IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 18, 2021, accepted September 2, 2021, date of publication September 6, 2021, date of current version September 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110605

Semi-Supervised Domain Adaptation Using Explicit Class-Wise Matching for Domain-Invariant and Class-Discriminative Feature Learning

BA HUNG NGO, (Graduate Student Member, IEEE), JAE HYEON PARK, (Student Member, IEEE), SO JEONG PARK, AND SUNG IN CHO, (Member, IEEE)
Department of Multimedia Engineering, Dongguk University, Seoul 04620, South Korea

Corresponding author: Sung In Cho (csi2267@dongguk.edu)

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1C1C1009662) and Korea Institute of Police Technology (KIPoT) grant funded by the Korea government(KNPA) (No.202100200902, AI driving ability test standardization and evaluation process development).

ABSTRACT Semi-supervised domain adaptation (SSDA) is a promising technique for various applications. It can transfer knowledge learned from a source domain having high-density labeled samples to a target domain having limited labeled samples. Several previous works have attempted to reduce the distribution discrepancy between source domain and target domain by using adversarial-based or entropy-based methods. These works have improved the performance of SSDA. However, there are still lacunae in producing class-wise domain-invariant features, which impair the improvement of the classification accuracy in the target domain. We propose a novel mapping function using explicit class-wise matching that can make a better decision boundary in the embedding space for superior classification accuracy in the target domain. In general, in a target domain with low-density label samples, it is more challenging to create a well-organized distribution for the classification than in a source domain where rich label information is available. In our mapping function, a representative vector of each class in the embedding spaces of the source and target domains is derived and aligned by using class-wise matching. It is observed that the distribution in the embedding space of the source domain can be effectively reproduced in the target domain. Our method achieves outstanding accuracy of classification in the target domain compared with previous works on the Office-31, Office-Home, Visda2017 and DomainNet datasets.

INDEX TERMS Semi-supervised learning, domain adaptation, classification, transfer learning, mapping function.

I. INTRODUCTION

Traditional supervised learning approaches are quite effective, but they require sufficient labeled samples to successfully train a model. However, collecting labeled data is often expensive and time-consuming. Domain adaptation has emerged as a new machine learning strategy in which the model is built using a large amount of labeled data from a source domain and a small amount of labeled data (or even none) from the target domain. In general, domain adaptation can reduce the labor cost of re-labeling by utilizing the knowledge learned in the primary domain (source domain) and then transferring that experience to the target domain, which shares common features but has a different distribution. The key issue of domain adaptation is how to approximate the joint distribution of the source domain and target domain, i.e., to predict the labels of unlabeled target data with the minimum prediction error. Domain adaptation is widely used in various real-world applications such as image classification [1]–[4], object detection [5]–[8], semantic segmentation [9]–[11], and person re-identification [12]–[14]. Depending on whether the label information of the

The associate editor coordinating the review of this manuscript and approving it for publication was Guitao Cao.

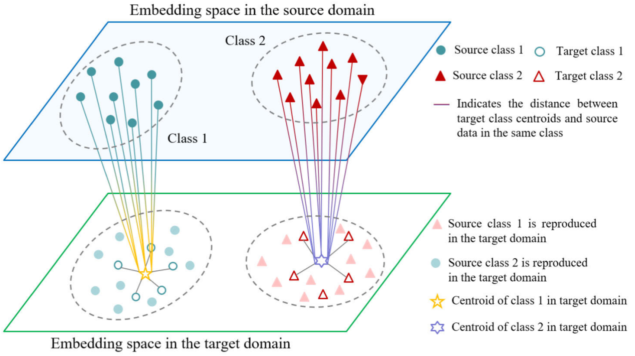

Embedding space in the source domain
Class 2
Class 1
Source class 1
Target class 1
Source class 2
Target class 2
Indicates the distance between target class centroids and source data in the same class
Source class 1 is reproduced in the target domain
Source class 2 is reproduced in the target domain
Centroid of class 1 in target domain
Centroid of class 2 in target domain
Embedding space in the target domain

FIGURE 1. Key idea of the proposed method. The goal of the proposed method is to reproduce a well-organized source distribution on the target domain.

target domain can be used for training, the domain adaptation method is categorized into two subgroups: unsupervised domain adaptation (UDA) and semi-supervised domain adaptation (SSDA).

In UDA [15]–[23], the source samples and unlabeled target samples are integrated for training. The knowledge from the source domain obtained by the supervision training is transferred to the target domain. However, unlabeled target samples that are less correlated with the source samples are less affected by the supervision in the source domain, which leads to inter-domain discrepancy in UDA. By contrast, SSDA [24]–[28] uses extra information by adding a few labeled targets and enforcing the corresponding target feature to be attracted toward source feature clusters, which guarantees partial alignment between the two domain distributions.

These methods help to improve the network performance in the target domains. However, they still show a poor generalization quality because of two main reasons: First, such as the model of S+T [24] is trained by the supervised learning manner with the labeled source and target samples without any information from the unlabeled target data. Therefore, the information from the target domain is exploited inefficiently. Moreover, the number of labeled target samples is much lower than the labeled source samples leading to the biased selection problem. In fact, MME [26] shows that the estimated prototypes of the labeled samples are biased toward the source domain when the vast majority of the labeled source samples and the minority labeled target samples are combined. Second, the knowledge being transferred from the source domain to the target domain also contains weakly-related source representations with the target domain, which can be harmful to the target performance. This phenomenon is known as negative transfer. Some previous works [15]–[17] use adversarial learning inspired by generative adversarial networks (GAN) [40] to mitigate the negative transfer. These methods show a good performance by reducing the domain discrepancy between source and target domains by confusing a domain classifier. However, they ignore class discriminability, leading to a limitation of the target classification performance.

Therefore, in this paper, we propose a novel SSDA method that can reconstruct the well-organized source distribution in the target domain via a proposed mapping function while using limited labeled target data. The concept of the proposed mapping function is shown in Fig. 1. The source distribution is almost perfectly organized, so that classification between classes is easily possible by training on large-scale labeled data. The distribution of the source domain is then reproduced by minimizing the distance between the class centroids in the target domain and source data samples within the same class. The proposed mapping function allows the target domain to actively select the feasible source features to be reproduced. In addition to this, dual feature extractors are used to separately capture features of the source and target domain. Therefore, it can avoid the accumulated error in a single network, and the negative effect from the noise labels in the source domain is mitigated.

Our contributions are summarized as follows:

- First, two feature extractors are used to train separately on the source domain and the target domain. Only the source features that are closely related to the target features are transferred to the target domain. This mitigates the negative transfer problem and accumulated errors from the bias learning.
- Second, a new mapping function is proposed to reconstruct the well-organized distribution of the source domain on the target domain by using explicit class-wise matching for domain-invariant and class-discriminative feature learning.
- Finally, we conduct extensive experiments on the Office-31, Office-Home, Visda2017 and DomainNet benchmark datasets to demonstrate the superiority of our proposed method.

II. RELATED WORK

In this section, we review the existing methods for UDA and SSDA.

A. UNSUPERVISED DOMAIN ADAPTATION

Domain-adversarial training of neural networks (DANN) [15] is a popular training method in UDA, which utilizes an adversarial manner to transfer both domains' data to a common feature space and shares weights between the source and target domains. DANN proposes a gradient reversal layer to reduce the discrepancy between the source and target domains. The adversarial discriminative domain adaptation (ADDA) [16] method uses two convolution neural networks, one each to extract the image features of the source and target. The discrepancy between source and target representations is minimized by using the adversarial adaptive method. Unsupervised Domain Adaptation with Deep Metric Learning (M-ADDA) [18] is an improved version of ADDA, which solves UDA tasks by using a metric-learning-based method. In this method, first, the source model is trained on the source dataset by using triplet loss. Then it works as a reference to

TABLE 1. Main concept, Pros and Cons of existing SSDA methods.

Method	Concept	Pros	Cons
S+T	Simple training strategy using labeled source and target data (without unlabeled target data)	• Easy to implement • Intra-class variation can be reduced	• Domain-shifting and negative transfer due to ignore the unlabeled target information • High intra-domain discrepancy • The target classification accuracy strickly depends on the number of labeled target samples
ENT	Joint training strategy using all available data (including unlabeled target data)	• Easy to implement • Regularization by using the unlabeled target data via the minimum entropy	• Negative transfer • Domain-shifting • High intra-domain discrepancy
MME	Class prototype-based optimization using the minimax loss on the conditional entropy of the unlabeled target samples	• Easy to implement • The distribution gap between source and target domains can be reduced • Between-class variance can be increased, and within-class variance can be reduced	• Negative transfer • The class prototypes are dominated by the source domain • Strictly depending on the accurately estimated prototypes
APE	The *attraction* scheme aims to estimate and minimize the intra-domain discrepancy, the *perturbation* scheme is used to avoid the negative transfer, and the *exploration* aims to a class-aware matching	• Intra-domain discrepancy can be reduced • Effective way to exploit the information from unlabeled target data	• Hard to implement
BiAT	Bidirectional adversarial strategy for reducing domain gap between source and target domains (Adaptive Adversarial Training (AAT) process for source to target domain and Entropy penalized Virtual Adversarial Training (E-VAT) process for target to source domain)	• The features of source and target domains can be exploited efficiently • The domain gap between source and target domain can be significantly reduced	• High computation cost • Hard to implement

train the target model on the target samples through adversarial training, aiming to achieve domain-invariance. In general, the framework of M-ADDA is similar to ADDA except for added triplet loss in the source training term.

B. SEMI-SUPERVISED DOMAIN ADAPTATION

In the SSDA [24]–[28], a few target labels are added, and it works as a bridge to leverage target distribution toward the source distribution. Semi-supervised domain adaptation via minimax entropy (MME) [26] that uses the minimax entropy technique is the most popular method. Specifically, each class in the source domain is represented by a prototype. Then, the classifier is trained to produce the domain-invariant prototype for each class by maximizing the entropy of the softmax prediction output of unlabeled samples in the target domain. The feature extractor is updated by minimizing the entropy on unlabeled samples in the target domain to reduce the distance between them and the class prototype. However, only unlabeled target samples having a close relationship with labeled targets move to the class prototypes. Other unlabeled target samples stay unaligned, which leads to an intra-domain discrepancy problem in the target domain [27]. APE [27] is one of the earlier methods to analyze the target intra-domain discrepancy issue and attempts to resolve it via three schemes, i.e., attraction, perturbation, and exploration. However, APE cannot solve the bias of the decision boundary, which is dominated by the source domain. Bidirectional Adversarial Training (BiAT) [28] exploits the advantages of adversarial learning to enforce the exchange of source and target domain knowledge mutually. In this method, a bidirectional strategy is created using two opposing adversarial learning methods. One approach uses adaptive adversarial training to transfer knowledge from the source domain to the target domain. Another one uses entropy-penalized virtual adversarial training for transferring target knowledge to the source domain.

The main concept, and pros and cons of each existing SSDA method are listed in Table 1.

III. PROPOSED METHOD

A. PROBLEM FORMULATION

In semi-supervised domain adaptation, we are given labeled data from the source domain and a few labeled samples from the target domain. The set of labeled source samples is denoted as $D_s = \{x^s, y^s\} = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where x^s is

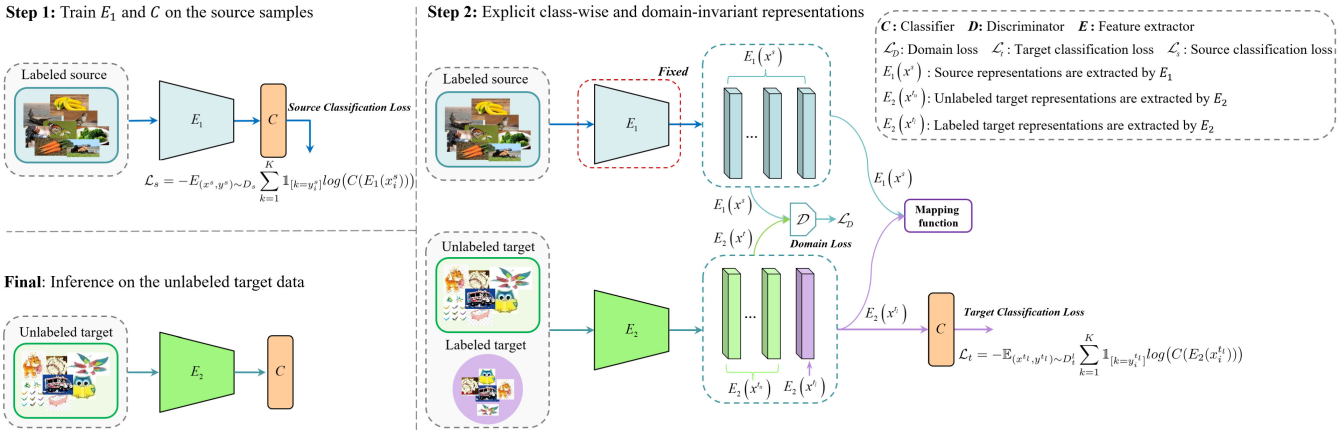

Step 1: Train E_1 and C on the source samples
Labeled source
E_1
C
Source Classification Loss
$\mathcal{L}_s = -E_{(x^s,y^s)\sim D_s}\sum_{k=1}^{K}\mathbb{1}_{[k=y_i^s]}log(C(E_1(x_i^s)))$
Step 2: Explicit class-wise and domain-invariant representations
Labeled source
Fixed
E_1
$E_1(x^s)$
$E_1(x^s)$
$E_1(x^s)$
$E_2(x^t)$
$\mathcal{D}$
$\mathcal{L}_D$
Domain Loss
Mapping function
Unlabeled target
Labeled target
E_2
$E_2(x^{t_u})$
$E_2(x^{t_l})$
$E_2(x^{t_l})$
C
Target Classification Loss
$\mathcal{L}_t = -\mathbb{E}_{(x^{t_l},y^{t_l})\sim D_t^l}\sum_{k=1}^{K}\mathbb{1}_{[k=y_i^{t_l}]}log(C(E_2(x_i^{t_l})))$
C: Classifier **D:** Discriminator **E:** Feature extractor
$\mathcal{L}_D$: Domain loss $\mathcal{L}_t$: Target classification loss $\mathcal{L}_s$: Source classification loss
$E_1(x^s)$: Source representations are extracted by E_1
$E_2(x^{t_u})$: Unlabeled target representations are extracted by E_2
$E_2(x^{t_l})$: Labeled target representations are extracted by E_2
Final: Inference on the unlabeled target data
Unlabeled target
E_2
C

FIGURE 2. **Architecture of the proposed method for training and inference stages. The domain adaptation problem is handled by using explicit class-wise matching and domain-invariant representations.**

TABLE 2. **Important symbols in the proposed method.**

Notation	Description
x^s, y^s	The set of source samples and their labels
x_i^s, y_i^s	The *i*-th element of source sample and its label
x^{t_l}/x^{t_u}	A set of label/unlabeled target samples
y^{t_l}	A set of target labels
D_t^l	A set of labeled target samples x^{t_l} and their labels y^{t_l}
D_t^u	A set of unlabeled target samples x^{t_u}
$D_t^{l,k}$	The set of labeled target samples with class *k*
c_k^t	The centroid of class *k*
n_s	The number of source samples
n_t^l/n_t^u	The number of labeled/unlabeled target samples
$n_t^{l,k}$	The number labeled target samples in class *k*
E_1/E_2	The source/target feature extractor
C	The source/target classifier

the set of source samples, x_i^s is the *i*-th element in this set, y^s is the label vector, y_i^s is its *i*-th component, and n_s is the number of source images. The labeled target samples are denoted as $D_t^l = \{x^{t_l}, y^{t_l}\} = \{(x_i^{t_l}, y_i^{t_l})\}_{i=1}^{n_t^l}$, where x^{t_l} is the group of labeled target samples, $x_i^{t_l}$ is the *i*-th labeled target sample, $y_i^{t_l}$ is its category label, and n_t^l is the number of labeled target samples. An unlabeled target set is denoted as $D_t^u = \{(x_i^{t_u})\}_{i=1}^{n_t^u}$, where $x_i^{t_u}$ is the *i*-th unlabeled target sample, and n_t^u is the number of unlabeled target samples. The total target dataset is denoted as $D_t = D_t^l \cup D_t^u$. All target images are denoted as $x^t = x^{t_l} \cup x^{t_u}$. Table 2 summarizes the important symbols used to explain the proposed method.

The core idea of our proposed method is to establish the mapping function that can reproduce a well-organized source distribution on the target domain with few labeled target samples by using class-wise matching (shown in Fig. 1) for domain-invariant and class-discriminative feature learning.

B. STEP 1: SUPERVISED TRAINING ON THE SOURCE DOMAIN

In step 1, we train the source feature extractor E_1 and classifier C, as shown in Fig. 2, by minimizing the standard cross-entropy loss with K classes on the source samples (x^s) and their corresponding labels y^s in a supervised manner as follows:

$$\mathcal{L}_s = -E_{(x^s,y^s)\sim D_s}\sum_{k=1}^{K}\mathbb{1}_{[k=y_i^s]}log\big(C(E_1(x_i^s))\big), \qquad (1)$$

where $\mathbb{1}_{[.]}$ is an indication function whose value is 1 if the input [] is true, otherwise 0. At the end of this step, the source distribution in the embedding space can be well-organized for the classification because it can utilize the rich labeled samples for the training.

C. STEP 2: EXTRACTION OF DOMAIN-INVARIANT REPRESENTATIONS AND EXPLICIT CLASS-WISE MATCHING

1) *Extraction of domain-invariant representations*: During step 2, the pre-trained feature extractor E_1 is fixed to extract the domain representation from the source domain. Similar to DANN [15], the feature extractor E_2 captures the target domain feature from the target samples and then its parameters are updated to minimize the domain discrepancy between the source and target domains by fooling the domain classifier D as follows:

$$\begin{aligned}\min_{E_2}\max_{\mathcal{D}} \mathcal{L}(\mathcal{D}, E_1, E_2) &= \mathbb{E}_{x^s\sim D_s}\big[log(\mathcal{D}(E_1(x^s)))\big] \\ &+ \mathbb{E}_{x^t\sim D_t}\big[log(1-\mathcal{D}(E_2(x^t)))\big], \end{aligned} \qquad (2)$$

where $\min_{E_2}\max_{D} \mathcal{L}(\mathcal{D}, E_1, E_2)$ is the domain loss and is indicated as $\mathcal{L}_d$. Overall, only the domain features of source and target domains are aligned by adopting (2) but the important information in each class cannot be considered. Thus, this process slightly improves the performance of the target classification. There is still a shortcoming in producing class-wise domain-invariant features, which prevents the classification accuracy in the target domain from improving.

Therefore, we propose a novel mapping function that uses explicit class-wise matching to establish a better decision boundary in the embedding space of the target domain based on the well-organized source distribution where rich label information is available. In the proposed mapping function, a representative vector of each class in the embedding spaces of the source and target domains is derived and aligned by using the proposed class-wise matching.

2) *Class-wise matching*: The feature extractor E_2 and classifier C are trained using the supervised learning method on the few labeled target samples as follows:

$$\mathcal{L}_t = -\mathbb{E}_{(x^{t_l}, y^{t_l})\sim D_t^l} \sum_{k=1}^{K} \mathbb{1}_{[k=y_i^{t_l}]} log\big(C(E_2(x_i^{t_l}))\big). \tag{3}$$

Using this, the feature extractor E_2 can correctly extract the unique characteristics in the target domain. Then, we compute a centroid c_k^t of the k-th class of target domain in the embedding space and c_k^t is indicated as in Fig. 1. Each class centroid is calculated by taking a mean vector of feature vectors that belong to the same class as follows:

$$c_k^t = \frac{1}{n_t^{l,k}} \sum_{x_i^{t_l} \in D_t^{l,k}} f(x_i), \tag{4}$$

where $D_t^{l,k}$ and $n_t^{l,k}$ denote the set of labeled target images and number of labeled target samples with class k, respectively. $f(x_i)$ is the feature vector of x_i. The class centroids represent the features of each class in the target domain. For each class centroid denoted in (4), we compute the distances from x_i^s, then produce the sample-to-centroid distance over K classes in cross-domain via a softmax function as follows:

$$P_{s\to t}\big(y = k \mid x_i^s\big) = \frac{exp\big(d(f(x_i^s), c_k^t)\big)}{\sum_{j=1}^{K} exp\big(d(f(x_i^s), c_j^t)\big)}, \tag{5}$$

where $d\big(.,.\big)$ is the function of Euclidean distance between source samples and class centroids of target data, and $P_{s\to t}\big(y = k \mid x_i^s\big)$ is the probability x_i^s belonging to class k in the target domain. This procedure is shown in Fig. 1 and implemented as shown in Fig. 2. The parameters of target feature extractor E_2 are optimized to minimize the distance between the location of each sample in the source domain and its corresponding c_k^t by minimizing the following cross-entropy loss:

$$\mathcal{L}_{s\to t} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{k=1}^{K} \mathbb{1}_{[k=y_i^s]} log P_{s\to t}(y = k \mid x_i^s). \tag{6}$$

D. STEP 3: SELF-TRAINING FOR ROBUSTNESS PREDICTION ON UNLABELED TARGET DATA

Given the lack of information in the target domain owing to the limited number of labeled target samples in the semi-supervised learning, many previous works [29]–[31] found ways to exploit the information from the unlabeled target samples that closely correlate with the labeled target sample. They showed their effectiveness by using data augmentation [32] and consistency regularization with pseudo-labeling [31]. Inspired from the current SOTA method [31], as shown in Fig. 3, weak augmentation and strong augmentation are applied to the unlabeled images before feeding them to the feature extractor E_2 which was trained in step 2 with limited labeled samples. While weak augmentation is a simple transformation such as flipping and blurring on images, strong augmentation is borrowed from RandAugment [32], which uses random augmentation techniques including rotation, polarization, brightness, and color variations on an input image. The prediction vectors of a weak augmented image and a strong augmented image can be defined as follows:

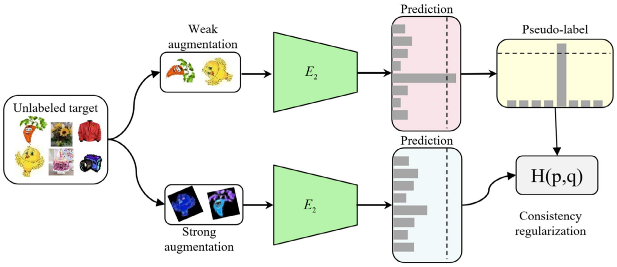


FIGURE 3. Diagram for pseudo labeling and the consistency regularization on the unlabeled target samples [31].

$$\begin{aligned} p_{x_i^{tu}}^{weak} &= softmax\big(C(E_2(x_i^{t_u} + \sigma))\big), \\ p_{x_i^{tu}}^{strong} &= softmax\big(C(E_2(x_i^{t_u} + \delta))\big). \end{aligned} \tag{7}$$

The pseudo labels of unlabeled samples are generated by taking the probability of prediction values of weak augmentation, $x + \sigma$. Then, consistency regularization is conducted by minimizing the cross-entropy of the prediction of a strong augmented image $x + \delta$ and its pseudo label. At this time, the model is very sensitive to incorrect pseudo labels. Therefore, only the prediction $p_{x_i^{tu}}^{weak}$, in which over the given threshold value ($\max p_{x_i^{tu}}^{weak} > \tau$), τ is the threshold value (the detailed process to select the optimal τ is showed in IV. D), is selected to sort out incorrect pseudo labels. Then, the regularization cost for the unlabeled target sample with a high confident pseudo label is computed as follows:

$$\mathcal{L}_u(x_i^{t_u}) = \mathbb{1}\big[\max p_{x_i^{tu}}^{weak} > \tau\big] H\big(\max(p_{x_i^{tu}}^{weak}), p_{x_i^{tu}}^{strong}\big), \tag{8}$$

where $\mathbb{1}_{[.]}$ is an indication function and $H(.,.)$ is the cross-entropy.

[26] and [41] show a way to successfully cluster the features of the unlabeled target data. They minimize conditional entropy measured using the similarity between the weight vector of the classifier, which represents a certain class, and unlabeled target features. This is calculated as follows:

$$H = -\mathbb{E}_{x^{tu}\sim D_t^u} \sum_{k=1}^{K} P(y = k \mid x^{tu}) log P(y = k \mid x^{tu}), \tag{9}$$

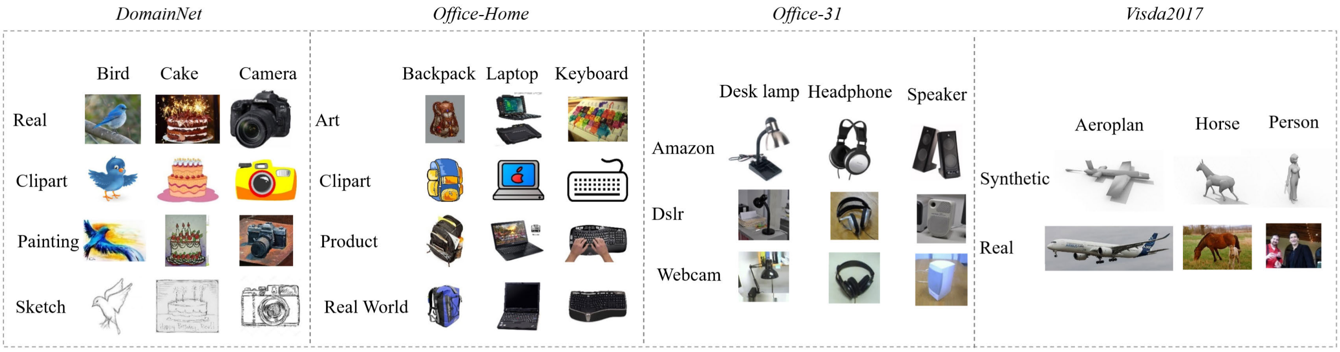


FIGURE 4. Example images in ***DomainNet***, ***Office-Home***, ***Office-31***, and ***Visda2017*** datasets.

TABLE 3. Description of datasets for the experiments.

Datasets	Domains	Type	Instance	Classes
Office-31	DSLR	Object	498	31
	Webcam		795	
	Amazon		2,817	
Office-Home	Artistic	Object	2,427	65
	Clip Art		4,365	
	Product		4,439	
	Real-World		4,357	
DomainNet	Clipart	Object	48,837	345
	Infograph		53,201	
	Painting		75,759	
	Quickdraw		172,500	
	Real		175,327	
	Sketch		70,386	
Visda2017	Synthetic	Object	152,397	12
	Real		55,388	

where $P(y = k \mid x^{t_u})$ represents the probability of x^{t_u} belonging to class k, namely the k-th dimension of softmax score vector $P(y = k \mid x^{t_u}) = \sigma(C(E_2(x^{t_u})))$. The classifier is trained to update its weight vectors by maximizing the entropy on the unlabeled target data, while the feature extractor is trained to generate the unlabeled target feature more similar to the updated weight vector by minimizing the entropy. Following this, the total cost functions used for training the feature extractor E_2 and classifier C are computed as follows:

$$\mathcal{L}_{E_2} = \mathcal{L}_d + \mathcal{L}_t + \mathcal{L}_{s\to t} + \frac{1}{n_t^u}\sum_{i=1}^{n_t^u}\mathcal{L}_u\big(x_i^{t_u}\big) + \lambda H \quad (10)$$

$$\mathcal{L}_C = \mathcal{L}_t + \frac{1}{n_t^u}\sum_{i=1}^{n_t^u}\mathcal{L}_u\big(x_i^{t_u}\big) - \lambda H, \quad (11)$$

where λ is a hyper-parameter used to balance between minimax entropy and supervision losses and will be explained in section IV. 4.

TABLE 4. Domain adaptation results (accuracy %) on ***Office-31*** dataset.

Net	Method	W→A		D→A		MEAN	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
AlexNet	S+T	50.4	61.2	50.0	62.4	50.2	61.8
	DANN	57.0	64.4	54.5	65.2	55.8	64.8
	CDAN	50.2	60.3	50.9	61.4	49.5	60.9
	ENT	50.4	64.0	48.5	66.2	50.4	65.1
	MME	50.7	67.3	50.0	67.8	56.5	67.6
	APE	-	67.6	-	69.0	-	68.3
	BiAT	57.2	68.2	55.8	68.5	56.3	68.4
	Ours	**58.7**	**71.5**	**58.8**	**71.4**	**58.8**	**71.5**

The components in (10) are summarized as: $\mathcal{L}_{E_2}$ and $\mathcal{L}_C$ are the costs used to train the feature extractor E_2 and classifier C, respectively. They consist of elements such as: $\mathcal{L}_d$ is the domain loss to minimize the discrepancy between source and target domains. $\mathcal{L}_t$ is the classification loss on the labeled target samples, which is computed by the standard entropy minimization. $\mathcal{L}_{s\to t}$ described in (6) is the mapping function loss, which is used to minimize the distance between the source samples and the class centroid of the target domain within the same class. $\mathcal{L}_u$ is the consistency regularization loss that was explained in (8). H is the conditional entropy which was described in (9).

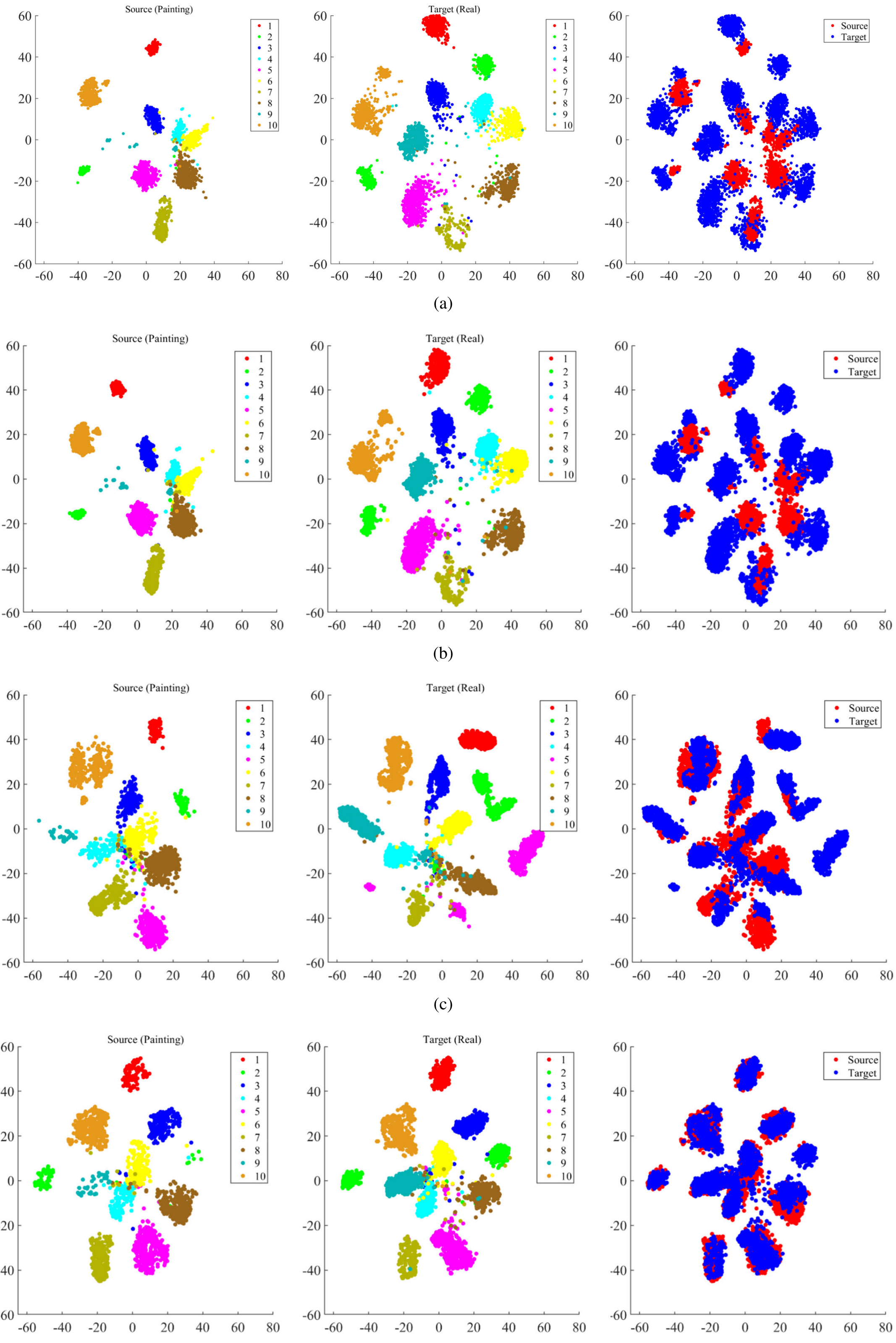
E. INFERENCE ON THE TARGET DATASET

In this step, by using feature extractor E_2 and classifier C, class prediction $y_{predict}$ on the target domain is given as:

$$y_{predict} = \underset{x_t^u \sim D_t^u}{\arg\max}\,\big(C(E_2(x_i^{t_u}))\big). \quad (12)$$

IV. EXPERIMENTS

In this section, first, benchmark datasets for experiments are described. Then, baseline and implementation details, results, and comparison are provided. Finally, we analyze the effectiveness of the proposed method based on some ablation studies.



(a)

(b)

(c)

(d)

FIGURE 5. Visualization of source and target features with t-SNE [39]. We plotted the features of ten classes on the source and target domains of (a) S+T [24], (b) ENT [25], (c) MME [26], and (d) Our method on *DomainNet* dataset with a scenario *Painting* to *Real*. Each class was represented by different colors. The left column illustrated the source distribution. The middle column showed the output features on the target domain. On the right, the features of the source and target domains were aligned to measure the gap between them to evaluate the efficiency of adaption methods. The features in the proposed method were well-aligned in the two domains compared with S+T, ENT, and MME methods.

A. DATASETS AND EXPERIMENT SETTINGS

1) *Office-31* [33] is a popular dataset that contains 4110 images belonging to 31 classes from three domains: Amazon (A), Webcam (W), and DSLR (D).

2) *Office-Home* [34] is a standard benchmark dataset for domain adaptation containing 15,500 images belonging to 65 categories, forming four domains: Artistic (Art), Clipart (Cl), Product (Pr), and Real-World (Rw), which represent artistic depictions for object images, picture collection of clipart, object images with a clear background, and object images collected with a regular camera, respectively.

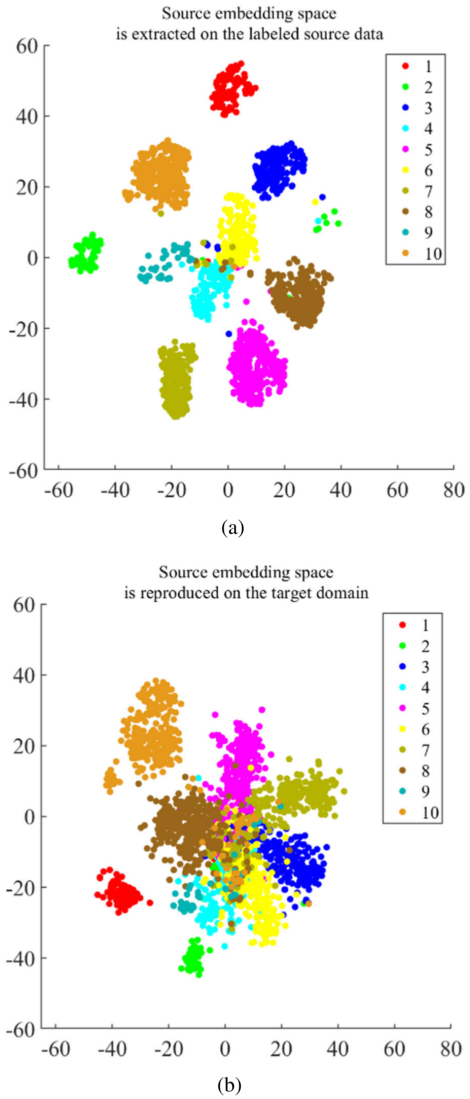
3) *DomainNet* [35] is a benchmark dataset for large-scale domain adaptation, which consists of six domains of 345 categories. For a fair comparison with the previous SSDA methods, we selected Real (R), Clipart (C), Painting (P), and Sketch (S) as the four evaluation domains and performed the following cross-domain evaluations: R←C (adaptation from source Real to target Clipart), R←P, P←C, C←S, S←P, R←S, and P←R with 126 classes. For each set of cross-domain experiments, we evaluated classification accuracy in the target domain with varying cases such as 1-shot, 3-shot, 5-shot, and 10-shot settings, where 1, 3, 5, and 10 are the number of available labeled target samples, respectively.

4) *Visda2017* [42] dataset consists of 55,388 Real images (R) and 152,397 Synthetic (Syn) images from 12 categories. Synthetic samples worked as the source domain, and Real samples were used for the target domains. We randomly selected three Real images in each of 12 categories for 3-shot setting to conduct SSDA experiments.

All results for comparison of *Office31*, *Office-Home*, and *DomainNet* datasets were collected from previous works [26], [27], and [28] based on ResNet-34 backbone. Except for results of *Visda2017* dataset, we ran them ourselves by using codes released by authors. [1,2]

A list of domains and classes in benchmark datasets for our experiments were presented in Table 3 and example images of datasets are shown in Fig. 4.

5) *Implementation details*: We adopted AlexNet [36] and ResNet-34 [37] as the backbone networks for SSDA. The number of images in each mini-batch were computed by $N \times (m + k)$, where N is the number of classes, m is the number of samples in each selected class of the source domain, and k is the labeled target samples in the target domain. For example, in our experiments, we set $N = 10$, $m = 10$, and $k = 3$ to implement a 3-shot setting. The indexes of 10 classes were selected randomly, in each class contains ten labeled images from the source domain and three labeled images from the target domain. N and m were maintained, and k could be adjusted depending on the shot setting. For instance, the values of k could be set at 1, 3, 5, or 10 corresponding to 1-shot, 3-shot, 5-shot, and 10-shot settings, respectively. In addition, the k labeled target images in each class were fixed during training. We used the Stochastic Gradient Descent (SGD) optimizer. The learning rate was computed by using the following formula: $\eta = (\eta_0/((1 + 10p)^{0.75}))$, where $\eta_0 = 0.01$ is an initial learning rate, $p = [0, 1]$ was the training progress. It was adjusted during the stochastic gradient descent (SGD) as following the strategy used in [15]. The weight decay was set as 0.0005, the momentum was 0.9. λ in (10) was set as 0.1. All implementations were done in PyTorch [38] and on a GeForce RTX3090 GPU.



(a)

(b)

FIGURE 6. Visualization of source features with t-SNE [39]. (a) showed source features extracted by supervision learning on the labeled source data, and (b) illustrated the reproduced source distribution on the target domain by using the mapping function.

6) *Comparison*: We compared our proposed method with seven recent approaches: S+T [24], DANN [15], CDAN [20], ENT [25], MME [26], APE [27], and BiAT [28]. For fair comparison, DANN and CDAN were modified to train on the labeled source, limited labeled target, and unlabeled target samples.

B. RESULT ANALYSIS

1) *Results on Office-31 and Office-Home*: Tables 4 and 5 reported the results of all methods on *Office-31* and

[1] https://github.com/VisionLearningGroup/SSDA_MME

[2] https://github.com/TKKim93/APE

TABLE 5. Domain adaptation results (accuracy %) for 12 scenarios on 3-shot settings on *Office-Home* dataset.

Net	Method	R→C	R→P	R→A	P→R	P→C	P→A	A→P	A→C	A→R	C→R	C→A	C→P	MEAN
AlexNet	S+T	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57.0	54.3	37.5	57.9	50.0
	DANN	47.2	66.7	46.6	58.1	44.4	36.1	57.2	39.8	56.6	54.3	38.6	57.9	50.3
	CDAN	41.8	69.9	43.2	53.6	35.8	32.0	56.3	34.5	53.5	49.3	27.9	56.2	46.2
	ENT	44.9	70.4	47.1	60.3	41.2	34.6	60.7	37.8	60.5	58.0	31.8	63.4	50.9
	MME	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
	APE	51.9	74.6	51.2	61.6	47.9	42.1	65.5	44.5	60.9	58.1	44.3	64.8	55.6
	Ours	**53.8**	**75.5**	**54.9**	**65.3**	**49.2**	**44.2**	**67.7**	**46.5**	**64.4**	**60.8**	**46.8**	**66.2**	**57.9**
ResNet-34	S+T	55.7	80.8	67.8	73.1	53.8	63.5	73.1	54.0	74.2	68.3	57.6	72.3	66.2
	DANN	57.3	75.5	65.2	69.2	51.8	51.8	68.3	54.7	73.8	67.1	55.1	67.5	63.5
	CDAN	61.4	80.7	67.1	76.8	58.1	61.4	74.1	59.2	74.1	70.7	60.5	74.5	68.2
	ENT	62.6	85.7	70.2	79.9	60.5	63.9	79.5	61.3	79.1	76.4	64.7	79.1	71.9
	MME	64.6	85.5	71.3	80.1	64.6	65.5	79.0	63.6	79.7	76.6	67.2	79.3	73.1
	APE	66.4	86.2	73.4	82.0	65.2	66.1	81.1	63.9	**80.2**	76.8	66.6	79.9	74.0
	Ours	**69.8**	**88.1**	**75.6**	**82.1**	**67.1**	**71.5**	**82.6**	**66.3**	80.1	**77.4**	**69.2**	**80.8**	**75.9**

TABLE 6. Domain adaptation results (accuracy %) for 7 scenarios on 1-shot and 3-shot settings on *DomainNet* dataset.

Net	Method	R→C		R→P		P→C		C→S		S→P		R→S		P→R		MEAN	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
AlexNet	S+T	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	55.8	58.7	40.0	43.4
	DANN	43.3	46.1	41.6	43.8	39.1	41.0	35.9	36.5	36.9	38.9	32.5	33.4	53.6	57.3	40.4	42.4
	CDAN	43.1	46.2	41.4	44.4	39.3	43.6	32.8	36.4	33.1	38.9	29.1	32.4	55.9	57.3	39.2	42.7
	ENT	37.0	45.5	35.6	42.6	26.8	40.4	18.9	31.1	15.1	29.6	18.0	29.6	52.2	60.0	29.1	39.8
	MME	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	60.7	44.2	48.2
	APE	47.7	54.6	49.0	50.5	46.9	52.1	38.5	42.6	38.5	42.2	33.8	38.7	57.5	61.4	44.6	48.9
	BiAT	54.2	58.6	49.2	50.6	44.0	52.0	37.7	41.9	39.6	42.1	37.2	42.0	56.9	58.8	45.5	49.4
	Ours	**56.1**	**61.4**	**56.2**	**58.6**	**53.3**	**60.5**	**44.5**	**48.4**	**49.4**	**53.4**	**43.7**	**47.5**	**66.0**	**67.6**	**52.7**	**56.8**
ResNet-34	S+T	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
	DANN	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
	CDAN	65.0	69.0	64.9	67.3	63.7	68.4	53.1	57.8	63.4	65.3	54.5	59.0	73.2	78.5	62.5	66.5
	ENT	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
	MME	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
	APE	70.4	76.6	70.8	72.1	72.9	76.7	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
	BiAT	73.0	74.9	68.0	68.8	71.6	74.6	57.9	61.5	63.9	67.5	58.5	62.1	77.0	78.6	67.1	69.7
	Ours	**75.3**	**77.0**	**74.0**	**75.0**	**74.3**	**77.0**	**65.8**	**69.5**	**73.0**	**73.3**	**67.5**	**69.2**	**81.7**	**83.3**	**73.1**	**74.9**

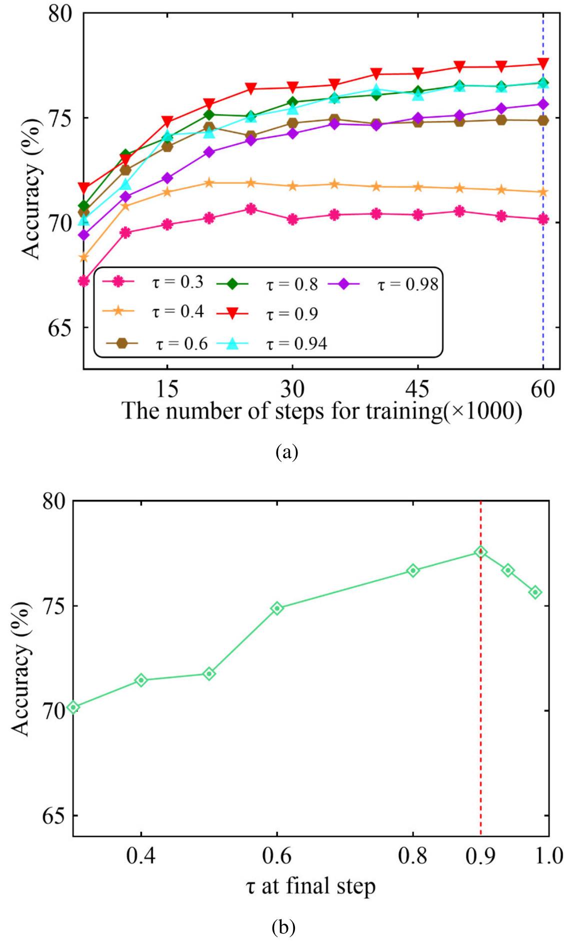
TABLE 7. Domain adaptation results (accuracy %) for 7 scenarios on 5-shot and 10-shot settings on *DomainNet* dataset.

Net	Method	R→C		R→P		P→C		C→S		S→P		R→S		P→R		MEAN	
		5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot
ResNet-34	S+T	64.5	68.5	63.1	66.4	64.2	69.2	59.2	64.8	60.4	64.2	56.2	60.7	75.7	77.3	63.3	67.3
	DANN	63.7	70.0	62.9	64.5	60.5	64.0	55.0	56.9	59.5	60.7	55.8	60.5	72.6	75.9	61.4	64.6
	CDAN	68.0	69.3	65.0	65.3	65.5	64.6	58.0	57.5	62.8	61.6	58.4	60.2	74.8	77.0	64.6	65.1
	ENT	77.1	79.0	71.0	72.9	75.7	78.0	61.9	68.9	66.2	68.4	64.6	68.1	81.1	82.6	71.1	74.0
	MME	75.5	77.1	70.4	71.9	74.0	76.3	65.0	67.0	68.2	69.7	65.5	67.8	79.9	81.2	71.2	73.0
	APE	77.7	79.8	73.0	75.1	76.9	78.9	67.0	70.5	71.4	73.6	68.8	70.8	80.5	82.9	73.6	76.8
	Ours	**78.4**	**82.6**	**76.2**	**77.6**	**78.0**	**82.3**	**71.5**	**74.5**	**74.1**	**76.7**	**70.9**	**73.0**	**83.9**	**86.8**	**76.1**	**79.1**

Office-Home, respectively. The proposed method showed the best performance in all the scenarios. On the *Office-31* dataset, considering the results in 1-shot as well as 3-shot settings, our method also reported outstanding performance when using AlexNet backbone. On *Office-Home* dataset, the average classification accuracy on the target domain of our method was higher than MME and APE, i.e., 2.7% and 2.3% with AlexNet backbone and 2.8% and 1.9% with ResNet-34 backbone in the 3-shot setting.

TABLE 8. Domain adaptation results (accuracy %) for 3-shot setting on *Visda2017* dataset from *Synthetic* to *Real*.

Net	Method	Plane	Bicycle	Bus	Car	Horse	Knife	Motorcycle	Person	Plant	Skboard	Train	Truck	MEAN
ResNet-34	S+T	86.0	55.6	62.4	55.0	60.6	55.5	79.1	52.2	68.1	59.9	82.2	38.0	62.9
	ENT	90.2	54.4	72.5	76.1	86.3	95.5	89.1	76.4	84.8	47.9	87.7	7.1	72.3
	MME	87.3	65.1	74.0	72.1	83.2	89.8	93.0	78.2	90.8	70.5	87.0	34.0	77.1
	APE	94.9	68.8	**87.4**	**81.1**	91.7	86.4	88.6	80.1	91.7	81.4	82.1	6.5	78.4
	Ours	**97.3**	**85.7**	87.1	75.0	**96.4**	**89.0**	**94.3**	**85.3**	**95.6**	**87.9**	**89.8**	**57.4**	**86.7**



(a)

(b)

FIGURE 7. (a) The target classification accuracy depended on threshold values in (8). (b) Experiments found the optimal threshold value in case *Real* to *Clipart* on the *DomainNet* dataset under the 3-shot setting.

2) *Results on DomainNet*: Table 6 presented the classification accuracy of the proposed and benchmark methods on *DomainNet* dataset for 1-shot and 3-shot settings. In experiments using ResNet-34 as the backbone network, the mean accuracy of our method in 1-shot and 3-shot settings was higher than S+T, i.e., 16.2% and 14.9%, respectively. Compared with APE, our method obtained notable accuracy improvements in 1-shot and 3-shot settings. In experiments using the AlexNet backbone, our method reported that the average classification accuracy on the target domain was higher than BiAT, up to 7.2% and 7.6%, respectively, in 1-shot and 3-shot settings.

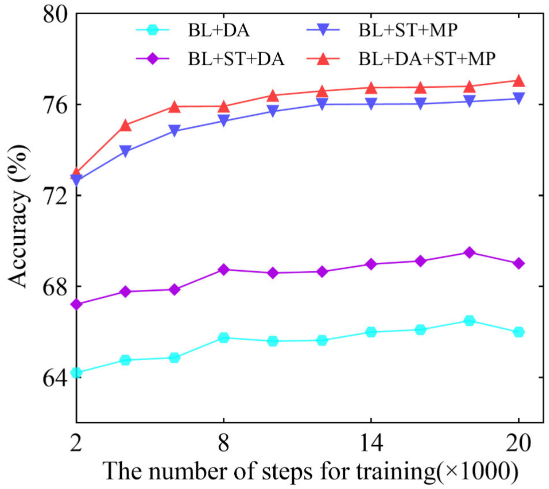


FIGURE 8. Ablation study for the adaptation from *Real* to *Clipart* on the *DomainNet* dataset.

To prove the efficiency of our proposed method for various few-shot cases, we additionally conducted experiments in 5-shot and 10-shot settings with the ResNet-34 backbone. As can be observed in Table 7, compared with APE, which is the SOTA method for SSDA, our method improved performances by 2.5% and 2.3% in 5-shot and 10-shot cases. With the same settings, the proposed method provided higher mean accuracy than S+T, up to 12.8% and 11.8%, respectively.

3) *Results on Visda2017*: We extensively evaluated the proposed method on *Visda2017* dataset. The detailed comparison results of our method and the state-of-the-art SSDA methods were listed in Table 8. The proposed method achieved the best mean accuracy, 86.7%, and gained 8.3% better than APE 28]. The S+T [24] showed the lowest results among the existing SSDA methods because the model of S+T was trained without using the unlabeled target data. In contrast, other methods tried to exploit the information from the target domain via the unlabeled target data. The classification performance of ENT [25] was lower than MME [26] because MME operated with minimization and maximization entropy terms of the unlabeled target data while ENT simply used only the minimum entropy regularization on the unlabeled target data.

C. FEATURE VISUALIZATION

In Fig. 5, we showed the extracted features of source and target domains with t-SNE [39] on *DomainNet* dataset with a scenario P to R using the ResNet-34 backbone. The left-side figures visualized the results of the distribution of source features. A different color was used for denoting each class. The middle images showed the output features on the target data. Then, the features of the source and target domains were aligned to measure the gap between them to evaluate the efficiency of adaption methods. They were displayed in the right-side figures. The red color represented the features of the source domain while the blue color indicated the features of the target domain. This figure showed that features of both domains extracted by our method were well-aligned compared to other benchmark methods. The extracted results in the embedding space of two methods, S+T and ENT, were relatively similar because they used the same strategy to train their models, which is the cross-entropy loss on the mixed labeled data of source and target. While our method was successful at reproducing a well-organized source distribution on the target domain through the proposed mapping function (MP), MME using a minimax entropy-based approach provided a worse alignment compared with the proposed method.

Figure 6 (a) showed the well-clustered features of the source distribution extracted by the feature extractor E_1 on the source data. Figure 6 (b) illustrated the reproduced version of the source distribution on the target domain when the same data was extracted by the feature extractor E_2. Overall, it was proven that the proposed mapping function worked effectively.

D. ABLATION STUDY

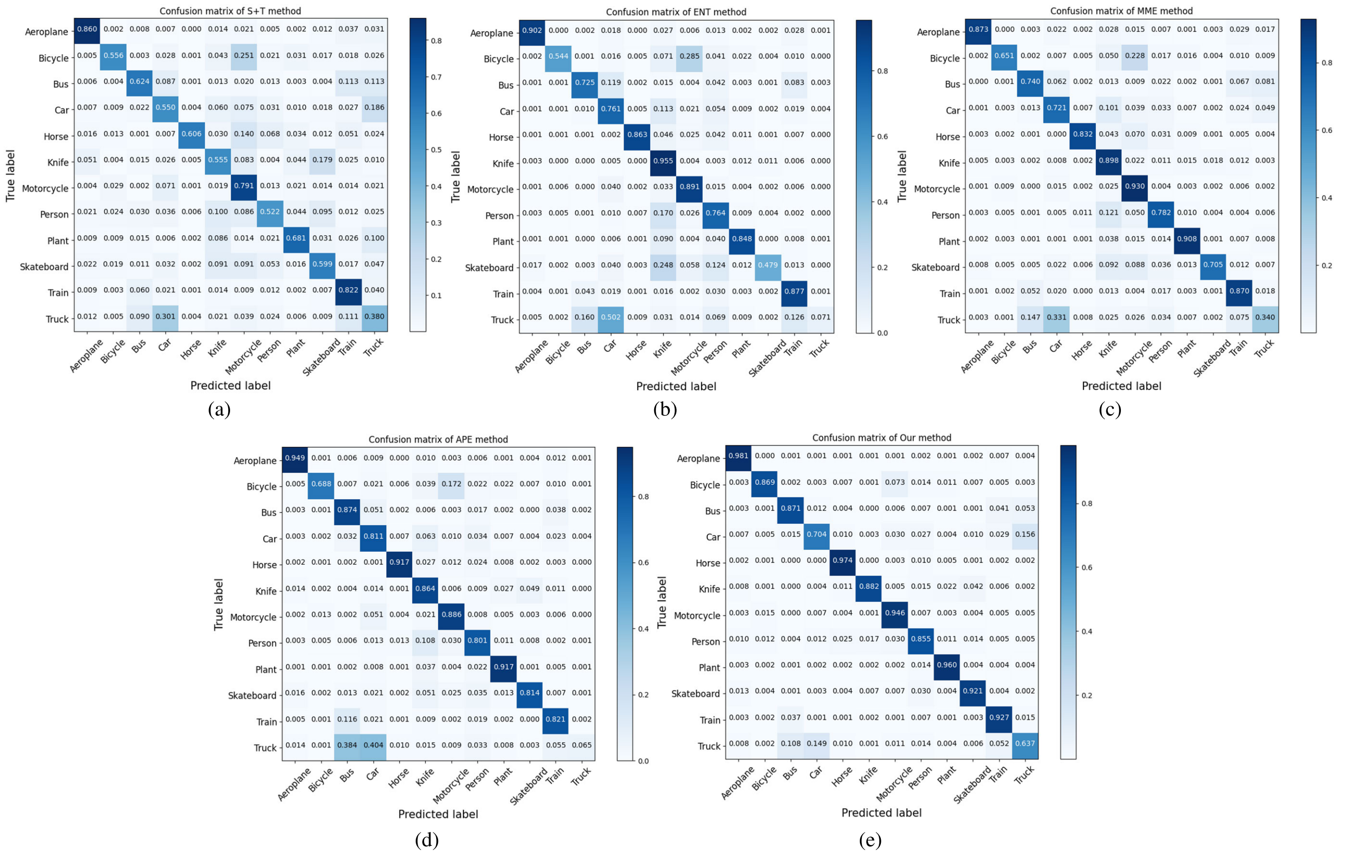
1) *Sensitiveness of threshold value for pseudo labeling*: During the pseudo labeling in step 2, the correct pseudo labels for unlabeled target samples can considerably increase the accuracy of classification. Figures 7 (a) and (b) showed the results of the adaptation from R to C on the *DomainNet* dataset under the 3-shot setting to analyze the sensitivity of the network performance with varying thresholds τ in (8) for the pseudo labeling.

Figure 7 (a) expressed the variation of inference accuracy depending on different τs. Figure 7 (b) showed the inference accuracy at the final training step denoted by a blue dashed line in Fig. 7 (a) to find the optimal τ. In Fig. 7 (a), when the τ is too small, such as $\tau = 0.3 \sim 0.6$, the inference accuracy changed negligibly or even decreased, as the model suffered from a negative effect due to incorrect pseudo labels. When τ was set to 0.94 or 0.98, the pseudo labels used for training could be chosen very strictly, leading to the classifier ignoring useful information. As shown in Fig. 7 (b), the final τ value was set to 0.9, indicated by a red dashed line. Furthermore, in Fig. 7 (a), the inference accuracy increased steadily with $\tau = 0.9$, while other cases started to decrease. However, we limited the number of iterations for ablation studies to fair comparison with the previous works.

2) *Impact of each component on the target learner*: In this portion, we analyzed the impact of components applied in our framework, including the domain adaptation module (DA, section III. C. 1), self-training (ST, section III. D), and the proposed mapping function (MP, section III. C. 2). The baseline (BL) was built by adding a feature extractor to the MME [26] architecture without the above three components. The results of various scenarios were displayed in Fig. 8, in which we studied the tendency of change of the target classification accuracy. Therefore, we could evaluate the impact of each component in the proposed framework. First, we analyzed the impact of DA on the baseline framework. In this case, the cost function was computed as the sum of (1), (2), (3), and (9). In this figure, the performance of the baseline with the domain adaptation module showed poor classification accuracy; just over 65%. This is lower than the result reported in MME. However, when we used SL, the loss function was identified by (1), (2), (3), (8), and (9). The classification accuracy on the target domain improved up to over 69%. This is easy to understand because the SL could support the proposed network to exploit the useful information effectively from the unlabeled target samples by establishing the relationship between labeled target samples and unlabeled target samples. Only when the MP was applied, the classification performance improved significantly. This is because the proposed MP successfully imitated the well-established source distribution built on the high-density labeled samples to the target domain as illustrated in Fig. 5 (d) and Fig. 6 (a). Thus, the inference accuracy when the baseline was combined with (ST+MP, cost function calculated by (1), (3), (8), (6), and (9)) and (DA+ST+MP, cost function computed by (1), (2), (3), (8), (6), and (9)) on DomainNet reached over 76% and over 77%, respectively, after 20,000 iterations. In this case, the baseline integrated with (DA+ST+MP) reported that its classification accuracy was slightly higher than the case of the baseline integrated with (ST+MP). This proved that the proposed method could achieve high performance without an adaptation module. The detailed results were provided in Table 9, which presented that the classification accuracy in the target domain increased steadily with the proposed MP.

TABLE 9. Ablation study to analyze the impact of each component on the inference accuracy (%) of the target domain in the case of R←C on the *DomainNet* dataset.

# Iterations	Components			
	BL+DA	BL+DA+SL	BL+SL+MP	BL+DA+SL+MP
2,000	64.21	67.21	72.64	73.01
4,000	64.77	67.77	73.93	75.10
6,000	64.87	67.87	74.83	75.91
8,000	65.74	68.74	75.27	75.93
10,000	65.60	68.60	75.70	76.40
12,000	65.63	68.65	76.00	76.60
14,000	66.00	68.98	76.02	76.74
16,000	66.10	69.12	76.03	76.76
18,000	66.50	69.49	76.13	76.80
20,000	66.00	69.02	76.26	77.06



Confusion matrix of S+T method

	Aeroplane	Bicycle	Bus	Car	Horse	Knife	Motorcycle	Person	Plant	Skateboard	Train	Truck
Aeroplane	0.860	0.002	0.008	0.007	0.000	0.014	0.021	0.005	0.002	0.012	0.037	0.031
Bicycle	0.005	0.556	0.003	0.028	0.001	0.043	0.251	0.021	0.031	0.017	0.018	0.026
Bus	0.006	0.004	0.624	0.087	0.001	0.013	0.020	0.013	0.003	0.004	0.113	0.113
Car	0.007	0.009	0.022	0.550	0.004	0.060	0.075	0.031	0.010	0.018	0.027	0.186
Horse	0.016	0.013	0.001	0.007	0.606	0.030	0.140	0.068	0.034	0.012	0.051	0.024
Knife	0.051	0.004	0.015	0.026	0.005	0.555	0.083	0.004	0.044	0.179	0.025	0.010
Motorcycle	0.004	0.029	0.002	0.071	0.001	0.019	0.791	0.013	0.021	0.014	0.014	0.021
Person	0.021	0.024	0.030	0.036	0.006	0.100	0.086	0.522	0.044	0.095	0.012	0.025
Plant	0.009	0.009	0.015	0.006	0.002	0.086	0.014	0.021	0.681	0.031	0.026	0.100
Skateboard	0.022	0.019	0.011	0.032	0.002	0.091	0.091	0.053	0.016	0.599	0.017	0.047
Train	0.009	0.003	0.060	0.021	0.001	0.014	0.009	0.012	0.002	0.007	0.822	0.040
Truck	0.012	0.005	0.090	0.301	0.004	0.021	0.039	0.024	0.006	0.009	0.111	0.380

(a)

Confusion matrix of ENT method

	Aeroplane	Bicycle	Bus	Car	Horse	Knife	Motorcycle	Person	Plant	Skateboard	Train	Truck
Aeroplane	0.902	0.000	0.002	0.018	0.000	0.027	0.006	0.013	0.002	0.002	0.028	0.001
Bicycle	0.002	0.544	0.001	0.016	0.005	0.071	0.285	0.041	0.022	0.004	0.010	0.000
Bus	0.001	0.001	0.725	0.119	0.002	0.015	0.004	0.042	0.003	0.001	0.083	0.003
Car	0.001	0.001	0.010	0.761	0.005	0.113	0.021	0.054	0.009	0.002	0.019	0.004
Horse	0.001	0.001	0.001	0.002	0.863	0.046	0.025	0.042	0.011	0.001	0.007	0.000
Knife	0.003	0.000	0.000	0.005	0.000	0.955	0.004	0.003	0.012	0.011	0.006	0.000
Motorcycle	0.001	0.006	0.000	0.040	0.002	0.033	0.891	0.015	0.004	0.002	0.006	0.000
Person	0.003	0.005	0.001	0.010	0.007	0.170	0.026	0.764	0.009	0.004	0.002	0.000
Plant	0.001	0.001	0.000	0.006	0.001	0.090	0.004	0.040	0.848	0.000	0.008	0.001
Skateboard	0.017	0.002	0.003	0.040	0.003	0.248	0.058	0.124	0.012	0.479	0.013	0.000
Train	0.004	0.001	0.043	0.019	0.001	0.016	0.002	0.030	0.003	0.002	0.877	0.001
Truck	0.005	0.002	0.160	0.502	0.009	0.031	0.014	0.069	0.009	0.002	0.126	0.071

(b)

Confusion matrix of MME method

	Aeroplane	Bicycle	Bus	Car	Horse	Knife	Motorcycle	Person	Plant	Skateboard	Train	Truck
Aeroplane	0.873	0.000	0.003	0.022	0.002	0.028	0.015	0.007	0.001	0.003	0.029	0.017
Bicycle	0.002	0.651	0.002	0.007	0.005	0.050	0.228	0.017	0.016	0.004	0.010	0.009
Bus	0.000	0.001	0.740	0.062	0.002	0.013	0.009	0.022	0.002	0.001	0.067	0.081
Car	0.001	0.003	0.013	0.721	0.007	0.101	0.039	0.033	0.007	0.002	0.024	0.049
Horse	0.003	0.002	0.001	0.000	0.832	0.043	0.070	0.031	0.009	0.001	0.005	0.004
Knife	0.005	0.003	0.002	0.008	0.002	0.898	0.022	0.011	0.015	0.018	0.012	0.003
Motorcycle	0.001	0.009	0.000	0.015	0.002	0.025	0.930	0.004	0.003	0.002	0.006	0.002
Person	0.003	0.005	0.001	0.005	0.011	0.121	0.050	0.782	0.010	0.004	0.004	0.006
Plant	0.002	0.003	0.001	0.001	0.001	0.038	0.015	0.014	0.908	0.001	0.007	0.008
Skateboard	0.008	0.005	0.005	0.022	0.006	0.092	0.088	0.036	0.013	0.705	0.012	0.007
Train	0.001	0.002	0.052	0.020	0.000	0.013	0.004	0.017	0.003	0.001	0.870	0.018
Truck	0.003	0.001	0.147	0.331	0.008	0.025	0.026	0.034	0.007	0.002	0.075	0.340

(c)

Confusion matrix of APE method

	Aeroplane	Bicycle	Bus	Car	Horse	Knife	Motorcycle	Person	Plant	Skateboard	Train	Truck
Aeroplane	0.949	0.001	0.006	0.009	0.000	0.010	0.003	0.006	0.001	0.004	0.012	0.001
Bicycle	0.005	0.688	0.007	0.021	0.006	0.039	0.172	0.022	0.022	0.007	0.010	0.001
Bus	0.003	0.001	0.874	0.051	0.002	0.006	0.003	0.017	0.002	0.000	0.038	0.002
Car	0.003	0.002	0.032	0.811	0.007	0.063	0.010	0.034	0.007	0.004	0.023	0.004
Horse	0.002	0.001	0.002	0.001	0.917	0.027	0.012	0.024	0.008	0.002	0.003	0.000
Knife	0.014	0.002	0.004	0.014	0.001	0.864	0.006	0.009	0.027	0.049	0.011	0.000
Motorcycle	0.002	0.013	0.002	0.051	0.004	0.021	0.886	0.008	0.005	0.003	0.006	0.000
Person	0.003	0.005	0.006	0.013	0.013	0.108	0.030	0.801	0.011	0.008	0.002	0.001
Plant	0.001	0.001	0.002	0.008	0.001	0.037	0.004	0.022	0.917	0.001	0.005	0.001
Skateboard	0.016	0.002	0.013	0.021	0.002	0.051	0.025	0.035	0.013	0.814	0.007	0.001
Train	0.005	0.001	0.116	0.021	0.001	0.009	0.002	0.019	0.002	0.000	0.821	0.002
Truck	0.014	0.001	0.384	0.404	0.010	0.015	0.009	0.033	0.008	0.003	0.055	0.065

(d)

Confusion matrix of Our method

	Aeroplane	Bicycle	Bus	Car	Horse	Knife	Motorcycle	Person	Plant	Skateboard	Train	Truck
Aeroplane	0.981	0.000	0.001	0.001	0.001	0.001	0.001	0.002	0.001	0.002	0.007	0.004
Bicycle	0.003	0.869	0.002	0.003	0.007	0.001	0.073	0.014	0.011	0.007	0.005	0.003
Bus	0.003	0.001	0.871	0.012	0.004	0.000	0.006	0.007	0.001	0.001	0.041	0.053
Car	0.007	0.005	0.015	0.704	0.010	0.003	0.030	0.027	0.004	0.010	0.029	0.156
Horse	0.002	0.001	0.000	0.000	0.974	0.000	0.003	0.010	0.005	0.001	0.002	0.002
Knife	0.008	0.001	0.000	0.004	0.011	0.882	0.005	0.015	0.022	0.042	0.006	0.002
Motorcycle	0.003	0.015	0.000	0.007	0.004	0.001	0.946	0.007	0.003	0.004	0.005	0.005
Person	0.010	0.012	0.004	0.012	0.025	0.017	0.030	0.855	0.011	0.014	0.005	0.005
Plant	0.003	0.002	0.001	0.002	0.002	0.002	0.002	0.014	0.960	0.004	0.004	0.004
Skateboard	0.013	0.004	0.001	0.003	0.004	0.007	0.007	0.030	0.004	0.921	0.004	0.002
Train	0.003	0.002	0.037	0.001	0.001	0.002	0.001	0.007	0.001	0.003	0.927	0.015
Truck	0.008	0.002	0.108	0.149	0.010	0.001	0.011	0.014	0.004	0.006	0.052	0.637

(e)

FIGURE 9. Visualization of the confusion matrix for different methods. These experiments were implemented on *Visda2017* dataset, based on ResNet-34 backbone. The results were obtained from task *Synthetic* to *Real*. Figures (a), (b), (c), (d) and (e) displayed the confusion matrix visualization of S+T [24], ENT [25], MME [26], APE [27], and Our method, respectively.

3) *The confusion matrix visualization analysis*: Figures 9 (a)-(e) displayed the confusion matrix of the different SSDA methods. As shown in Fig. 9, the existing SSDA methods caused the intra-class problem seriously. Specifically, in the ENT [25] and APE [27] methods, the inference accuracy of the Truck class was very low. The model of these methods was confused for the representations among Truck, Bus, and Car classes because they contained many similar representations.

On the contrary, the accuracies of inference results on the Truck class of S+T [24] and MME [26] methods were improved. Because S+T was proposed to reduce the intra-class variation problem, while MME utilized the advantages of the minimax strategy on the entropy of unlabeled target samples, however, their target classification accuracy was limited. This is because S+T ignored the unlabeled target information during training, and MME had the bias learning problem. The proposed method achieved the highest classification performance. It demonstrated that the proposed mapping function and dual feature extractor worked effectively to mitigate the bias learning and accumulated error in the single network.

The feature visualization in Fig. 5 and the confusion matrix visualization in Fig. 9 demonstrated that the proposed method achieved the class feature discriminability on the target domain. By using dual feature extractors, it could mitigate accumulated errors in the single network due to bias selection. In addition, the proposed mapping function was successful in reproducing the well-organized source distribution on the target domain. Therefore, it boosted the target classification results.

V. CONCLUSION

In this work, we developed a new structure with dual feature extractors to capture discriminative features on source and target domains, respectively. Specifically, a feature was trained with high-density samples on the source domain to establish a well-organized distribution. Then, it was connected to the target domain, which is trained using a few target samples, by the class-wise mapping function to reconstruct the well-organized source distribution in the embedding space on the target domain. Experiment results on the cross-domain dataset verified that the embedding space of source and target domain generated by our proposed method was well aligned comparing to several previous domain adaptation methods. Furthermore, the inference accuracy in the target domain was improved considerably compared with the benchmark methods.

REFERENCES

[1] S. Azarbarzin and F. Afsari, ''Domain adaptation by manifold transfer for image classification,'' in *Proc. 4th Iranian Conf. Signal Process. Intell. Syst. (ICSPIS)*, Dec. 2018, pp. 71–75, doi: 10.1109/ICSPIS.2018.8700537.

[2] P. Ge, C.-X. Ren, D.-Q. Dai, and H. Yan, ''Domain adaptation and image classification via deep conditional adaptation network,'' 2020, *arXiv:2006.07776*. [Online]. Available: http://arxiv.org/abs/2006.07776

[3] M. Karimpour, S. N. Saray, J. Tahmoresnezhad, and M. P. Aghababa, ''Multi-source domain adaptation for image classification,'' *Mach. Vis. Appl.*, vol. 31, no. 6, p. 44, Sep. 2020.

[4] G. He, X. Liu, F. Fan, and J. You, ''Classification-aware semi-supervised domain adaptation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 964–965.

[5] R. Gopalan, R. Li, and R. Chellappa, ''Domain adaptation for object recognition: An unsupervised approach,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 999–1006.

[6] F. Yu, D. Wang, Y. Chen, N. Karianakis, T. Shen, P. Yu, D. Lymberopoulos, and X. Chen, ''Unsupervised domain adaptation for object detection via cross-domain semi-supervised learning,'' in *Proc. CVPR Workshop*, 2020, pp. 1–15.

[7] S. Yang, L. Wu, A. Wiliem, and B. C. Lovell, ''Unsupervised domain adaptive object detection using forward-backward cyclic adaptation,'' in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–17.

[8] W. Li, M. Wang, H. Wang, and Y. Zhang, ''Object detection based on semi-supervised domain adaptation for imbalanced domain resources,'' *Mach. Vis. Appl.*, vol. 31, Mar. 2020, Art. no. 18, doi: 10.1007/s00138-020-01068-3.

[9] M. Biasetton, U. Michieli, G. Agresti, and P. Zanuttigh, ''Unsupervised domain adaptation for semantic segmentation of urban scenes,'' in *Proc. CVPRW*, Jun. 2019, pp. 1–10.

[10] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, ''ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation,'' in *Proc. CVPR*, Jun. 2019, pp. 2517–2526.

[11] Y. Li, L. Yuan, and N. Vasconcelos, ''Bidirectional learning for domain adaptation of semantic segmentation,'' in *Proc. CVPR*, Jun. 2019, pp. 6936–6945.

[12] W. Deng, L. Zheng, G. Kang, Y. Yang, Q. Ye, and J. Jiao, ''Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 994–1003.

[13] S. Bak, P. Carr, and J.-F. Lalonde, ''Domain adaptation through synthesis for unsupervised person re-identification,'' in *Proc. ECCV*, 2018, pp. 189–205.

[14] L. Qi, L. Wang, J. Huo, L. Zhou, Y. Shi, and Y. Gao, ''A novel unsupervised camera-aware domain adaptation framework for person re-identification,'' in *Proc. ICCV*, Oct. 2019, pp. 8080–8089.

[15] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, ''Domain-adversarial training of neural networks,'' *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.

[16] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, ''Adversarial discriminative domain adaptation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, p. 4.

[17] A. Chadha and Y. Andreopoulos, ''Improved techniques for adversarial discriminative domain adaptation,'' *IEEE Trans. Image Process.*, vol. 29, pp. 2622–2637, 2020.

[18] I. H. Laradji and R. Babanezhad, ''M-ADDA: Unsupervised domain adaptation with deep metric learning,'' in *Proc. ICML Workshop Domain Adaptation Vis. Understand. (DAVU)*, 2018, pp. 17–31.

[19] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, ''Maximum classifier discrepancy for unsupervised domain adaptation,'' in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.

[20] M. Long, Z. Cao, J. Wang, and M. I. Jordan, ''Conditional adversarial domain adaptation,'' in *Proc. NIPS*, 2018, pp. 1640–1650.

[21] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, ''A closer look at few-shot classification,'' in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[22] M. Wang and W. Deng, ''Cycle label-consistent networks for unsupervised domain adaptation,'' *Neurocomputing*, vol. 422, pp. 186–199, Jan. 2021.

[23] T. Fu and Y. Li, ''Unsupervised domain adaptation based on pseudo-label confidence,'' *IEEE Access*, vol. 9, pp. 87049–87057, 2021.

[24] Y. Si, J. Pu, S. Zang, and L. Sun, ''Extreme learning machine based on maximum weighted mean discrepancy for unsupervised domain adaptation,'' *IEEE Access*, vol. 9, pp. 2283–2293, 2021.

[25] Y. Grandvalet and Y. Bengio, ''Semi-supervised learning by entropy minimization,'' in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 529–536.

[26] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, ''Semi-supervised domain adaptation via minimax entropy,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8050–8058.

[27] T. Kim and C. Kim, ''Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 591–607.

[28] P. Jiang, A. Wu, Y. Han, Y. Shao, M. Qi, and B. Li, ''Bidirectional adversarial training for semi-supervised domain adaptation,'' in *Proc. IJCAI*, Jul. 2020, pp. 934–940.

[29] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, ''ReMixMatch: Semi-supervised learning with distribution matching and augmentation anchoring,'' in *Proc. 8th Int. Conf. Learn. Represent.*, 2020, pp. 1–13.

[30] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, ''MixMatch: A holistic approach to semi-supervised learning,'' in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 1–14.

[31] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, ''FixMatch: Simplifying semisupervised learning with consistency and confidence,'' in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 1–21.

[32] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, ''Randaugment: Practical automated data augmentation with a reduced search space,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 702–703.

[33] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, ''Adapting visual category models to new domains,'' in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

[34] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, ''Deep hashing network for unsupervised domain adaptation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5018–5027.

[35] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, ''Moment matching for multi-source domain adaptation,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1406–1415.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[37] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, ''Automatic differentiation in PyTorch,'' in *Proc. NIPS-W*, 2017, pp. 1–4.

[39] L. van der Maaten and G. Hinton, ''Visualizing data using t-SNE,'' *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[40] I. Goodfellow, ''NIPS 2016 tutorial: Generative adversarial networks,'' 2017, *arXiv:1701.00160*. [Online]. Available: http://arxiv.org/abs/1701.00160

[41] R. Shu, H. H. Bui, H. Narui, and S. Ermon, ''A DIRT-T approach to unsupervised domain adaptation,'' in *Proc. Int. Conf. Learn. Represent.* Vancouver, BC, Canada: OpenReview, 2018, pp. 1–19.

[42] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, ''VisDA: The visual domain adaptation challenge,'' 2017, *arXiv:1710.06924*. [Online]. Available: http://arxiv.org/abs/1710.06924

BA HUNG NGO (Graduate Student Member, IEEE) received the B.S. degree in control engineering and automation from Hanoi University of Mining and Geology, Vietnam, in 2014, and the M.S. degree in control engineering and automation from Hanoi University of Science and Technology, in 2016. He is currently pursuing the Ph.D. degree with Dongguk University, Republic of Korea. His current research interests include computer vision and deep learning, especially deep transfer learning, domain adaptation, and deep learning in medical imaging.

JAE HYEON PARK (Student Member, IEEE) received the B.S. degree in electronic engineering from Daegu University, Republic of Korea, in 2019. He is currently pursuing the M.S. degree with Dongguk University, Seoul, Republic of Korea. His current research interests include image analysis and enhancement, tone mapping processing, deep learning classification, and panel defects evaluation.

SO JEONG PARK received the B.S. degree in multimedia engineering from Dongguk University, Seoul, Republic of Korea, in 2021, where she is currently pursuing the M.S. degree. Her current research interests include semantic segmentation and domain adaptation.

SUNG IN CHO (Member, IEEE) received the B.S. degree in electronic engineering from Sogang University, Republic of Korea, in 2010, and the Ph.D. degree in electrical and computer engineering from Pohang University of Science and Technology, in 2015. From 2015 to 2017, he was a Senior Researcher with LG Display. From 2017 to 2019, he was an Assistant Professor of electronic engineering with Daegu University. He is currently an Assistant Professor of multimedia engineering with Dongguk University, Seoul. His current research interests include image analysis and enhancement, video processing, multimedia signal processing, and circuit design for LCD and OLED systems.