a_t) + L_{ia\_ce}(x^{ia}_{t\_s}, \hat{y}^{ia}_t) \tag{3.12}$$

After training, we use G_t, C_t as the final model.

Algorithm 1 Source-Free Domain Adaptation for Image Classification Based on Inverse Distillation.

Input

Source domain model: $M_s = C_s(G_s)$

Target domain data: $D = \{x_1, x_2, \ldots, x_{n_t}\}$

Initialize network parameters: Feature generator G_t, classifier C_t

Initialize hyperparameters: λ, N.

Output

Feature extractor G_t and classifier C_t.

Process

Extract the last fully connected layer parameters of the source model's classification head as the class prototypes for the source domain. Select the top N confident target samples for each class$\{x_{pt}, y_{pt}\} = \{\{x^1_{pt}, 1\}, \{x^2_{pt}, 2\}, ..., \{x^N_{pt}, N\}\}$.

for epoch =1 to M **do:**

Step 1

Train the target model G_t and C_t using confident target samples:

$$\min_{\theta_t,\varphi_t} L_{cls\_t}(x_{pt,}y_{pt})$$

Step 2

Perform inverse distillation from the target model generator G_s to the source model generator G_t , guiding the source model's feature space to update towards the target domain. Simultaneously, minimize the classification loss on the confident target samples in the source model.

$$\min_{\theta_s,\varphi_s} L_{src} = \lambda L_{kd} + L_{cls\_s}$$

Step 3

Use the structured pseudo-label selection strategy to divide the target samples into stable and active samples. Minimize the classification loss of target samples on the target model G_t and classifier C_t using the following pseudo-labels:

$$\min_{\theta_t,\varphi_t} L_{psd} = L_{a\_ce}(x^a_{t\_s}, \widehat{y}^a_t) + L_{ia\_ce}(x^{ia}_{t\_s}, \widehat{y}^{ia}_t)$$

End for

4. Experiments

In this section, we conduct a comprehensive set of experiments on visual tasks in the domain adaptation field to evaluate the effectiveness of proposed method. We first introduce the datasets used in the experiments and provide details on the specific parameter settings for our model across different experiments. Next, we compare our method with several baseline approaches. Additionally, we analyze the specific role and effectiveness of the modules we proposed in this section. These results demonstrate the effectiveness of the proposed method.

4.1. Dataset

We conducted experiments on four commonly used benchmark datasets for SFDA: Digits, Office-31, Office-Home, and VisDA-2017. The descriptions of these four datasets are as follows.

Digits [36]: The Digits dataset is a classic benchmark in the Unsupervised Domain Adaptation (UDA) problem and is widely used for digit image classification tasks. We use the three most commonly used subsets of the Digits dataset: the Street View House Numbers (SVHN, denoted as S), the Modified National Institute of Standards and Technology (MNIST, denoted as M), and the United States Postal Service (USPS, denoted as U). These subsets contain images of the digits 0_9 written in different environments, each offering handwritten digits with varying writing styles and backgrounds. This makes the dataset suitable for evaluating the cross-domain adaptation ability of models.

Office-31 [37]: The Office-31 dataset is a publicly available dataset for domain adaptation research. It consists of 31 categories of images collected from different office and daily life scenes, covering three different domains: Amazon (A), webcam (W), and digital single-lens reflex (DSLR) cameras (D). The Amazon domain contains product images from the Amazon website, representing a standard office environment. The webcam domain contains images taken by a webcam, representing lower resolution and lower-quality images. The DSLR domain contains high-quality images captured by a DSLR camera.

Office-Home [38]: The Office-Home dataset is a classic benchmark, consisting of 65 categories and widely used for image classification and object detection tasks. It includes four domains, namely art (A), clipart (C), product (P), and real World (R), with each domain containing images from various categories. These domains cover a wide range of visual styles, from artistic illustrations to real-world scenes, making the dataset ideal for testing a model's ability to adapt across different visual styles.

VisDA-2017 [39]: The VisDA-2017 dataset is a big dataset used in UDA problems and is widely applied in cross-domain classification tasks in the visual domain. This dataset contains images from 12 categories, divided into a source domain and a target domain. The source domain consists of synthetic images, while the target domain consists of real-world images. These images cover various types of objects, such as airplanes, cars, ships, etc., and are designed to test a model's adaptability between synthetic and real-world images. We conducted multiple challenging cross-domain tasks using the VisDA-2017 dataset, with the main task being the migration from synthetic images to real images.

4.2. Experimental setup

Following the standard task setup of SFDA, we train the model using a classification model that is trained only on the source domain data, and unlabeled target domain data without any real labels.

We implement the method proposed in this paper using the PyTorch framework [40]. Similar to most approaches, we use the ResNet [41] series, pre-trained on ImageNet [42], as the generator's network architecture. The classification head consists of a BottleNeck layer (input dimension * 1024) and a fully connected layer (1024 * number of classes), with the stochastic gradient descent (SGD) optimizer chosen. For Digits, Office-31 and Office-Home, we use ResNet-50 as the generator network, with a learning rate of 1e-3, momentum of 0.9, and a batch size of 32. On the VisDA-2017 dataset, we use ResNet-101 as the generator network, with a learning rate of 1e-5 and a batch size of 24 due to graphical processing unit (GPU) memory limitations. Additionally, we set the weight decay to 4e-5 and the maximum number of iterations to 100. The hyperparameter $\lambda_1 = 0.7$ is chosen for optimal performance in the classification task. For different datasets, the number of confident target samples N is set to 5 for Digits and Office-31, 3 for Office-Home, and 8 for VisDA-2017.

For the training of the source model, we uniformly use self-supervised cross-entropy loss and train on the source domain data until convergence (set to 100 epochs in all experiments), then save the source model for use in the experiments.

4.3. Experimental results

Experimental results on Digits: As shown in the Table 1, our proposed framework performs excellently in digit recognition. Even compared with methods with access to the source domain, our approach achieves a competitive advantage. The average accuracy across the three tasks surpasses source hypothesis transfer (SHOT [25]) by 0.4 percentage points.

Table 1. Classification accuracies (%) on the Digits dataset.

Method	SF	S→M	U→M	M→U	AVG
Source model	–	73.3	90.2	78.8	80.8
ADDA [43]	×	76.0	90.1	89.4	85.2
ADR [44]	×	95.0	93.1	93.2	93.8
CyCADA [36]	×	90.4	96.5	95.6	94.2
CDAN [7]	×	89.2	98.0	95.6	94.3
CAT [45]	×	98.8	96.0	94.0	96.3
SWD [46]	×	**98.9**	97.1	98.1	98.0
SHOT [25]	✓	**98.9**	97.5	97.9	98.1
Ours	✓	**98.9**	**98.3**	**98.4**	**98.5**

Experimental results on Office-31: As shown in Table 2, our method achieves the best classification accuracy on Office-31, outperforming the class prototype discovery (CPD) method by 0.2 percentage points. Notably, our method demonstrates optimal performance on the tasks A→D, A→W, D→W, and W→D, while it slightly lags behind other methods on the D→W and W→A tasks. Upon analysis, we found that our method has a certain reliance on confident target samples. For tasks with large distributional differences, the accuracy of confident target samples significantly impacts the model's training outcomes. Conversely, when the confident target samples have high accuracy, the model achieves better classification performance than the other methods.

Table 2. Classification accuracy (%) on Office-31 for SFDA (Resnet-50).

Method	A→D	A→W	D→A	D→W	W→ A	W→D	Avg
SDDA [1]	85.3	82.5	66.4	99.0	67.7	99.8	83.5
SFDA [47]	94.6	98.1	**100.0**	92.0	74.6	75.2	89.1
SHOT [25]	94.0	90.1	74.7	98.4	74.3	99.9	88.6
U-SFAN+ [48]	92.8	98.0	99.0	94.2	74.6	74.4	88.8
3C-GAN [22]	92.7	93.7	75.3	98.5	77.8	99.8	89.6
SFIT [2]	89.9	91.8	73.9	98.7	72.0	99.0	87.7
BAIT [49]	94.6	98.1	100.0	92.0	74.6	75.2	89.1
D-MCD [50]	94.1	93.5	76.4	98.8	76.4	**100.0**	89.9
A2Net [27]	94.5	94.0	76.7	99.2	76.1	**100.0**	90.1
SFDA-DE [51]	96.0	94.2	76.6	98.5	75.5	99.8	90.1
DIPE [52]	96.6	93.1	75.5	98.4	77.2	99.6	90.1
SFADA [53]	91.0	97.9	99.8	92.6	73.8	75.1	89.4
ASM [54]	96.0	95.1	75.3	98.7	77.2	**100.0**	90.4
CPD [55]	96.6	94.2	**77.3**	98.2	**78.3**	**100.0**	90.8
Ours	**97.7**	**96.8**	75.8	**99.3**	76.4	**100.0**	**91.0**

Experimental results on Office-Home: Compared with Office-31, Office-Home has a larger dataset and more categories. We conducted 12 transfer tasks and compared the performance on Office-Home with publicly available SFDA methods. The results are presented in Table 3. It is evident that our method achieves the highest average accuracy, with a 1% performance improvement over the latest SFDA method, target prediction distribution searching (TPDS). This indicates that our method is capable of handling more complex tasks.

Table 3. Classification accuracy (%) on Office-Home for SFDA (Resnet-50).

Methods	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
SFDA [47]	48.4	73.4	76.9	64.3	69.8	71.7	62.7	45.3	76.6	69.8	50.5	79.0	65.7
BAIT [49]	57.4	77.5	82.4	68.0	77.2	75.1	67.1	55.5	81.9	73.9	59.5	84.2	71.6
SHOT [25]	56.7	77.9	80.6	68.0	78.0	79.4	67.9	54.5	82.3	74.2	58.6	84.5	71.9
SFADA [53]	56.1	78.0	81.6	68.5	79.5	78.5	67.8	56.0	82.3	73.6	57.8	83.0	71.9
NRC [26]	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
U-SFAN+ [48]	57.8	77.8	81.6	67.9	77.3	79.2	67.2	54.7	81.2	73.3	60.3	83.9	71.9
AaD [56]	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
CoWA [57]	56.9	78.4	81.0	69.1	80.0	79.9	67.7	57.2	82.4	72.8	60.5	84.5	72.5
ELR [58]	58.4	78.7	82.5	69.2	79.5	79.3	66.3	58.0	82.6	73.4	59.8	85.1	72.6
PLUE [59]	49.1	73.5	78.2	62.9	73.5	74.5	62.2	48.3	78.6	68.6	51.8	81.5	66.9
DIPE [52]	56.5	79.2	80.7	70.1	79.8	78.8	67.9	55.1	83.5	74.1	59.3	84.8	72.5
CPD [55]	59.1	79.0	82.4	68.5	79.7	79.5	67.9	**57.9**	82.8	73.8	**61.2**	84.6	73.0
TPDS [60]	**59.3**	80.3	82.1	**70.6**	79.4	80.9	**69.8**	56.8	82.1	**74.5**	**61.2**	85.3	73.5
Ours	57.8	**82.4**	**82.7**	69.2	**80.2**	**81.3**	68.7	55.4	**84.2**	74.0	60.1	**86.7**	**74.5**

Experimental results on VisDA-2017: VisDA contains only one task: From simulation to real. In Table 4, we report the recognition rate and average accuracy across 12 categories. Our method achieves the best performance, surpassing ASM by 2.4 percentage points, demonstrating that our method can still maintain good transfer performance in large-scale, complex task scenarios. This is also attributed to the relatively small distributional differences between multiple categories in the source and target domains, which results in more accurate confident target samples.

Table 4. Classification accuracy (%) on VisDA-2017 for SFDA (Resnet-101).

Methods	Airplane	Bike	Bus	Car	Horse	Knife	Motorbike	Person	Plant	Skateboard	Train	Truck	Avg
SFDA [47]	86.9	81.7	84.6	63.9	93.1	91.4	86.6	71.9	84.5	58.2	74.5	42.7	76.7
3C-GAN [22]	94.8	73.4	68.8	**74.8**	93.1	**95.4**	88.6	**84.7**	89.1	84.7	83.5	48.1	81.6
SFIT [2]	94.3	79.0	84.9	63.6	92.6	92.0	88.4	79.1	92.2	79.8	87.6	43.0	81.4
A2Net [27]	94.0	87.8	**85.6**	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
BAIT [49]	93.7	83.2	84.5	65.0	92.9	**95.4**	88.1	80.8	90.0	89.0	84.0	45.3	82.7
SFADA [53]	94.2	79.6	79.8	65.7	92.6	94.1	87.3	80.8	88.1	91.4	83.3	55.0	82.7
SHOT [25]	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
U-SFAN+ [48]	94.9	87.4	78.0	56.4	93.8	95.1	80.5	79.9	90.1	90.1	85.3	60.4	82.7
ASM [54]	**95.2**	87.8	79.7	60.3	94.1	94.8	85.0	81.1	91.9	89.9	87.3	**61.6**	84.1
Ours	94.8	**96.7**	84.1	66.1	**94.7**	95.1	**89.9**	82.4	**94.2**	**90.7**	**91.4**	58.4	**86.5**

5. Model analysis

5.1. Ablation study

In this section, we primarily discuss the effectiveness of the inverse transfer strategy and the pseudo-label selection strategy, presenting the results of the ablation study and providing an analysis.

Table 5 shows the results of our ablation study on the inverse distillation strategy and the pseudo-label selection strategy. In the training process without the pseudo-label selection strategy, the predicted labels are directly used as pseudo-labels. The model framework without inverse distillation is shown in Figure 3. First, the pseudo-label selection strategy takes both the feature structure relationships and the predicted scores of the samples into account. It not only preserves the correct information for stable samples but also promotes the update of active samples toward the target domain, thereby leading to performance improvement.

Table 5. Ablation study of the A → D, A → W adaptation tasks with different loss functions.

Inverse distillation	Pseudo-label selection	A→D	A→W
-	-	91.8	92.1
-	✓	92.2	92.4
✓	-	94.7	94.8
✓	✓	97.7	96.8

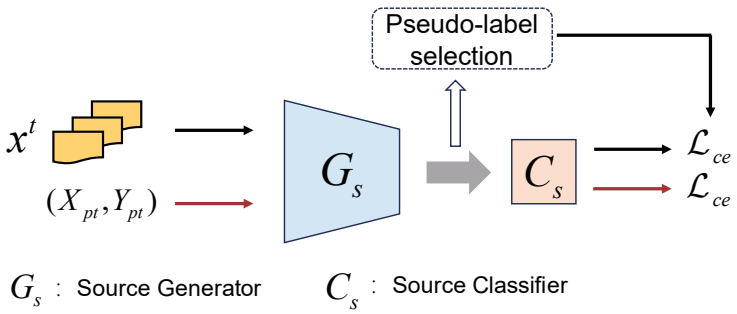

Figure 3. Schematic diagram of the framework of the noninverse distillation strategy.

For the inverse distillation module, we followed mainstream SFDA methods by directly fine-tuning the source model's parameters. The results of the ablation study clearly show that even with the introduction of the pseudo-label selection strategy, fine-tuning the source model does not bring significant performance improvements. This confirms the difficulty of adapting the source model to the target domain in SFDA. Upon analysis, the pseudo-labels are predicted on the basis of the source model, meaning that these pseudo-labels inevitably carry information from the source model, which mainly consists of domain-independent features from the source domain, with only a small portion involving features shared between the source and target domains. In contrast, inverse distillation creates an additional target domain model and directly learns the target domain features. The inverse distillation of target domain-specific features enables a more direct update direction for the source model. From the data, it is evident that the inverse distillation strategy improved accuracy by 2.9% and 2.7% on the A→D and A→W tasks, respectively.

5.2. Hyperparameter analysis

As shown in Figure 4, we conducted hyperparameter analysis on the number of confident target samples per class (N) and the inverse distillation loss hyperparameter (λ) using the A→D task on Office-31. It is clear that the model is not highly sensitive to the inverse distillation loss parameter under different values, with a significant performance drop only when λ is set to 0.1. We deduce that when λ is 0.1, the distillation loss causes minimal model updates compared with the overall loss, preventing the source model from learning the target domain-specific features and thus hindering domain adaptation.

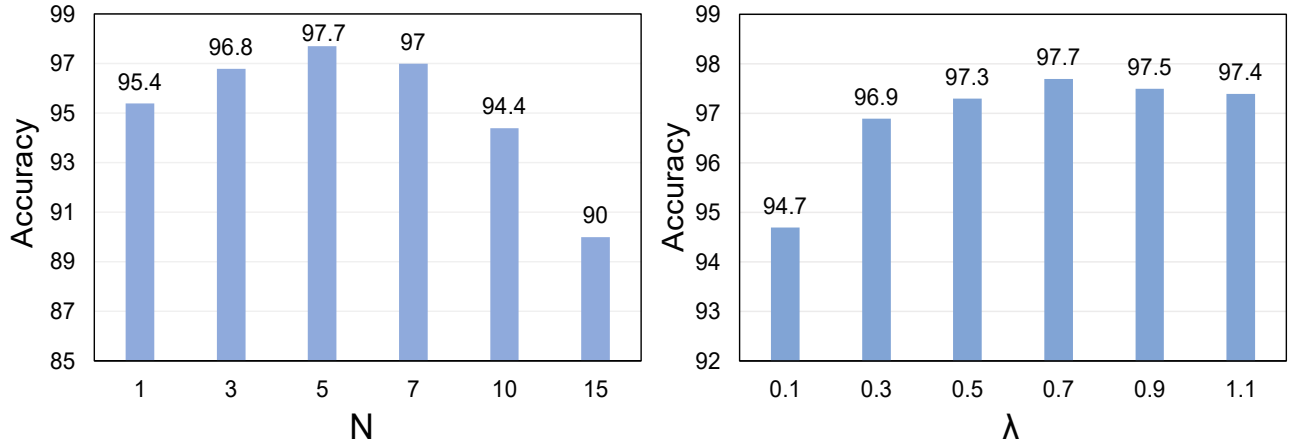

Figure 4. Hyperparameter performance. Left: Number of confident target samples per class, N; right: Inverse distillation loss hyperparameter λ (A → D).

For the number of confident target samples per class (N), the model is more sensitive. This is why different values of N were chosen for different datasets in the experimental setup. During training, we consistently assume the correctness of confident target samples. On the one hand, when the number of confident target samples is small, the correctness is guaranteed, but the limited quantity prevents the model from achieving optimal performance. On the other hand, when there are too many confident

target samples, their correctness cannot be guaranteed, leading to a rapid drop in accuracy, as shown in Figure 4 (left). This is a balance that needs to be struck. However, compared with the erroneous information brought by incorrect samples, a smaller number of confident target samples is not a critical issue. Therefore, for each dataset, we conservatively select a lower value for N.

5.3. Representative examples

Due to the existence of domain shifts, the source domain model performs poorly on the task of target domain recognition. This phenomenon is largely due to the model's bias towards the source feature space, failing to properly capture the target domain's feature space. As a result, when recognizing images, the model cannot focus on the objects to be recognized. We use Grad-CAM [61] technology to visualize the areas our method focuses on during training. As shown in Figure 5, when using the ResNet model, the concentration on the object is significantly higher than the overall concentration, which is highly detrimental to the classification of object categories. Our proposed method learns a certain distribution of the target domain's feature space, and after inverse distillation to the source domain, it still retains a large amount of target domain information. This is highly effective in mitigating the detrimental effects caused by a domain shift.

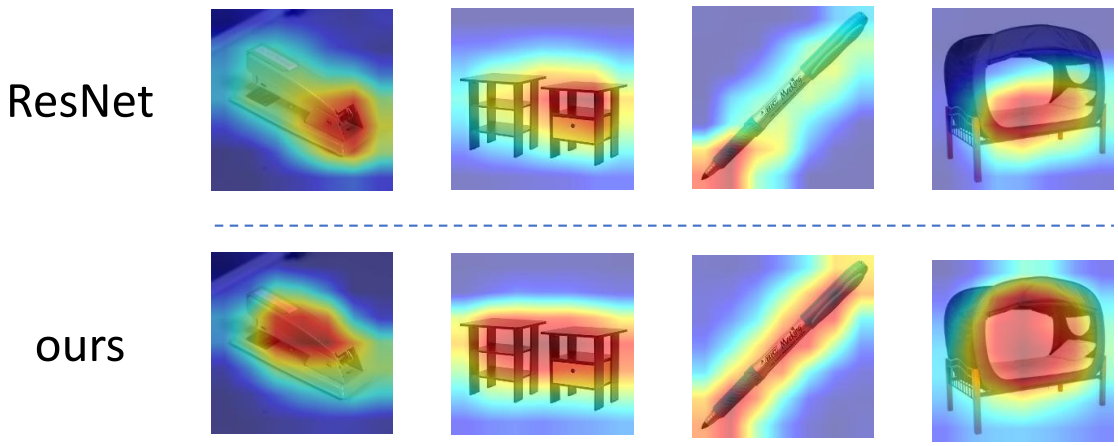

Figure 5. Visualizations of concentration of the last convolutional layer of the samples.

5.4. Training convergence

Figure 6 shows the transfer experiment on the A→D task in the Office-31 dataset, with the x-axis representing the number of training iterations. Our method converges quickly and stably, reaching optimal performance after 20 epochs, and stabilizing completely after 25 epochs. With the increase in training iterations, our method significantly reduces the loss and improves accuracy, demonstrating that the entire training process of our method is stable and converges effectively.

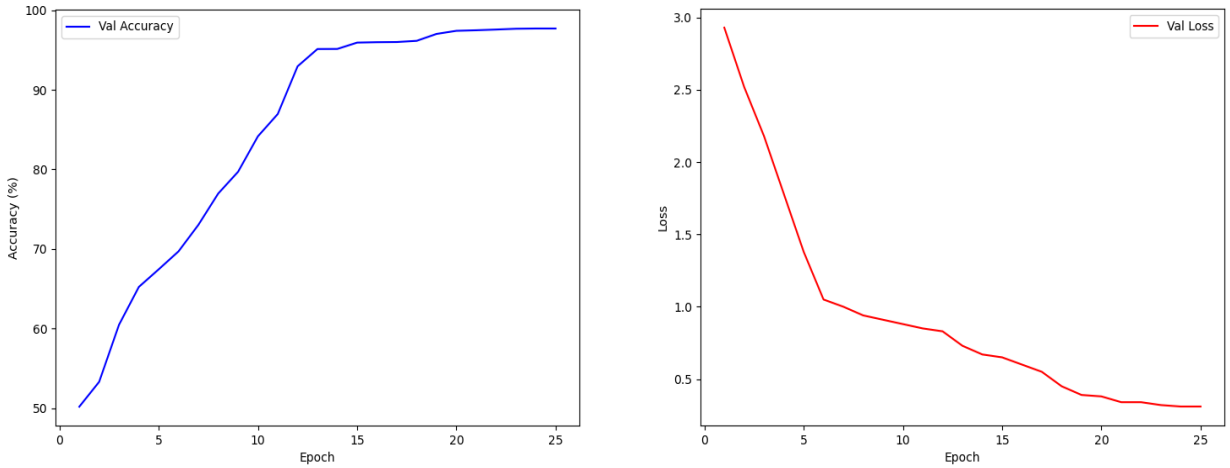

Figure 6. Training process. The convergence of the model on the A→D task in Office-31 is described. The left plot shows the change in recognition accuracy during the training process, while the right plot shows the change in the loss function during training.

6. Conclusions

This paper proposes an inverse distillation-based SFDA method for image classification, specifically designed for SFDA tasks where access to the source domain data is unavailable. However, existing SFDA models often overlook the issue of overfitting in the source model. These methods typically fine-tune the source model directly or copy the weight parameters before fine-tuning, which can lead to excessive reliance on source-domain-specific features, hindering the model's ability to shift effectively to the target domain. In contrast, we introduce a pure target domain model that learns target-domain-specific features, which are then transferred to the source model via inverse distillation. This approach helps the source model adapt to the target domain. Additionally, the method classifies the target samples into stable and active categories depending on the consistency between the features' structure and predicted scores. It leverages the update direction of active samples to fine-tune the model further. Extensive experiments validate the effectiveness of the proposed method.

Use of AI tools declaration

AI tools were used solely to assist with language polishing, including grammar correction and improvement of academic expression. All research design, analysis, and conclusions were independently completed by the author.

Acknowledgments

This work was supported by the Natural Science Research Project of Hefei Normal University (KYSR2025073, KYSR2025092, KYSR2025129 and KYSR2025130), the Open Fund of Key Laboratory of Philosophy and Social Science of Anhui Province on Adolescent Mental Health and Crisis Intelligence Intervention (SYS2024B05), Natural Science Research Project of Anhui

Educational Committee (2022AH052144), Anhui Mine IOT and Security Monitoring Technology Key Laboratory (2109Y-09-04).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. V. K. Kurmi, V. K. Subramanian, V. P. Namboodiri, Domain impression: A source data free domain adaptation method, in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2021), 615–625. https://doi.org/10.1109/WACV48630.2021.00066
2. Y. Hou, L. Zheng, Visualizing adapted knowledge in domain transfer, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 13819–13828. https://doi.org/10.1109/CVPR46437.2021.01361
3. Y. Ding, L. Sheng, J. Liang, A. Zheng, R. He, Proxymix: Proxy-based mixup training with label refinery for source-free domain adaptation, *Neural Netw.*, **167** (2023), 92–103. https://doi.org/10.1016/j.neunet.2023.08.005
4. Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in *Proceedings of the 32nd International Conference on Machine Learning*, **37** (2015), 1180–1189.
5. H. Liu, J. Wang, M. Long, Cycle self-training for domain adaptation, in *Advances in Neural Information Processing Systems*, **34** (2021), 22968–22981.
6. W. Zhou, G. Guan, Y. Yi, W. Cui, Y. Chen, Progressive pseudo-labels enhancement for source-free domain adaptation medical image segmentation, *Biomed. Signal Process. Control*, **109** (2025), 108053. https://doi.org/10.1016/j.bspc.2025.108053
7. M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in *Proceedings of the 32nd International Conference on Machine Learning*, **37** (2015), 97–105.
8. S. Li, S. Song, G. Huang, Z. Ding, C. Wu, Domain invariant and class discriminative feature learning for visual domain adaptation, *IEEE Trans. Image Process.*, **27** (2018), 4260–4273. https://doi.org/10.1109/TIP.2018.2839528
9. B. Xu, Z. Zeng, C. Lian, Z. Ding, Few-shot domain adaptation via mixup optimal transport, *IEEE Trans. Image Process.*, **31** (2022), 2518–2528. https://doi.org/10.1109/TIP.2022.3157139
10. Y. H. Tsai, W. C. Hung, S. Schulter, K. Sohn, M. H. Yang, M. Chandraker, Learning to adapt structured output space for semantic segmentation, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 7472–7481. https://doi.org/10.1109/CVPR.2018.00780
11. Y. Zou, Z. Yu, B. V. K. Kumar, J. Wang, Unsupervised domain adaptation for semantic segmentation via class-balanced self-training, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), 289–305.

12. J. Peng, Y. Huang, W. Sun, N. Chen, Y. Ning, Q. Du, Domain adaptation in remote sensing image classification: A survey, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **15** (2022), 9842–9859. https://doi.org/10.1109/JSTARS.2022.3220875

13. Y. Huang, J. Peng, W. Sun, N. Chen, Q. Du, Y. Ning, et al., Two-branch attention adversarial domain adaptation network for hyperspectral image classification, *IEEE Trans. Geosci. Remote Sens.*, **60** (2022), 1–13. https://doi.org/10.1109/TGRS.2022.3215677

14. Y. Ning, J. Peng, L. Sun, Y. Huang, W. Sun, Q. Du, Adaptive local discriminant analysis and distribution matching for domain adaptation in hyperspectral image classification, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **15** (2022), 4797–4808. https://doi.org/10.1109/JSTARS.2022.3181577

15. L. Zha, Y. Chen, P. Zhou, Y. Zhang, Intensifying the consistency of pseudo label refinement for unsupervised domain adaptation person re-identification, in *2023 IEEE International Conference on Multimedia and Expo (ICME)*, (2023), 1547–1552. https://doi.org/10.1109/ICME55011.2023.00267

16. M. Long, H. Zhu, J. Wang, M. I. Jordan, Deep transfer learning with joint adaptation networks, in *Proceedings of the 34th International Conference on Machine Learning*, **70** (2017), 2208–2217.

17. K. Mei, C. Zhu, J. Zou, S. Zhang, Instance adaptive self-training for unsupervised domain adaptation, in *Computer Vision–ECCV 2020: 16th European Conference*, (2020), 415–430. https://doi.org/10.1007/978-3-030-58574-7_25

18. W. Zhang, W. Ouyang, W. Li, D. Xu, Collaborative and adversarial network for unsupervised domain adaptation, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 3801–3809. https://doi.org/10.1109/CVPR.2018.00400

19. Z. Zheng, Y. Yang, Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation, *Int. J. Comput. Vis.*, **129** (2021), 1106–1120. https://doi.org/10.1007/s11263-020-01395-y

20. M. Shao, S. Chen, F. Wang, L. Zhang, Adaptive pseudo-label threshold for source-free domain adaptation, *Neural Comput. Appl.*, **37** (2025), 1875–1887. https://doi.org/10.1007/s00521-024-10697-y

21. Z. Qiu, Y. Zhang, H. Lin, S. Niu, Y. Liu, Q. Du, et al., Source-free domain adaptation via avatar prototype generation and adaptation, in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, (2021), 2921–2927. https://doi.org/10.24963/ijcai.2021/402

22. R. Li, Q. Jiao, W. Cao, H. S. Wong, S. Wu, Model adaptation: Unsupervised domain adaptation without source data, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 9638–9647. https://doi.org/10.1109/CVPR42600.2020.00966

23. H. Yan, Y. Guo, C. Yang, Source-free unsupervised domain adaptation with surrogate data generation, in *BMVC*, (2021), 198. https://doi.org/10.5244/c.35.324

24. Y. Du, H. Yang, M. Chen, H. Luo, J. Jiang, Y. Xin, et al., Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation, *Mach. Learn.*, **113** (2024), 3611–3631. https://doi.org/10.1007/s10994-023-06432-8

25. J. Liang, D. Hu, J. Feng, Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, in *Proceedings of the 37th International Conference on Machine Learning*, **119** (2020), 6028–6039.

26. S. Yang, J. Van de Weijer, L. Herranz, S. Jui, Exploiting the intrinsic neighborhood structure for source-free domain adaptation, in *Advances in Neural Information Processing Systems*, **34** (2021), 29393–29405.

27. H. Xia, H. Zhao, Z. Ding, Adaptive adversarial network for source-free domain adaptation, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 8990–8999. https://doi.org/10.1109/iccv48922.2021.00888

28. J. Huang, D. Guan, A. Xiao, S. Lu, Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data, in *Advances in Neural Information Processing Systems*, **34** (2021), 3635–3649.

29. S. Qu, G. Chen, J. Zhang, Z. Li, W. He, D. Tao, General class-balanced multicentric dynamic prototype pseudo-labeling for source-free domain adaptation, *Int. J. Comput. Vis.*, **133** (2025), 3327–3348. https://doi.org/10.1007/s11263-024-02335-w

30. J. Liang, D. Hu, Y. Wang, R. He, J. Feng, Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2021), 8602–8617. https://doi.org/10.1109/tpami.2021.3103390

31. K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, et al., Fixmatch: Simplifying semi-supervised learning with consistency and confidence, in *Advances in Neural Information Processing Systems*, **33** (2020), 596–608.

32. B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, et al., Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling, in *Advances in Neural Information Processing Systems*, **34** (2021), 18408–18419.

33. C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, et al., Progressive feature alignment for unsupervised domain adaptation, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 627–636. https://doi.org/10.1109/CVPR.2019.00072

34. K. Tanwisuth, X. Fan, H. Zheng, S. Zhang, H. Zhang, B. Chen, et al., A prototype-oriented framework for unsupervised domain adaptation, in *Advances in Neural Information Processing Systems*, **34** (2021), 17194–17208.

35. Y. Yang, S. Chen, X. Li, L. Xie, Z. Lin, D. Tao, Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network?, in *Advances in Neural Information Processing Systems*, **35** (2022), 37991–38002.

36. J. Hoffman, E. Tzeng, T. Park, J. Y. Zhu, P. Isola, K. Saenko, et al., Cycada: Cycle-consistent adversarial domain adaptation, in *International Conference on Machine Learning*, (2018), 1989–1998.

37. K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in *Computer Vision–ECCV 2010: 11th European Conference*, (2010), 213–226. https://doi.org/10.1007/978-3-642-15561-1_16

38. H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 5385–5394. https://doi.org/10.1109/CVPR.2017.572

39. X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, K. Saenko, Visda: The visual domain adaptation challenge, preprint, arXiv:1710.06924.

40. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., Pytorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems*, **32** (2019), 1–12.

41. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90

42. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (2009), 248–255. https://doi.org/10.1109/CVPR.2009.5206848

43. E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 2962–2971. https://doi.org/10.1109/CVPR.2017.316

44. K. Saito, Y. Ushiku, T. Harada, K. Saenko, Adversarial dropout regularization, preprint, arXiv:1711.01575.

45. Z. Deng, Y. Luo, J. Zhu, Cluster alignment with a teacher for unsupervised domain adaptation, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), 9943–9952. https://doi.org/10.1109/ICCV.2019.01004

46. C. Y. Lee, T. Batra, M. H. Baig, D. Ulbricht, Sliced wasserstein discrepancy for unsupervised domain adaptation, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 10277–10287. https://doi.org/10.1109/CVPR.2019.01053

47. Y. Kim, D. Cho, K. Han, P. Panda, S. Hong, Domain adaptation without source data, *IEEE Trans. Artif. Intell.*, **2** (2021), 508–518. https://doi.org/10.1109/TAI.2021.3110179

48. S. Roy, M. Trapp, A. Pilzer, J. Kannala, N. Sebe, E. Ricci, et al., Uncertainty-guided source-free domain adaptation, in *Computer Vision – ECCV 2022*, (2022), 537–555. https://doi.org/10.1007/978-3-031-19806-9_31

49. S. Yang, Y. Wang, L. Herranz, S. Jui, J. Van de Weijer, Casting a BAIT for offline and online source-free domain adaptation, *Comput. Vis. Image Underst.*, **234** (2023), 103747. https://doi.org/10.1016/j.cviu.2023.103747

50. T. Chu, Y. Liu, J. Deng, W. Li, L. Duan, Denoised maximum classifier discrepancy for source-free unsupervised domain adaptation, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **36** (2022), 472–480. https://doi.org/10.1609/aaai.v36i1.19925

51. N. Ding, Y. Xu, Y. Tang, C. Xu, Y. Wang, D. Tao, Source-free domain adaptation via distribution estimation, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 7202–7212. https://doi.org/10.1109/CVPR52688.2022.00707

52. F. Wang, Z. Han, Y. Gong, Y. Yin, Exploring domain-invariant parameters for source free domain adaptation, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 7141–7150. https://doi.org/10.1109/CVPR52688.2022.00701

53. J. He, L. Wu, C. Tao, F. Lv, Source-free domain adaptation with unrestricted source hypothesis, *Pattern Recognit.*, **149** (2024), 110246. https://doi.org/10.1016/j.patcog.2023.110246

54. M. Jing, J. Li, K. Lu, L. Zhu, H. T. Shen, Visually source-free domain adaptation via adversarial style matching, *IEEE Trans. Image Process.*, **33** (2024), 1032–1044. https://doi.org/10.1109/TIP.2024.3353539

55. L. Zhou, N. Li, M. Ye, X. Zhu, S. Tang, Source-free domain adaptation with class prototype discovery, *Pattern Recognit.*, **145** (2024), 109974. https://doi.org/10.1016/j.patcog.2023.109974

56. S. Yang, Y. Wang, K. Wang, S. Jui, J. Van de Weijer, Attracting and dispersing: A simple approach for source-free domain adaptation, in *Advances in Neural Information Processing Systems*, **35** (2022), 5802–5815. https://doi.org/10.52202/068431-0420

57. J. Lee, D. Jung, J. Yim, S. Yoon, Confidence score for source-free unsupervised domain adaptation, in *International Conference on Machine Learning*, **162** (2022), 12365–12377.

58. L. Yi, G. Xu, P. Xu, J. Li, R. Pu, C. Ling, et al., When source-free domain adaptation meets learning with noisy labels, preprint, arXiv:2301.13381.

59. M. Litrico, A. Del Bue, P. Morerio, Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 7640–7650. https://doi.org/10.1109/cvpr52729.2023.00738

60. S. Tang, A. Chang, F. Zhang, X. Zhu, M. Ye, C. Zhang, Source-free domain adaptation via target prediction distribution searching, *Int. J. Comput. Vis.*, **132** (2024), 654–672. https://doi.org/10.1007/s11263-023-01892-w

61. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, *Int. J. Comput. Vis.*, **128** (2020), 336–359. https://doi.org/10.1007/s11263-019-01228-7

AIMS Press