IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 October 2023, accepted 25 November 2023, date of publication 28 November 2023,
date of current version 5 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337438

RESEARCH ARTICLE

Multiple Tasks-Based Multi-Source Domain Adaptation Using Divide-and-Conquer Strategy

BA HUNG NGO, (Member, IEEE), YEON JEONG CHAE, (Member, IEEE), SO JEONG PARK, JU HYUN KIM, 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 work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant RS2023-00208763, in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development Grant funded by the Korea Government (MSIT) under Grant IITP-2023-RS-2023-00254592, and in part by the Development of Neural Network Architecture and Circuits for Few-Shot Learning under Grant NRF-2022M3F3A2A01085463.

ABSTRACT In single-source unsupervised domain adaptation (SUDA), it is often assumed that a single-source domain can cover all target domain features. However, the limitation of labeled samples means that a model trained on a labeled source domain cannot always cover all target representations in practice. Therefore, multi-source unsupervised domain adaptation (MSUDA) has recently become an attractive topic because it can provide richer information than SUDA. In the MSUDA setting, multiple labeled source datasets and an unlabeled target dataset are available. The differently labeled source domains follow distinct distributions to provide different contributions to the target domain. Therefore, when combining multiple source domains into one source domain, the model tends to focus on whichever source domain makes a dominant contribution to the target domain, which induces bias in learning in the MSUDA setting. To solve this problem, this paper proposes a divide-and-conquer-based MSUDA framework that divides the MSUDA problem into multiple tasks (SUDAs) that it then conquers using multiple task-specific models. Each task is a pair that consists of a single source domain and a target domain, and the tasks provide different views on the target domain because each task has a different source domain. Then, they cooperate to supplement their knowledge via collaborative learning. This cooperation between multiple views can suppress noisy information and preserve critical information, thus mitigating the negative transfer problem during DA and significantly boosting the classification accuracy on the target domain as a result. The proposed method achieved state-of-the-art performance on several real-world visual domain adaptation datasets.

INDEX TERMS Multiple source domains, image classification, domain adaptation, transfer learning, multi-task learning, collaborative learning.

I. INTRODUCTION

Traditional deep learning-based approaches generally assume that the training and testing sets come from the same domain and that the model is trained on a labeled dataset to infer results from an unlabeled dataset. However, in real-world applications, the labeling of large-scale training data consumes a lot of time and requires expertise. Domain adaptation (DA) is a widely used approach for tackling the limitation of the low availability of labeled data. The learning mechanism in this approach is inspired by the fact that humans often solve new tasks using experience gained from previous similar tasks in a process called "*transfer learning*." Thus, the purpose of DA is to train a model using labeled data from the source domain(s) and then transfer the knowledge from the prior trained model to either an unlabeled or sparsely labeled testing set of a target domain that has the same feature space but different distribution by learning domain-invariant

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

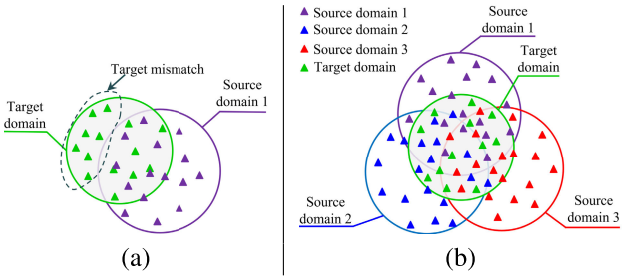

FIGURE 1. Illustration of SUDA and MSUDA tasks. (a) SUDA struggles to cover the target domain. (b) The target domain can be covered by the abundant source samples in MSUDA.

representations. The many advantages of DA mean that it has been widely applied in various applications such as object detection [1], [2], [3], semantic segmentation [4], [5], [6], and image classification [7], [8], [9], [10].

Based on the number of source domains available for training, the DA can be categorized into two subgroups: single-source unsupervised domain adaptation (SUDA) and multi-source unsupervised domain adaptation (MSUDA). SUDA has been widely explored in previous studies [17], [18], [19], [20], [21], [22], [23]. In these methods, they assume that the source samples can cover the entire embedding space of the target domain. However, [40] shows that it is difficult for a single source domain to cover the entire target domain. Therefore, these methods have room for performance improvement, because target samples with representations that are mismatched in a common space can be misclassified, as illustrated in Fig. 1 (a). Thus, MSUDA has recently attracted attention because it can exploit rich information from multiple source domains and transfer it to the target domain. An early MSUDA method [38] combined all labeled source domains into a single source domain. Then, the DA process minimized the domain discrepancy between the source and target domains using SUDA [17]. However, the improvement might not be significant because this method only focuses on learning domain-invariant representations for all domains and does not consider discrimination between classes. In MSUDA, the differently labeled source domains have distinct distributions and deliver different contributions to the target domain. Each pair of source and target domains has a different class decision boundary. Therefore, when combining multiple source domains into one source domain, the model tends to focus on the source domain, which has a dominant contribution to the target domain, leading to bias in learning for the MSUDA setting. This is explained in detail later in Section IV. Furthermore, recent studies [26] and [69] proved that if we naively combine all training data, the semantic information in each domain can be damaged. Thus, the class-discriminative ability of the classifier is reduced due to an increase in the intra-class variance.

To preserve the unique features in each source domain, and alleviate dominant domain and intra-class variance problems, we introduce a novel framework called Divide and Conquer Using Multiple Tasks (DCMT). Specifically, the dividing stage is proposed to preserve the specific characteristics in each source domain and reduce the variance of the training set from the multiple source domains, while the conquering stage is proposed to leverage information learned from each group, including multiple tasks, to benefit the training of the specific task as shown in Fig. 2. Our method assumes that MSUDA includes N source domains and a target domain and is divided into N SUDA tasks. Each task contains a labeled source domain and an unlabeled target domain. The proposed framework includes a shared feature extractor, N classifiers, and N discriminators. In the dividing stage, the feature extractor tries to generate the representations of all domains. Simultaneously, the classifiers trained on different labeled source domains hold the unique features of each source domain. In this way, we can mitigate the intra-class variance among multiple source domains. Besides, multiple classifiers have unique characteristics in each source domain, providing different views on the target domain. Each view contains partial target information, as shown in Fig. 1(b). In the conquering stage, to unify the target information from different views, we propose collaborative learning (Co-learning) that produces multi-view consistency for alleviating the dominant domain problem and mapping the common characteristics across different domains in a joint embedding space. The contributions of this paper can be summarized as follows:

1) To optimally exploit information from multiple source domains, we propose a divide-and-conquer-based framework that divides the MSUDA task into multiple SUDA tasks. Each SUDA task is handled by a different model built from N classifiers, N discriminators, and the shared feature extractor. In this way, we can alleviate the intra-class variance problem. Besides, these models can obtain unique information from various source domains to provide different views on the target domain.
2) We propose Co-learning that allows multiple classifiers to exchange their high-confidence predictions of target samples in every iteration. Thus, they can provide complementary information to each other to enhance learning effectiveness and alleviate bias in learning (dominant domain) in MSUDA. This approach can address the negative transfer problem by suppressing noise and preserving important information because the multiple views of the model are trained to encourage proving consistent prediction.
3) We showed the effectiveness of our method by comprehensive experiments with various benchmark datasets, including *Office-31*, *Office-Caltech10*, *ImageCLEF-DA*, *Office-Home*, and *DomainNet*.

II. RELATED WORK

In this section, we review previous studies into single-source domain adaptation and multiple-source domain adaptation.

A. SINGLE SOURCE DOMAIN ADAPTATION

In recent decades, research groups have been forced to propose methods for transferring knowledge from a source domain to a target domain by resolving the domain shift [21] on SUDA. These methods can be categorized into three groups: adversarial-based [17], [18], [19], [20], discrepancy-based [21], [22], [23] and autoencoder-based methods [24], [25], [26].

Domain-Adversarial Training of Neural Networks (DANN) [17] is a popular adversarial-based method used in unsupervised domain adaptation; it consists of three components: a feature extractor, a classifier, and a discriminator. The feature extractor is shared between the source and target domains and is trained to maximize the classification accuracy on labeled source samples and to confuse the discriminator, making it impossible to distinguish between source and target domains. Instead of using the common feature extractor, Adversarial Discriminative Domain Adaptation (ADDA) [18] uses two feature extractors, one for the source domain and one for the target domain. The training process of ADDA has two stages. In the first stage, the parameters of the source feature extractor are optimized by minimizing the classification error on the labeled source domain. In the second stage, the target feature extractor parameters are initialized by the pre-trained source feature extractor. Then, they are fine-tuned on the unlabeled target data to confuse the discriminator by reducing the difference between the source and target distributions.

Discrepancy-based methods [21], [22], [23] align the target and source distributions by minimizing their domain dissimilarity. Maximum Mean Discrepancy (MMD) [32] is the most popular method for measuring the similarity between two different distributions. Domain Adaptation via Transfer Component Analysis (TCA) [21] utilizes MMD, aiming to reduce the mean deviation of the data of the source and target domains by minimizing the marginal distribution difference. Learning Transferable Features with Deep Adaptation Networks (DAN) [22] uses MMD to enhance the transferable features by explicitly minimizing domain discrepancies in the adaptation layers of the deep neural network.

Transfer Learning with Deep Autoencoders (TLDA) [24] is a popular autoencoder-based method that resolves the domain shift by using Kullback–Leibler (KL) divergence to make the two domains' distributions close in the embedding space. A Bi-shifting Auto-Encoder (BAE) network [25] has been proposed to minimize the shift in representations between the source and target domains by bidirectional transformation learning. In Dual-Representation-Based Autoencoder for Domain Adaptation (DRAE) [26], the distribution divergence between the source and target domains is minimized by learning the global representations of both domains in the first phase. The local representation is learned in the second phase to maintain class-discriminative information in each class of both domains.

B. MULTI-SOURCE DOMAIN ADAPTATION

Multi-Source-TrAdaBoost (MTrA) [33] is one of the earliest frameworks for MSUDA. It was developed based on the TrAdaBoost [34] algorithm for transferring extracted knowledge from multiple source domains to a target domain, to improve the classification accuracy of the target domain. However, in this method, to reduce the impact of negative transfer, only a source domain closely related to the target domain is considered in each iteration of the adaptation process. Self-supervised Implicit Alignment (SImpAl) [41] extends the concept of Maximum Classifier Discrepancy (MCD) [19] for multi-source domain adaptation using multiple classifiers. This method enforces an agreement between the different classifiers to implicitly align the domains in the latent space without requiring any additional components, such as a discriminator for adversarial learning. Adversarial Multiple Source Domain Adaptation (MDAN) [38], inspired by the concept of DANN [17], provides a novel generalization bound for multi-source domain adaptation. This method has two versions: *hard* and *soft*. In the hard version, DANN is applied to minimize the domain discrepancy for each source-target pair. However, in the soft version, DANN is applied to the combined source domain containing all source domains and the target domain. Multi-Source Adaptation Network (MSAN) [42] mitigates the domain shift between the target domain and multiple source domains by applying multiple GAN architectures to simultaneously transfer information from the source domains to the target domain. An Attention Guided Multiple Source and Target Domain Adaptation (AMDA) [49] focuses on extracting the important semantic information in images from multiple source domains by exploring the attention mechanism, which improves the classification performance by alleviating the negative transfer.

III. PROPOSED METHOD

As shown in Fig. 2, the architecture of the proposed divide-and-conquer-based method consists of three components: a common feature extractor, N classifiers, and N discriminators, where N is the number of source domains. Each model includes the shared feature extractor, a classifier, and a discriminator; it is used to extract the representations of a task that consists of a source domain and a target domain. The source domain for each task is different. Thus, the number of divided tasks is the same as the number of source domains, N.

The shared feature extractor extracts the representations of all samples from the multiple source domains and the target domain. Therefore, it can represent knowledge that is common to all domains. The different classifiers are used to categorize the representations of the different labeled source domains. The discriminator minimizes the domain discrepancy between the source domain and the target domain in each task.

The labeled source domains are used to train the N different models. These models obtain representations of the

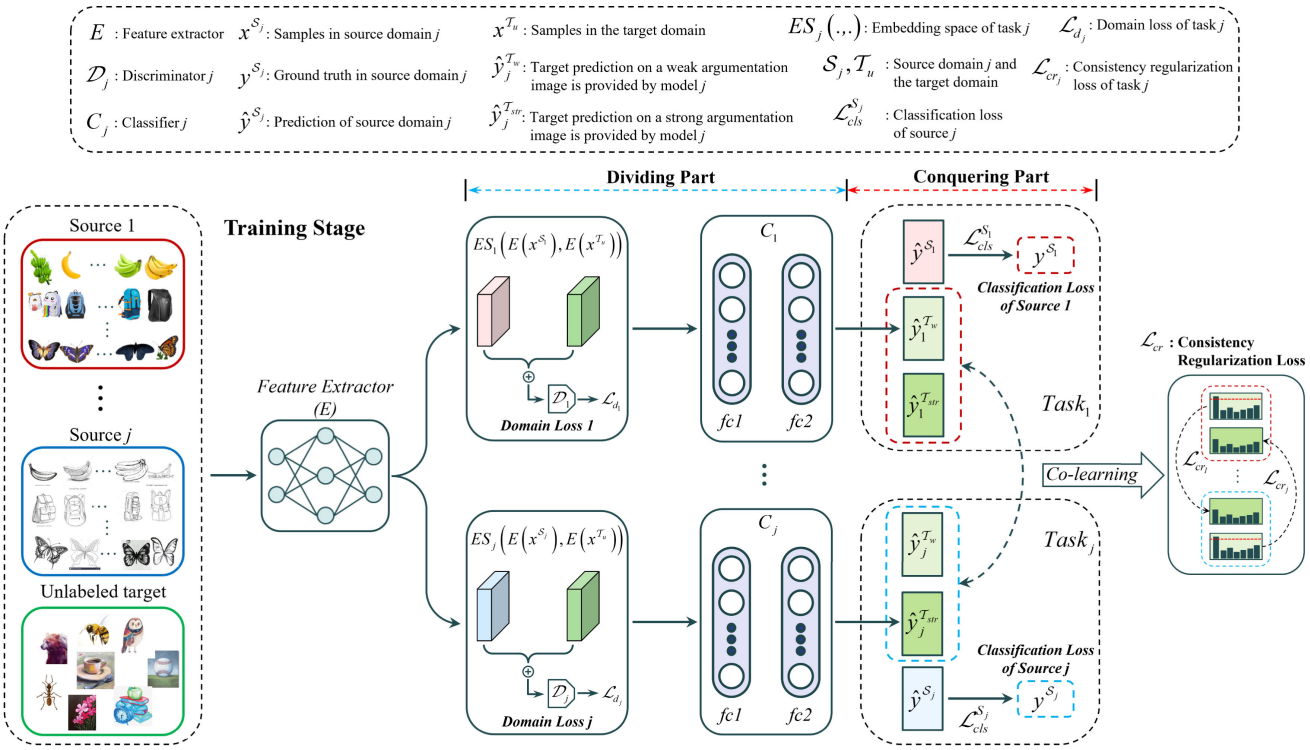

FIGURE 2. The architecture of the proposed method. A shared feature extractor *E* extracts the representations of an unlabeled target domain and *N* labeled source domains. MSUDA is decomposed into various SUDA tasks. A model consists of a shared feature extractor, a classifier, and a discriminator. It works as an expert when trained on different labeled source domains in a *j*-th task using supervised learning. Each model can fully exploit the information of a source domain. Then, we used Co-learning to exchange knowledge among multi-expert models. These models exchange their pseudo labels, which are generated by selecting the high-confidence scores from the weak augmentation images of unlabeled target samples to teach each other via Co-learning.

N source domains and provide their views on the unlabeled target domain. Since a source domain holds only partial information about the target domain, we use the Co-learning algorithm, which exchanges the knowledge of different models, to supplement the different views on the target domain. Our method aims to extract the target information from different source domains and then unify this information to obtain all of the target representations in a process called "*conquering*".

A. PROBLEM DEFINITION AND NOTATION

The problem setting of MSUDA, involves N labeled source datasets $S = \{S_j\}_{j=1}^{N}$, where each $S_j = \left\{(x_i^{S_j}, y_i^{S_j})\right\}_{i=1}^{n_{S_j}}$ contains n_{S_j} labeled samples, $x_i^{S_j}$ is the i-th image in source j and $y_i^{S_j}$ is its category label. There is a single unlabeled target dataset $T = \{x_i^T\}_{i=1}^{n_T}$, where n_T is the total number of images in the unlabeled target domain, and x_i^T is the i-th unlabeled target image. We summarize the important symbols used throughout our paper in Table 1.

In the proposed method, the problem of MSUDA is solved by dividing it into multiple single-source unsupervised domain adaptation (SUDA) tasks. Data samples in each task come from the source and target domains. The features in each task are extracted by a model. Consequently, there are N models $\omega = \{\omega_j\}_{j=1}^{N}$ used to decompose N tasks in a MSUDA problem. These models contain different classifiers and discriminators but share the same feature extractor. As shown in Fig. 2, the data samples of task j aggregated from $S_j \cup T$ are used to train the model ω_j, which consists of the feature extractor E, classifier C_j, and discriminator D_j.

TABLE 1. Important notation.

Notation	Description
S_j	Source domain j
ω_j	The model j
$E(.)$	The common feature extractor
D_j	The discriminator of model j
C_j	The classifier of model j
$E\left(x^{S_j}\right)$	The feature space of the samples in source domain j
$\left(x_i^{S_j}, y_i^{S_j}\right)$	The i-th element sample and its label in source domain j
x_i^T	The i-th element sample in the target domain
n_{S_j}	The number of samples in source domain j
n_T	The number of samples in the target domain
$E\left(x_i^T\right)$	The feature of the i-th unlabeled sample in the target domain

B. MULTI-VIEW LEARNING PROCESS

In MSUDA, multiple source domains carry various features. Each source might only contain a part of the target

information [9], [40]. Therefore, the whole set of features in the target domain can be covered by combining the features from all source domains, as shown in Fig. 1 (b). We verify this assumption by demonstrating the embedding space of the real data in Section IV.

There are N tasks created from a given unlabeled target, and N labeled source domains. The features of task j are $ES_j = \big(E(x^{S_j}), E(x^T)\big)$, where x^{S_j} is the set of labeled images of the source domain S_j and x^T is the set of the unlabeled target images, extracted by model $\omega_j, j \in \{1, 2, \ldots, N\}$. First, models are trained on their own labeled source domain. Then, each model generates predicted values on the target samples, using the trained knowledge from its own source domain. Since the knowledge obtained from the different models is significantly different, they provide different views on the target domain. Co-learning is used to allow the different models to provide the same prediction on the unlabeled target data. The closeness of these predictions is called the prediction consistency in the target domain. Section III explains the details of co-learning. Co-learning can boost the target classification accuracy and alleviate bias in the learning of the different models trained on different labeled source domains. The detailed training process is as follows.

C. TRAINING PROCESS

The feature extractor E and classifier C_j of the model ω_j are trained in a supervised manner to minimize the classification loss on labeled samples from the source domain S_j over K classes. The j-th classifier is trained using the standard cross-entropy loss according to the following equation:

$$\mathcal{L}_{cls}^{S_j} = -\mathbb{E}_{(x_i^{S_j}, y_i^{S_j}) \sim S_j} \sum_{k=1}^{K} \mathbb{1}_{[k=y_i^{S_j}]} \log\big(C_j(E(x_i^{S_j}))\big), \quad (1)$$

where $\mathbb{1}_{[.]}$ is an indication function with a value of either 1 or 0, depending on whether the input [] is true or false.

The cost of training the shared feature extractor E is computed across all labeled source domains, as follows:

$$\mathcal{L}_{E_{cls}} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}_{cls}^{S_j}. \quad (2)$$

The common feature extractor is trained by applying Eq. (2) to N source domains; thus, it obtains features from multiple source domains. The classifier in each model only holds the representations of its source domain.

1) DOMAIN-LEVEL ADAPTATION FOR ALIGNMENT

We then use adversarial training strategies to reduce the domain discrepancy between each pair of source and target domains in each task. Similar to DANN [17], the adversarial loss function is calculated as follows:

$$\min_E \max_{D_j} \mathcal{L}_{d_j} = \mathbb{E}_{x_s \sim S_j}\big[\log(\mathcal{D}_j(E(x_s)))\big] + \mathbb{E}_{x_t \sim T}\big[\log(1 - \mathcal{D}_j(E(x_t)))\big], \quad (3)$$

where $\mathcal{L}_{d_j}$ is the domain loss between the target domain and the source domain S_j. The cost function of the adversarial process for training the shared feature extractor E across all source and target domains is computed as follows:

$$\mathcal{L}_{E_{domain}} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}_{d_j}. \quad (4)$$

The discriminator in each model is trained using Eq. (3).

2) CO-LEARNING ON AN UNLABELED TARGET DOMAIN

In practice, the differently labeled source domains have distinct distributions, so the different source domains have very different contributions to the target domain. The target domain is attracted by the source domain that contributes the most to it, leading to bias in learning in MSUDA. Therefore, we use Co-learning to allow different models to teach each other by mutually exchanging information about the target domain provided by multiple views.

The Co-learning process includes two steps. In the first step, augmentation is applied over an unlabeled target image x_i^T to create two versions: the weakly augmented, $a(x_i^T)$, and strongly augmented versions, $A(x_i^T)$, where $a(.)$ is the weak augmentation function consisting of simple transformations such as flipping and randomly cropping images, $A(.)$ is the strong augmentation functions inspired by the RandAugment [35], it transforms the input image by randomly selecting from various augmentation methods such as equalization, image sharpening, brightness variation, rotation, or color variation and applying them to an input image. Then, the model ω_m corresponding to task m is used to generate the prediction values of the weakly augmented version $a(x_i^T)$, while the model ω_j corresponding to task j provides the predictions of the strongly augmented version $A(x_i^T)$ with the same unlabeled target image as follows:

$$\begin{aligned} p_m^w &= p_m^w\left(x_i^T\right) = softmax\big(C_m(E(a(x_i^T)))\big), \\ p_j^{str} &= p_j^{str}\left(x_i^T\right) = softmax\big(C_j(E(A(x_i^T)))\big), \end{aligned} \quad (5)$$

where p_m^w and p_j^{str} are the prediction vectors of the weak and strong augmentation versions of a target image x_i^T generated by models ω_m and ω_j, respectively. m and j are the indexes of the model. In the second step, Co-learning is implemented as follows: model ω_j provides its prediction on the strongly augmented image p_j^{str}. The other models offer their pseudo labels by selecting the highest confidence score from a weakly augmented image $\max\left(p_m^w\right)$, where $m \in \{1, 2, \ldots, N\}$ and $m \neq j$. Then, consistency regularization is conducted by minimizing the cross-entropy of each selected pseudo label $argmax\left(p_m^w\right)$ and prediction p_j^{str} of the strongly augmented image.

Incorrect pseudo-labels can have a negative effect on the performance of models. Therefore, only the output with high probability, over a given threshold value, is selected as a pseudo label, $(\max\left(p_m^w\right) \geq \tau)$, where τ is the threshold value. The method for selecting the threshold value is detailed in

Section IV. The consistency loss between the j-th model and the rest of the models is calculated as follows:

$$\mathcal{L}_{cr_j} = \sum_{\substack{m=1 \\ m \neq j}}^{N-1} \sum_{i=1}^{n_T} \mathbb{1}\left[\max\left(p_m^w\right) \geq \tau\right] \cdot \hat{y}_i^{T_m} \log\left(p_j^{str}\right), \quad (6)$$

where $\mathbb{1}_{[.]}$ is an indication function, and $\hat{y}_i^{T_m}$ is the pseudo label of the unlabeled target sample x_i^T created by the m-th model. Eq. (6) describes that the proposed method leverages knowledge from (N-1) models to benefit the training of the j-th specific model by using pseudo labels. First, (N-1) models offer their pseudo labels by selecting the highest prediction on the weakly augmented image of the target samples. These pseudo labels then are converted to one-hot encoded labels, $\hat{y}_i^{T_m} = argmax\left(p_m^w\right)$, to compute the cross-entropy loss with the prediction of the j-th model on the strongly augmented version of the same target samples. This process is called collaborative learning (Co-learning), which leverages information learned from each group, including multiple tasks, to benefit the training of the specific task. The Co-learning loss used to train the shared feature extractor E is computed as follows:

$$\mathcal{L}_{E_{Co}} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}_{cr_j}. \quad (7)$$

The classifier in each model is updated using Eq. (6). When $N = 2$, Co-learning is simplified to co-training, in which two models collaborate to make a consistent prediction on the target domain. The proposed Co-learning method allows different models to supplement information mutually by interacting among multiple views. Therefore, knowledge from multiple source domains can be transferred to the target domain more robustly. We show that Co-learning reduces the performance degradation due to bias in learning in Section IV.

In previous work [36], the model could efficiently categorize unlabeled target data by encouraging the features to cluster around a specific class in the source domain. Thus, we implemented an added cost function to train the feature extractor and classifier in each model, using a minimax strategy inspired by [36]. The classifier C_j is trained on the labeled source domain j and consists of weight vectors $W_j = \left[w_j^1, w_j^2, \ldots, w_j^K\right]$, where $j \in \{1, 2, \ldots, N\}$, K indicates the number of classes, and the weight vector w_j^i represents the i-th class prototype. The output of classifier C_j, $\frac{1}{\alpha} W_j^T f$, is fed into a softmax layer to obtain the final probability output $p(x) = \sigma\left(\frac{1}{\alpha} W_j^T f\right)$, where α is the temperature, and f is the input feature. The entropy maximization is performed on unlabeled target data to make each w_j^i similar to the target features f for the generation of a domain-invariant prototype. The feature extractor is trained to discriminate features between the source and target domains, in which the f is assigned to one of the prototypes by minimizing the entropy. The impact of this cost on the model for classifying the target domain is discussed in Section IV. The unlabeled target data is fed to model ω_j. The entropy is calculated as follows:

$$H_j = -\mathbb{E}_{x_i^T \sim T} \sum_{k=1}^{K} p_j(y = k \mid x_i^T) \log\left(p_j(y = k \mid x_i^T)\right), \quad (8)$$

where K represents the number of classes and $p_j(y = k \mid x_i^T)$ represents the probability of x_i^T belonging to class k, which is the prediction output of model ω_j.

The total cost functions for training the shared feature extractor E are computed as follows:

$$\mathcal{L}_E = \mathcal{L}_{E_{cls}} + \mathcal{L}_{E_{domain}} + \mathcal{L}_{E_{Co}} + \lambda \frac{1}{N} \sum_{j=1}^{N} H_j, \quad (9)$$

where λ is a balancing parameter [36]. The cost functions for training the classifiers are calculated as follows:

$$\mathcal{L}_{C_j} = \mathcal{L}_{cls}^{S_j} + \mathcal{L}_{cr_j} - \lambda \sum_{j=1}^{N} H_j. \quad (10)$$

The discriminator in each model is computed using Eq. (3).

D. INFERENCE

The final prediction on the i-th unlabeled target sample is computed by taking the averaged softmax outputs of multiple classifiers as follows:

$$y_{prediction} = argmax\left(\frac{1}{N} \sum_{j=1}^{N} \left(C_j(E(x_i^T))\right)\right). \quad (11)$$

IV. EXPERIMENTS

We evaluated our proposed method on benchmark datasets for MSUDA tasks. Then, we analyzed the contributions of the proposed method via extensive ablation.

A. DATASETS

For the experiments, we used five standard benchmark datasets: *Office-31*, *Office-Home*, *Office-Caltech10*, *ImageCLEF-DA*, and the challenging large-scale benchmark, *DomainNet*.

- *Office-31* [50] is an unbalanced dataset consisting of 4,110 images from three different domains: *Amazon* (**A**), Webcam (**W**), and *DSLR* (**D**). *Amazon* has 2,817 images, *Webcam* holds 795 images, and *DSLR* contains 498 images. They share 31 categories. We implemented the proposed method for three scenarios: **A**, **W**→**D** (where **A** and **W** are the source domains and **D** is the target domain); **A**, **D**→**W**; and **D**, **W**→**A**, as in [49].
- *Office-Home* [51] is an unbalanced dataset containing 15,588 images from four domains: *Real World* (**Rw**), *Clipart* (**Cl**), *Art* (**Ar**), and *Product* (**Pr**) which share 65 categories. **Ar** contains 2,427 images, **Cl** comprises 4,365 images, **Pr** holds 4,439 images, and **Rw** has 4,357 images. The proposed method was evaluated via two cases: two-source domain and three-source domain.

The two-source domain setting includes 12 tasks: **Rw**, **Pr→Ar/Cl** (where *Real World* and *Product* are the source domains and *Art* and *Clipart* are the target domains); **Cl**, **Rw→Ar/Pr**; **Pr**, **Cl→Ar/Rw**; **Rw**, **Ar→Cl/Pr**; **Ar**, **Pr→Cl/Rw**; **Cl**, **Ar→Pr/Rw**. The three-source domain setting includes four tasks: **Ar**, **Cl**, **Pr→Rw**; **Ar**, **Cl**, **Rw→Pr**; **Ar**, **Pr**, **Rw→Cl**; and **Cl**, **Pr**, **Rw→Ar**.

- *Office-Caltech10* [52] is an unbalanced dataset containing mixed images from *Office-31* and *Caltech10*, with 2,533 images sharing 10 categories. The *Office-31* dataset includes three domains: *Amazon* (**A**) holds 958 images, *Webcam* (**W**) contains 295 images, and *DSLR* (**D**) consists of 157 images. *Caltech10* contributes a domain, *Caltech* (**C**), which has 1,123 images. The proposed method was tested via four tasks: **A**, **D**, **W→C**; **C**, **D**, **W→A**; **A**, **C**, **D→W**; and **A**, **C**, **W→D**, as in [48].
- *ImageCLEF-DA* [22] contains four different domains: *Caltech-256* (**C**) [53], *ImageNet ILSVRC 2012* (**I**) [54], *Pascal VOC 2012* (**P**) [55], and *Bing* (**B**). Each domain has 12 categories, and each category contains 50 images. We evaluated the proposed method for the following tasks: **B**, **C→I/P**; **B**, **I→C/P**; **B**, **P→C/I**; **C**, **I→B/P**; **C**, **P→B/I**, and **I**, **P→B/C**, where **B**, **C→I/P** indicates that the knowledge of source domains **B** and **C** is transferred to the target domain **I** or **P**, as in [49].
- *DomainNet* [43] is a challenging large-scale domain adaptation dataset containing 345 categories with six different domains. Where *Real* (**rel**) has 175,327 images, *Clipart* (**clp**) contains 48,837 images, *Painting* (**pnt**) consists of 75,759 images, *Sketch* (**skt**) holds 70,386 images, *Infograph* (**inf**) has 53,201 images, and *Quickdraw* (**qkd**) contains 172,500 images. In experiments, for a fair comparison with previous MSUDA works, we selected four domains, *Real*, *Painting*, *Clipart*, and *Sketch*, with 126 categories in each domain from the *DomainNet-126* dataset. We constructed 12 scenarios to evaluate the proposed method: **rel**, **skt→clp/pnt**; **skt**, **pnt→clp/rel**; **pnt**, **rel→clp/skt**; **clp**, **skt→pnt/rel**; **rel**, **clp→pnt/skt**; and **clp**, **pnt→rel/skt** in the two-source domain setting, as in [49]. In the three-source domain setting, the proposed method was evaluated for four scenarios: **rel**, **pnt**, **skt→clp**; **pnt**, **clp**, **skt→rel**; **rel**, **clp**, **skt→pnt**; and **clp**, **rel**, **pnt→skt**. We also verified the classification performance of the proposed method with the challenge domain adaptation tasks on the five-source domain setting with *DomainNet-345*[1] dataset, where 345 was the number of categories included in each source domain. We reported the experimental results of six domain adaptation tasks: **inf**, **pnt**, **qdr**, **rel**, **skt→clp**; **clp**, **pnt**, **qdr**, **rel**, **skt→inf**; **clp**, **inf**, **qdr**, **rel**, **skt→pnt**; **clp**, **inf**, **pnt**, **rel**, **skt→qdr**; **clp**, **inf**, **pnt**, **qdr**, **skt→rel**; **clp**, **inf**, **pnt qdr**, **rel→skt**.

[1] http://ai.bu.edu/M3SDA/#dataset

TABLE 2. Description of datasets used in the experiments.

Datasets	Domains	Instance	Classes
Office-31	Amazon	2,817	31
	Webcam	795	
	DSLR	498	
Office-Home	Real World	4,357	65
	Clipart	4,365	
	Art	2,427	
	Product	4,439	
Office-Caltech10	Amazon	958	10
	Webcam	295	
	DSLR	157	
	Caltech10	1,123	
ImageCLEF-DA	Caltech-256	600	12
	ImageNet-ILSVRC	600	
	Pascal VOC	600	
	Bing	600	
DomainNet	Real	175,327	345
	Clipart	48,837	
	Painting	75,759	
	Sketch	70,386	
	Infograph	53,201	
	Quickdraw	172,500	

Detailed information about all of the datasets used for the experiments is given in Table 2.

B. EXPERIMENTAL SETTING

The baseline (BL) of the proposed method used for experiments consisted of a shared feature extractor, N classifiers, and N discriminators, where N is the number of source domains. Therefore, in the two-source domain, three-source domain, and five-source domain settings, the numbers of classifiers and discriminators were two, three, and five, respectively. Similar to AMDA [49] and DRT [47], we selected ResNet-50 and ResNet-101 as the backbones and pre-trained on the ImageNet dataset [54] for the shared feature extractor of MSUDA. A classifier consisted of networks with two fully connected layers containing 512 hidden units, and a discriminator containing three fully connected layers with 1,024 hidden units. All parameters in the shared feature extractor, classifiers, and discriminators were updated using backpropagation with stochastic gradient descent (SGD). The momentum was 0.9, and the initial learning rate was $\eta_0 = 0.01$. The weight decay was set to 0.0005. The balancing parameter in Eqs. (9) and (10) was set to 0.1. The batch size (b) used in the experiments was 96. The threshold value, τ in Eq. (6), was set to 0.88, as reported in the ablation study section. All experiments were implemented in Pytorch framework [56] on a GeForce RTX3090 GPU.

TABLE 3. Comparison of different methods on the *Office-Home* dataset based on a ResNet-50 backbone in the two-source domain setting.

Standard	Source	Rw, Pr		Cl, Rw		Pr, Cl		Rw, Ar		Ar, Pr		Cl, Ar		Mean
	Target	Ar	Cl	Ar	Pr	Ar	Rw	Cl	Pr	Cl	Rw	Pr	Rw	
Single best	DAN [23]	63.1	43.6	45.8	74.3	44.0	60.4	51.5	57.0	43.6	67.7	57.0	60.4	56.3
	DANN [18]	63.2	43.7	47.0	76.8	46.1	60.9	51.8	59.3	45.6	68.5	59.3	60.9	57.6
	JAN [24]	63.9	43.4	50.4	76.8	45.8	61.0	52.4	61.2	45.9	70.3	61.2	61.0	58.3
	MCD [20]	60.3	50.0	57.4	69.2	47.4	66.1	50.0	69.2	45.0	70.1	69.9	60.1	59.6
	DMDA [16]	66.8	57.6	66.8	82.6	57.9	**79.2**	57.6	82.6	55.7	79.2	76.1	78.6	70.1
	ATM [11]	73.3	58.9	73.3	83.4	61.1	79.1	58.9	**83.4**	52.4	79.1	72.6	78.0	71.1
	DSAN [15]	**73.8**	60.6	**73.8**	83.1	62.6	78.5	60.6	83.1	55.9	78.5	70.8	75.4	71.4
Multi-Source	MDAN [39]	62.5	48.2	63.5	74.6	52.6	69.7	48.6	74.3	44.0	67.7	67.6	73.1	62.2
	AMDA [50]	69.5	53.2	70.5	**83.5**	**66.5**	78.0	55.2	79.5	50.9	78.9	**76.2**	**78.6**	70.1
	DCMT (ours)	69.6	**64.6**	70.5	81.2	66.1	77.9	**66.1**	79.1	**62.5**	**81.9**	73.4	78.1	**72.6**

TABLE 4. Comparison of different methods on the *ImageCLEF-DA* dataset based on a ResNet-50 backbone in the two-source domain setting.

Standard	Source	I, P		P, C		C, I		B, P		I, B		B, C		Mean
	Target	B	C	B	I	B	P	C	I	C	P	I	P	
Single best	DAN [23]	-	89.8	-	86.3	-	74.5	-	82.2	92.8	-	-	69.2	-
	DANN [18]	61.3	91.5	57.8	87.0	63.2	75.0	89.8	86.0	96.2	69.5	84.3	74.3	77.9
	JAN [24]	-	91.7	-	89.5	-	76.8	-	88.0	94.7	-	-	74.2	-
	MADA [28]	-	92.2	-	88.8	-	75.0	-	87.9	96.0	-	-	75.2	-
	DCAN [65]	63.2	95.5	62.5	90.8	63.2	**79.8**	92.4	85.5	**95.5**	**79.8**	90.8	65.2	80.4
	ADAN [63]	-	95.9	-	91.2	-	78.9	-	-	-	-	-	-	-
	CRCB [62]	-	94.5	-	90.8	-	78.8	-	-	-	-	-	-	-
	ETDS [64]	-	**96.1**	-	91.9	-	77.5	-	-	-	-	-	-	-
Multi-Source	AMDA [50]	63.3	94.2	63.0	91.5	66.3	77.0	94.3	90.3	94.5	75.8	91.3	**76.2**	81.5
	DCMT (ours)	**64.5**	95.2	**66.0**	**92.2**	**66.5**	**77.7**	**95.0**	**93.3**	94.8	76.2	**91.7**	75.2	**82.4**

For evaluating our method, we used the *Office-Home*, *ImageCLEF-DA*, and *DomainNet-126* datasets as in [49], the *Office-31*, *Office-Caltech10* datasets as in [48], and the *DomainNet-345* datasets as in [45], [46], and [47]. The *DomainNet-345* dataset had two versions, including the original and cleaned versions. The cleaned version was generally recommended; however, during implementation, we found a problem with the cleaned version. The *'t-shirt'* class with the index of 327 in the *Painting* training set (*painting_train.txt* file) was excluded, but this class was inserted in the test set (*painting_test.txt* file), which could negatively affect to the final classification accuracy. This problem was also mentioned in the previous work [46]. Therefore, all experiments on the *DomainNet-345* datasets were conducted by using the original version.

C. COMPARISON WITH SOTA METHODS

For a fair comparison with previous publications, we used the ResNet-50 backbone to extract the results on *ImageCLEF-DA*, *Office-31*, *Office-Caltech10*, *Office-Home*, and *DomainNet-126* datasets, while we used the ResNet-101 backbone to extract the results of the *DomainNet-345* dataset.

D. ANALYSIS OF RESULTS

In this section, we reported the domain adaptation results of the proposed method in three settings: two-source domain, three-source domain, and five-source domain. In the two-source domain setting, two labeled source domains were used for domain adaptation on an unlabeled target domain. In the three-source domain, three labeled source domains transferred their knowledge to an unlabeled target domain, and in the five-source domain setting, the knowledge of five labeled source domains was extracted to adapt to an unlabeled target domain.

Tables 3, 4, 5, and 6 reported comparisons of the results of the proposed method and SOTA domain adaptation methods in the two-source domain setting on *Office-Home*, *ImageCLEF-DA*, *DomainNet-126*, and *Office-31* datasets, respectively. The results in Tables 3, 4, and 5 were divided into two parts. One part reported the results of single-source-single-target methods of which the best classification accuracy on the target domain was selected, called the single best. For example, as shown in Table 3, **Rw** and **Pr** were selected to work as source domains, and **Ar** was a target domain. The highest result between **Rw**→**Ar** and **Pr**→**Ar** was chosen to report as *single best*. The other part contains the results of multi-source-single-target methods, called *multi-source*.

The proposed method achieved outstanding performance compared to other benchmark methods in most of the domain adaptation tasks. Compared to AMDA, which is one of the latest methods, the mean accuracy of our method was 2.5% higher on the *Office-Home* dataset and 0.9% higher on the *ImageCLEF-DA* dataset, as recorded in Tables 3 and 4,

TABLE 5. Comparison of different methods on the *DomainNet-126* dataset based on a ResNet-50 backbone in the two-source domain setting.

Standard	Source	rel, skt		skt, pnt		pnt, rel		clp, skt		rel, clp		clp, pnt		Mean
	Target	clp	pnt	clp	rel	clp	skt	pnt	rel	pnt	skt	rel	skt	
Single best	DAN [23]	37.4	28.2	39.1	42.1	29.9	26.4	23.4	36.2	33.3	29.7	37.9	26.1	32.5
	DANN [18]	36.5	26.3	37.9	41.5	29.1	24.5	23.2	35.3	33.9	28.6	37.6	24.7	31.6
	JAN [24]	33.5	27.7	35.3	43.1	27.5	21.9	24.5	36.8	32.5	25.7	38.1	23.9	30.9
	MCD [20]	42.6	27.6	41.2	50.5	34.4	29.3	26.1	34.8	42.6	33.8	45.0	28.4	36.4
	MCC [71]	64.6	63.4	64.6	73.5	64.7	56.1	56.4	65.3	63.4	54.3	73.5	56.1	63.0
	PCT [72]	73.1	71.4	73.4	80.1	73.4	65.1	65.5	73.3	71.4	64.5	80.1	65.1	71.4
	TransPar-MCC [73]	67.2	70.0	66.2	78.1	67.2	60.4	60.8	66.4	70.0	59.4	78.1	60.4	67.0
Multi-Source	MDAN [39]	53.5	55.5	53.3	64.7	47.2	42.7	53.0	64.5	56.2	47.1	68.7	53.5	54.5
	AMDA [50]	66.2	65.4	63.4	72.2	58.4	54.8	60.3	69.8	61.6	56.2	73.3	59.4	63.4
	DCMT (ours)	**81.9**	**76.6**	**78.9**	**82.3**	**79.5**	**71.5**	**73.8**	**80.0**	**75.0**	**73.5**	**82.8**	**73.8**	**77.5**

TABLE 6. Comparison of different methods on the *Office-31* dataset based on a ResNet-50 backbone in the two-source domain setting.

Standard	Source	A, W	A, D	W, D	Mean
	Target	D	W	A	
Single best	ResNet-50	99.3	96.7	62.5	86.2
	DAN [23]	99.5	96.8	66.7	87.7
	D-CORAL [30]	99.7	98.0	65.3	87.7
	RevGard [21]	99.1	96.9	68.2	88.1
	MADA [28]	99.6	97.4	70.3	89.1
	MRAN [32]	99.8	96.9	70.9	89.2
	ALDA [17]	100.0	97.7	72.5	90.1
	ETD [69]	100.0	**100.0**	71.0	90.3
	DMDA [16]	99.4	98.6	74.0	90.7
	DWL [68]	100.0	99.2	73.1	90.8
	DSAN [15]	100.0	98.3	**74.8**	91.0
	ATM [11]	100.0	99.3	74.1	91.1
Source combine	DAN [23]	99.6	97.8	67.6	88.3
	D-CORAL [30]	99.3	98.0	67.1	88.1
	RevGrad [21]	99.7	98.1	67.6	88.5
Multi-Source	LtC-MSDA [45]	99.6	97.2	56.9	84.6
	DCTN [40]	99.3	98.2	64.2	87.2
	MADAN [61]	99.4	98.4	63.9	87.2
	SImpAI [42]	99.2	97.4	70.6	89.0
	MFSAN [41]	99.5	98.5	72.7	90.2
	MSCLDA [49]	99.8	98.8	73.7	90.8
	DCMT (ours)	**100.0**	99.1	74.6	**91.2**

respectively. The proposed method significantly improved the target classification accuracy on the challenging large-scale benchmark, *DomainNet-126*, as shown in Table 5. Compared to the PCT [70], which produced the best performance among the SUDA methods, the average accuracy of the proposed method was 6.1% higher. The mean accuracy of the target domain increased up to 14.0% compared to AMDA [49]. Table 6 reports the results on the *Office-31* dataset. Our method recorded slightly higher accuracy in the average classification results compared to the previous studies.

Tables 7 and 8 showed the classification accuracies of the proposed method and the benchmark methods on *Office-Caltech10*, and *Office-Home*, in the three-source domain setting. In addition to results from the *single-best* and *multi-source* experiments, these tables contain *source-combined* results for the case in which multiple sources are concatenated as one source domain. Experiments on the *Office-Caltech10* dataset using the proposed method showed

TABLE 7. Comparison of different methods on the *Office-Caltech10* dataset based on a ResNet-50 backbone in the three-source domain setting.

Standard	Source	A, D, W	C, D, W	A, C, D	A, C, W	Mean
	Target	C	A	W	D	
Single best	ResNet-50	82.5	91.2	98.9	99.2	93.0
	ADDA [19]	88.8	94.5	99.1	98.0	95.1
	DCDA [70]	87.6	93.2	100.0	100.0	95.2
	CyCADA [29]	89.7	96.2	98.9	97.3	95.5
	WAN [66]	89.2	95.2	99.2	100.0	95.9
	DCAN [65]	90.4	94.7	99.1	100.0	96.1
Source combine	DAN [23]	89.7	94.8	99.3	98.2	95.5
	ADDA [19]	90.2	95.0	99.4	98.2	95.7
	CyCADA [29]	91.0	95.9	99.0	97.8	95.9
Multi-Source	DCTN [40]	90.2	92.7	99.4	99.0	95.3
	M^3SDA [44]	92.2	94.5	99.5	98.2	96.4
	SImpAI [42]	91.5	94.1	99.3	99.8	96.7
	MFSAN [41]	93.8	95.1	99.1	98.7	96.7
	MSCLDA [49]	94.1	95.3	99.1	98.5	96.8
	MULTI-EPL [74]	93.5	96.2	99.9	100.0	97.4
	TWMDA [13]	93.9	96.2	100.0	100.0	97.5
	DCMT (ours)	**94.4**	**96.3**	**100.0**	**100.0**	**97.7**

TABLE 8. Comparison of different methods on the *Office-Home* dataset based on a ResNet-50 backbone in the three-source domain setting.

Standard	Source	Ar, Cl, Pr	Ar, Cl, Rw	Ar, Pr, Rw	Cl, Pr, Rw	Mean
	Target	Rw	Pr	Cl	Ar	
Sb	ResNet-50	75.4	79.7	49.6	65.3	67.5
	D-CORAL [30]	76.3	80.3	53.6	67.0	69.3
	RevGard [21]	75.8	80.4	55.9	67.9	70.0
	DAN [23]	75.9	80.3	56.5	68.2	70.2
	MDDA [14]	77.8	81.8	57.6	67.9	71.3
	ALDA [17]	77.1	82.1	56.3	70.2	71.4
	MRAN [32]	77.5	**82.2**	60.0	70.4	72.5
Sc	D-CORAL [30]	**82.7**	79.5	58.6	68.1	72.2
	DAN [23]	82.5	79.0	59.4	68.5	72.4
	RevGrad [21]	**82.7**	79.5	59.1	68.4	72.4
Ms	AMDA [50]	78.4	74.8	49.6	60.9	65.9
	MADAN [61]	81.5	78.2	54.9	66.8	70.4
	MSCLDA [49]	80.6	79.9	61.4	**71.6**	73.4
	DCMT (ours)	80.9	79.5	**64.0**	70.4	**73.7**

better target classification accuracy than that of all benchmark methods. Our method could achieve perfect classification in some tasks such as **A**, **C**, **D**→**W** and **A**, **C**, **W**→**D**. On the *Office-Home* dataset, our method only showed outstanding classification accuracy in the case of **Ar**, **Pr**, **Rw**→**Cl**. However, the average classification accuracy on the target domain of the proposed method was the highest.

Table 9 reported the comparison results of the proposed method and existing methods on the *DomainNet-345* dataset with the five-source domain setting. The proposed method

TABLE 9. Comparison of different methods on the *DomainNet-345* dataset based on a ResNet-101 backbone in the five-source domain setting.

Method	inf, pnt, qdr rel, skt→clp	clp, pnt, qdr rel, skt→inf	clp, inf, qdr rel, skt→pnt	clp, inf, pnt rel, skt→qdr	clp, inf, pnt qdr, skt→rel	clp, inf, pnt qdr, rel→skt	Mean
DCTN [40]	48.6	23.5	48.8	7.2	53.5	47.3	38.2
M^3SDA [44]	58.6	26.0	52.3	6.3	62.7	49.5	42.6
LtC-MSDA [45]	63.1	28.7	56.1	16.3	66.1	53.8	47.4
DAEL [47]	70.8	26.5	57.4	12.2	65.0	60.6	48.7
DRT [48]	71.0	**31.6**	**61.0**	12.3	71.4	60.7	51.3
MSCAN [46]	69.3	28.0	58.6	**30.3**	73.3	59.9	53.2
TWMDA [13]	65.5	23.7	52.7	13.6	63.0	52.2	45.1
CMSDA [75]	71.0	26.6	57.6	21.3	68.1	59.5	50.7
DAC-Net [77]	72.5	27.6	57.8	23.0	66.7	59.5	51.2
DIDA [67]	73.6	28.6	58.7	21.2	68.9	60.4	51.9
DCMT (ours)	**74.1**	31.5	60.2	23.6	**73.3**	**62.1**	**54.1**

TABLE 10. Ablation studies for comparison of classification accuracy (%) between a single classifier and multiple classifiers on *DomainNet-126* dataset with $N = 3$ over four domain adaptation tasks using a ResNet-50 backbone.

Method		rel, pnt, skt→clp	clp, pnt, skt→rel	rel, clp, skt→pnt	clp, rel, pnt→skt	Mean
Single Classifier		65.47	65.36	64.27	60.0	63.78
Multiple Classifiers	Classifier 1	81.81	83.26	77.50	73.59	79.04
	Classifier 2	81.76	83.31	77.42	73.61	79.03
	Classifier 3	81.82	83.28	77.47	73.68	79.06

TABLE 11. Impact of each component on the proposed method. The experiments were conducted on the *DomainNet-126* dataset with $N = 3$ over four domain adaptation tasks using a ResNet-50 backbone.

Baseline	Adaptation	Minimax	Co-learning	rel, pnt, skt→clp	clp, pnt, skt→rel	rel, clp, skt→pnt	clp, rel, pnt→skt	Mean
✓				67.7	72.1	64.7	64.1	67.2
✓	✓			70.8	72.2	66.7	65.3	68.8
✓	✓	✓		76.3	77.0	70.1	68.4	73.0
✓	✓	✓	✓	**82.1**	**83.3**	**77.5**	**73.8**	**79.2**

was better than the second best method, MSCAN [45], 0.9% in the averaged classification accuracy on the target domain.

In general, multi-source domain adaptation showed superior results to those from a single source domain because it could access the rich information of multiple source domains. The proposed method also recorded improvements over simply combining all source domains, as it could solve the problem of bias in learning.

E. ABLATION STUDIES

In this section, first, we analyzed the impact of the number of classifiers on classification accuracy in the MSUDA setting. Second, we determined the best way to select appropriate pseudo labels by analyzing the classification accuracy variation of the proposed method according to the threshold value in Eq. (6). Third, we analyzed the contribution of each module in the proposed method: Baseline (BL), domain-level adaptation (DA), Co-learning, and minimax strategy (Minimax), as described in Section III. Finally, we analyzed the impact of the number of source domains for adaptation on the target domain.

1) ABLATION STUDIES FOR IMPACT OF THE NUMBER OF CLASSIFIERS IN THE MSUDA SETTING

Table 10 reported the comparison of classification accuracies of a single classifier and multiple classifiers for four domain adaptation tasks on the *DomainNet-126* dataset in the three-source domain setting. These results illustrated that the transfer performance of multiple classifiers was generally superior compared to the single classifier. Besides, the different classifiers successfully exchanged their knowledge to ensure that they provided similar predictions on the target samples, which indicated the efficiency of the Co-learning algorithm.

Figure 3 (a) provided the classification accuracies of single classifier and multiple classifiers on the *DomainNet-126* dataset in the two-source domain setting. As shown in this figure, when the numbers of source domains and classes in each domain are small, the number of classifiers constructed in the overall framework can less affect the classification performance. However, when the number of source domains increases, a single classifier can struggle to discriminate the various information within the same class containing the different domains, as shown in Fig. 3 (b). When both

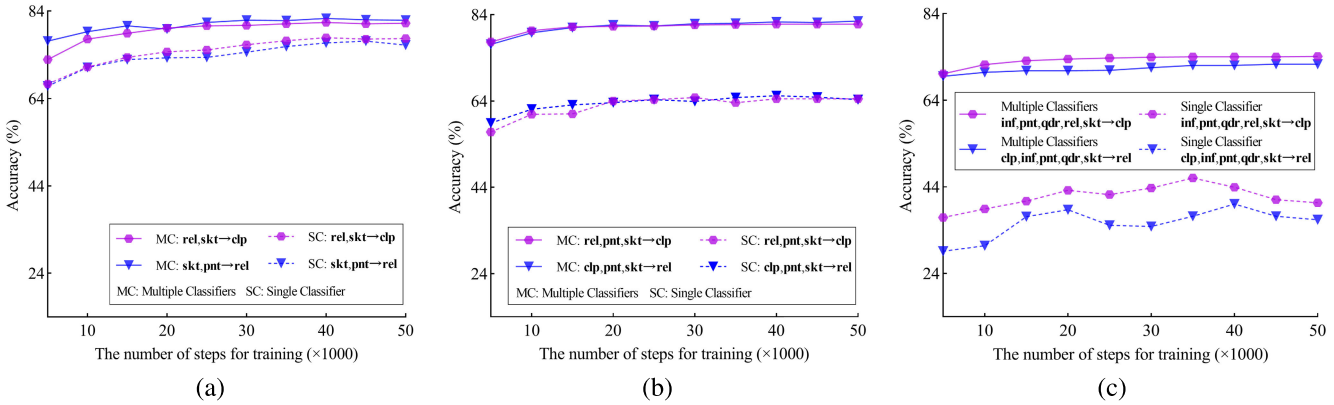

FIGURE 3. Comparison results of classification accuracies of a single classifier and multiple classifiers. Different colors indicate the different domain adaptation tasks. The dashed line denotes the classification accuracy of a single classifier, while the solid line denotes the classification accuracy of multiple classifiers. (a) results on the *DomainNet-126* dataset in the two-source domain setting. (b) results on the *DomainNet-126* dataset in the three-source domain setting. (c) results on the *DomainNet-345* dataset in the five-source domain setting.

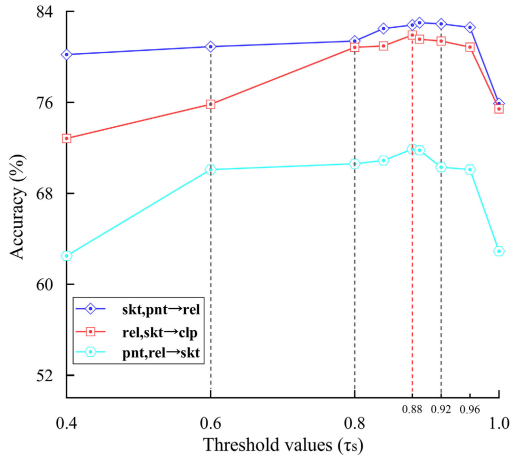

FIGURE 4. The impact of the threshold on the classification performance. The results were extracted from tasks **skt, pnt→rel; rel, skt→clp; and pnt, rel→skt** on the *DomainNet-126* dataset.

the numbers of source domains and classes in each source domain significantly increased, a single classifier showed the worst classification performance while multiple classifiers still provided outstanding classification results, as shown in Fig. 3 (c). In this figure, the classification accuracy by the multiple classifiers was slightly decreased when the number of source domains reached five because two additional source domains (*Quickdraw* and *Infograph*) contain very noisy labels.

2) SENSITIVITY OF THRESHOLD VALUE FOR SELECTING PSEUDO LABELS

The quality of pseudo labels affected the inference results on the target domain. We conducted a study to evaluate the impact of threshold values on selecting pseudo labels, as in [37]. Tasks **skt**, **pnt→rel**; **rel**, **skt→clp**; **pnt**, **rel→skt** on the *DomainNet-126* dataset were implemented with various threshold values τ in Eq. (6) using a ResNet-50 backbone. The threshold value corresponding to the best inference result on the target domain was selected, as reported in Fig. 4. As shown in this figure, when the threshold value was small ($\tau = 0.4 \sim 0.6$), the model generated many incorrect pseudo labels, which negatively affected the inference accuracy on the target domain. However, when τ was raised to 0.96, the model discarded useful information, which also led to a decrease in the classification performance on the target domain. It was obvious that the classification performance on the target domain was not sensitive when the threshold value was in an interval [0.6, 0.92], significantly stable in an interval [0.8, 0.92], and achieved the highest accuracy around 0.88, as indicated by the red dashed line. Thus, we selected 0.88 as the optimal threshold value for all our experiments.

3) IMPACT OF EACH MODULE IN THE PROPOSED METHOD ON THE TARGET LEARNER

The proposed method consisted of four modules: BL, DA, Minimax, and Co-learning. Each module had a different contribution to the classification accuracy of the target domain. We analyzed the impact of each module. The BL was implemented by training the shared feature extractor, and N classifiers over N labeled source domains to produce inferences on the target domain. For instance, in the results reported in Table 11 with $N = 3$ on task **rel**, **pnt**, **skt→clp**, three labeled source domains, **rel**, **pnt** and **skt**, transferred their knowledge to the target domain **clp**. In this case, three classifiers were trained to minimize misclassification of the labeled images **rel**, **pnt**, and **skt**, using Eq. (1). The shared feature extractor was trained for the correct classification of all **rel**, **pnt**, and **skt** images using Eq. (2). Finally, they were used to produce inferences on the target domain **clp** images. The average results of BL were just over 67%, as shown in Table 11, after applying the DA to the BL in which three discriminators were used to minimize the domain discrepancy between pairs (**rel**, **clp**), (**pnt**, **clp**), and (**skt**, **clp**), respectively. Thus, the mean classification accuracy on the target domain over four tasks slightly increased. In case (BL+DA), the cost

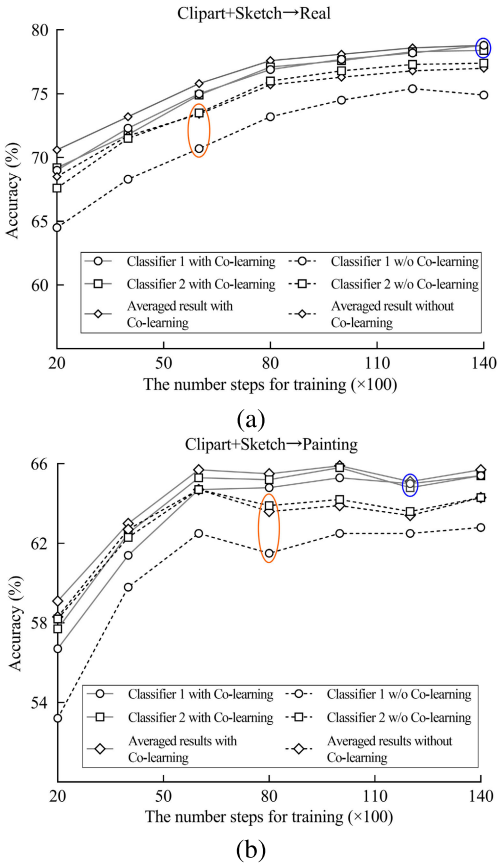

FIGURE 5. Test classification accuracy on the target domain of the proposed method with and without Co-learning. (a) results from **clp**, **skt**→**rel**, and (b) results from **clp**, **skt**→**pnt** on the *DomainNet-126* dataset.

function for the shared feature extractor was computed by the sum of Eqs. (2) and (4). The cost functions of each classifier and discriminator were computed respectively as Eqs. (1) and (3). When the minimax strategy in [36] was integrated into (BL+DA), it became (BL+DA+Minimax), and the average classification performance on the target domain was improved to 73%. Then, Co-learning was added to (BL+DA+Minimax). The average inference results on the target domain in (BL+DA+Minimax+Co-learning) reached over 79%. In both cases (BL+DA+Minimax) and (BL+DA+Minimax+Co-learning), the cost function to train the discriminators did not change. However, the total cost functions used to train the shared feature extractor and classifiers were updated as in Eqs. (9) and (10), respectively.

To investigate the efficiency of Co-learning for migrating the bias in learning in MSUDA, we extended our experiments on the *DomainNet-126* dataset over two tasks, **clp**, **skt**→**rel** and **clp**, **skt**→**pnt**, using (BL+DA+Minimax) and (BL+DA+Minimax+Co-learning), respectively. Because these scenarios were conducted with $N = 2$, target classifier 1 of model 1 extracted the target prediction over source domain 1 (**clp**), and target classifier 2 of model 2 extracted the target prediction over source domain 2 (**skt**). The final target prediction was averaged from the results of these two classifiers. The results are reported in Figs. 5 (a) and (b). Fig. 5(a) shows the results of **clp**, **skt**→**rel**, and Fig. 5 (b) shows the results of **clp**, **skt**→**pnt**. Without Co-learning, the output prediction results over the target domain of the two classifiers were significantly different, as indicated by the orange circles in Figs. 5 (a) and (b). In contrast, with Co-learning, the bias in prediction between these two classifiers almost disappeared. Even at the end of the training, the classification accuracies of the different classifiers were similar, as shown by the blue circles in Figs. 5 (a) and (b).

F. VISUALIZATION OF THE ANALYSIS OF THE PROPOSED METHOD

We visualized the feature distribution of the source and target domains, confusion matrices, and attention maps, to analyze the efficiency of the proposed method for MSUDA.

1) FEATURES VISUALIZATION

We visualized the feature distribution of multiple sources and target domains using t-SNE [57]. Figures 6 and 7 show the representations of multiple source domains and the target domain in two settings: two-source domain and three-source domain, respectively.

Figures 6 (a), (b), and (c) show the features of source domain 1 (**clp**), source domain 2 (**pnt**), and the target domain (**rel**) in the *DomainNet-126* dataset on task **clp**, **pnt**→**rel** with $N = 2$. Model 1 was trained on source domain 1 (**clp**), while model 2 was trained on source domain 2 (**pnt**). Then, they provided different views over the target domain (**rel**).

Given the view from model 1, this model had limited information from *class 1*. It could not generalize to classify *class 1* in the target domain, as indicated by the horizontal boundary red dashed line shown in Fig. 6 (a). However, the missed information from model 1 could be complemented by model 2 via the Co-learning algorithm, as denoted by the black box in Fig. 6 (b). Because model 2 is trained on source domain 2 (**pnt**), it contains abundant information of *class 1*. Similarly, following the view from model 2, source domain 2 (**pnt**) lacked the information to cover *classes 2* and *3* in the target domain. These were indicated by green and blue dashed lines of vertical boundaries 1 and 2, respectively, in Fig. 6 (b). However, this model could be generalized to the target domain because model 1 trained on source domain 1 (**clp**) could supplement the shortage of information about *classes 2* and *3* in source domain 2. This is illustrated by red boxes in Fig. 6 (a). The two source domains, **clp** and **pnt**, collaborated to transfer their knowledge to the target domain **rel**. The well-organized representations of the target domain are shown in Fig. 6 (c).

To show that our method could work well on various datasets, we ran the proposed method on task **A**, **D**, **W**→**C** ($N = 3$) using the *Office-Caltech10* dataset. The embedding space of source domain 1 (**A**), source domain 2 (**D**), source domain 3 (**W**), and the target domain (**C**) are represented in

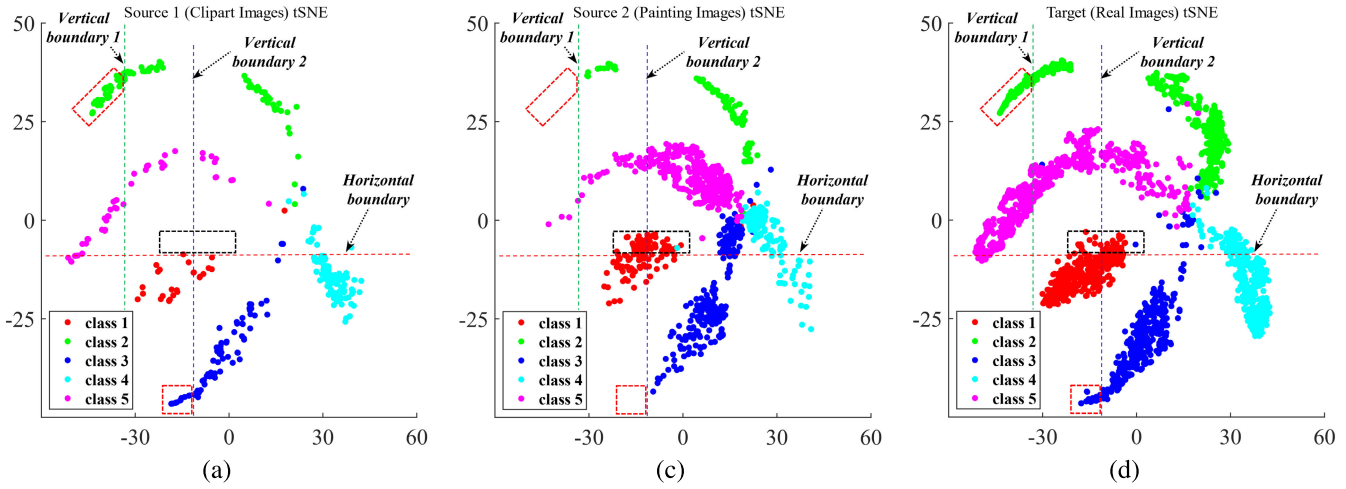

FIGURE 6. t-SNE visualization of different sources and target domains (two source domains) on the *DomainNet-126* dataset. Source domain 1 (*Clipart*) was used to train model 1, while model 2 was trained by using source domain 2 (*Painting*). Each model provided a different view of the target domain (*Real*). (a) The view from model 1. It could obtain more information from *classes 2* and *3* than model 2 for adaptation to the target domain (*Real*), as indicated by green and blue lines of the vertical boundaries 1 and 2, respectively. (b) The view from model 2, which held more information from *class 1* than model 1 to adapt to the target domain, as illustrated by the horizontal boundary red line. (c) Visualization of the representations of the target domain.

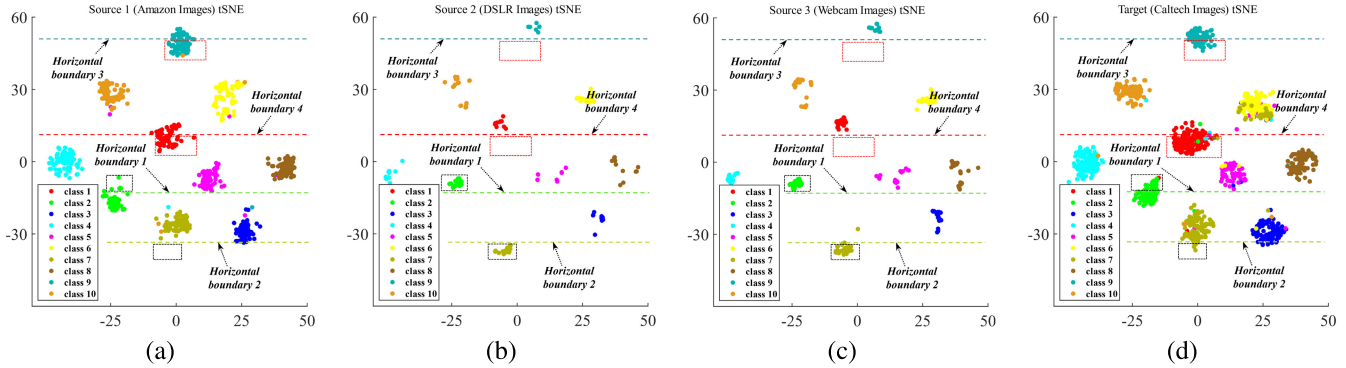

FIGURE 7. t-SNE visualization of different source and target domains (three source domains) on the *Office-Caltech10* dataset in case A, D, W→C. (a), (b), (c), and (d) show the representations of classes in the source and target domains with horizontal views.

Figs. 7 (a), (b), (c), and (d). Similar to the previous analysis for $N = 2$, if only source domain 1 was used to adapt to the target domain, some information about *class 2* and *class 7* in source domain 1 was missed. This is indicated by horizontal boundaries 1 and 2, and the missed information is denoted by the black boxes in Fig. 7 (a). Information from source domains 2 and 3 supplemented the shortage of information from source domain 1, to generalize the target domain, and the black boxes indicate the supplementary information in Figs. 7 (b) and (c). When only the knowledge of source domain 2 or source domain 3 was used for domain adaptation, the models trained on these source domains missed information from *class 1* and *class 9*. This situation is illustrated by horizontal boundaries 3 and 4, and the lost information is denoted by the red boxes in Figs. 7(b) and (c). However, source domain 1 supplemented this lost information from source domains 2 and 3 to adapt to the target domain, as displayed by the red boxes in Fig. 7 (a).

Using feature visualization analysis, we could observe that the complementary information from multiple views of the source domains could be integrated via Co-learning to elicit a good representation of the target domain.

2) CONFUSION MATRIX VISUALIZATION

Figures 8 (a) – (f) show the confusion matrix visualization of classifiers 1, 2, and 3 in the case $N = 3$ (three-source domain setting) on the *Office-Caltech10* dataset. Figures 8 (a) – (c) show the confusion matrix results of classifiers 1, 2, and 3, respectively, without Co-learning, while Figures 8(d) – (f) display the confusion matrix results of these classifiers with Co-learning.

Following the representations of source domains, as shown in Figs. 7 (a) – (c), source domain 1 held more information about *classes 5* and *7*, corresponding to *Keyboard* and *Monitor* in Fig. 8, than source domains 2 and 3. In cases without Co-learning, serious bias in learning occurred. Therefore, the classifier trained on source domain 1 performed well on these classes in the target domain compared with the classifiers trained on source domains 2 and 3, respectively, as shown in Figs. 8 (a) – (c). This problem almost disappeared when Co-learning was applied, as shown in Figs. 8 (d) – (f).

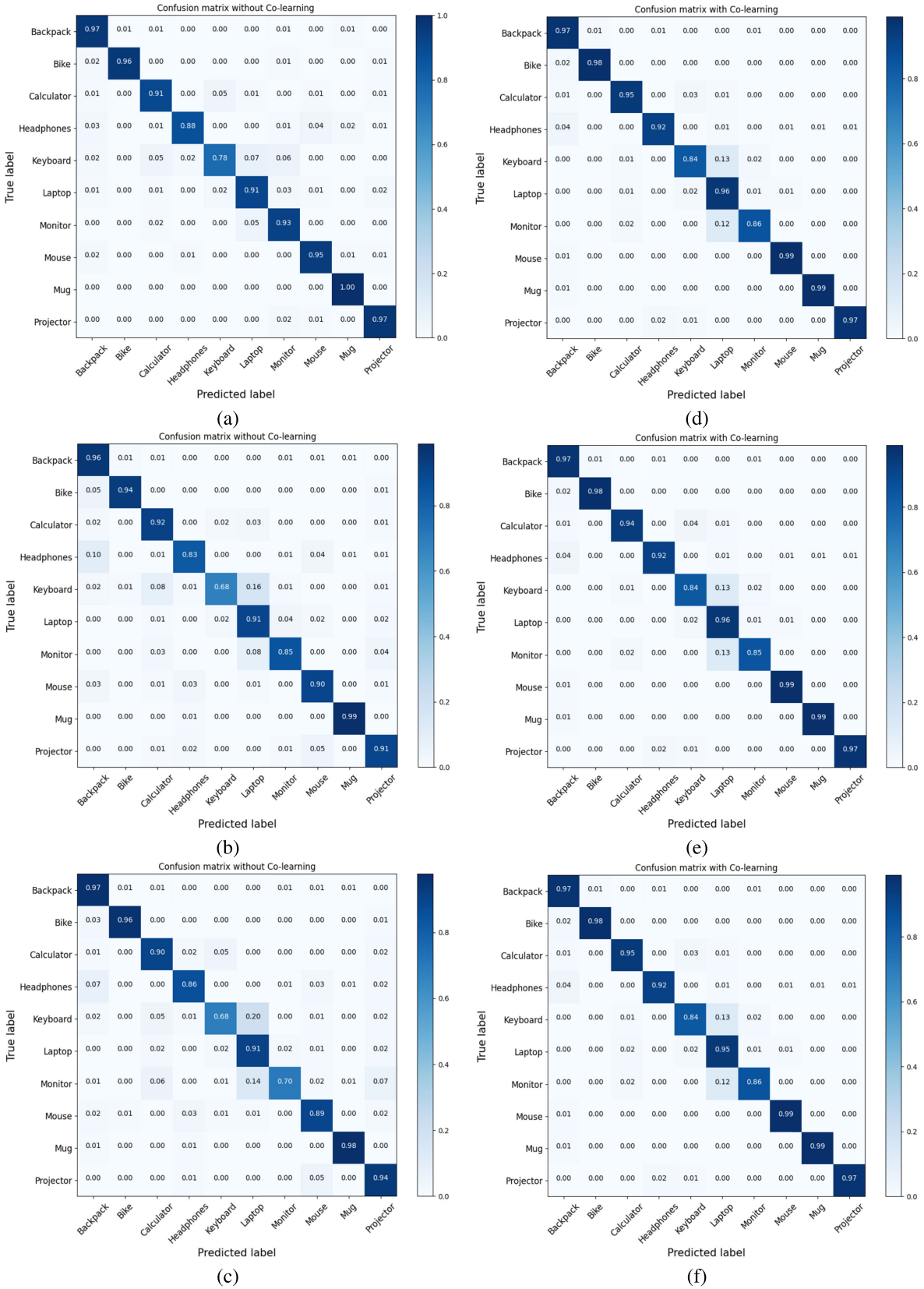

FIGURE 8. Confusion matrix visualization of different classifiers. These experiments were implemented on the *Office-Caltech10* dataset based on a ResNet-50 backbone on task **A, D, W→C**. (a), (b), and (c) show the confusion matrices for classifiers 1, 2, and 3 without Co-learning. (d), (e), and (f) show the confusion matrices for classifiers 1, 2, and 3 with Co-learning.

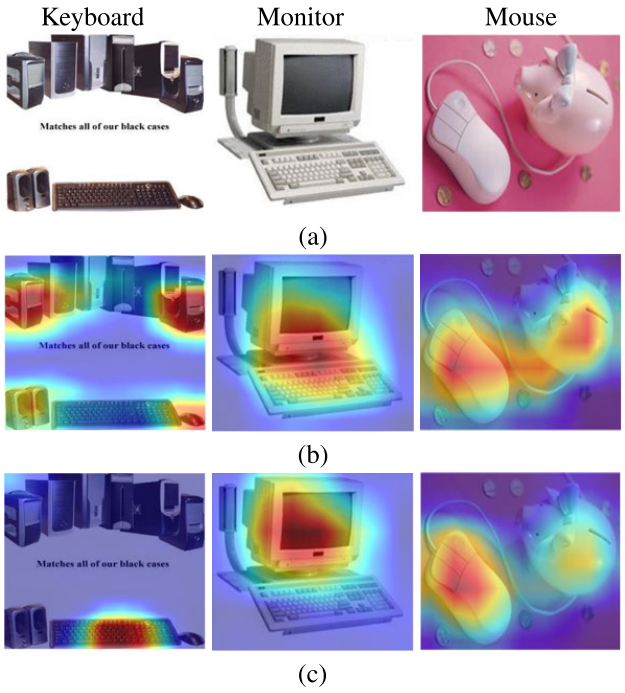

FIGURE 9. The features of the last convolutional layer in the ResNet-50 backbone were extracted for attention map visualization analysis. These results were obtained from task **A, D, W→C** on the *Office-Caltech10* dataset. Input images were randomly selected from *Keyboard*, *Monitor*, and *Mouse* classes in the target data. (a) The input images. (b) Results of the attention maps obtained from the model without Co-learning. (c) Results of the attention maps obtained from the model with Co-learning.

The classification performance on the target domain could be degraded because the classifier was confused by similar features that came from the different classes within the same domain, leading to negative transfer, a phenomenon called inter-class similarity. For example, as shown in Figs. 8 (b) – (c), the classifier found it hard to discriminate between *class 5* and *class 6* (*Laptop*) or *class 6* and *class 7*. The reason was that these classes contained many common features. This result was concordant with the feature visualization in Fig. 7 (d), in which a few representations of *classes 5*, *6*, and *7* overlapped. However, this problem was alleviated, as shown in Figs. 8 (d) – (f), using Co-learning.

The results reported in Figs. 4, 5, 6, and 7 indicated that Co-learning boosted target classification accuracy because it could mitigate the bias in learning caused by the imbalanced classes problem. Moreover, it allowed the different source domains to exchange their knowledge to alleviate negative transfer because of the inter-class similarity.

3) VISUALIZATION OF ATTENTION MAPS

We used GradCam [58] to visualize attention maps, in which the features of an input image extracted by the last convolutional layer in ResNet-50 were displayed. The experiments were conducted on the *Office-Caltech10* dataset with $N = 3$ on task **A**, **D**, **W→C**, to analyze the efficiency of Co-learning for classifying objects. The input images were randomly selected from classes *Keyboard*, *Monitor*, and *Mouse* in the target dataset. As shown in Figs. 9 (a) – (c), Co-learning enabled the adaptation model to focus on the main regions of objects. For example, the information about objects that were extracted by the models that used Co-learning was more discriminative than that from the models without Co-learning. As shown in Fig. 9 (c), the model with Co-learning could capture object information more accurately than the model without Co-learning, shown in Fig. 9 (b). The model without Co-learning was sensitive to noise from the background, while the model with Co-learning only concentrated on the main regions of the objects. These results were concordant with the confusion matrix analysis of Fig. 8 in Section IV. They also illustrated that our framework was successful in transferring content information across domains.

V. CONCLUSION

In this paper, we present the divide-and-conquer-based method for MSUDA. The complex problem of MSUDA was simplified by dividing it into many single-source-single-target tasks. The models were trained on different source domains in each task, each of which contained different views on the target domain. The conquering stage is then proposed to leverage information learned from each group, including multiple models, to benefit the training of the specific model. Using experiments, we showed that the proposed framework could alleviate the bias learning problem in MSUDA. The experimental results also showed that our method achieved state-of-the-art results on several benchmark DA datasets.

REFERENCES

[1] Y. J. Chae, S. J. Park, E. S. Kang, M. J. Chae, B. H. Ngo, and S. I. Cho, "Point2Lane: Polyline-based reconstruction with principal points for lane detection," *IEEE Trans. Intell. Transp. Syst.*, early access, Jul. 27, 2023, doi: 10.1109/TITS.2023.3295807.

[2] 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.

[3] D. Guan, J. Huang, A. Xiao, S. Lu, and Y. Cao, "Uncertainty-aware unsupervised domain adaptation in object detection," *IEEE Trans. Multimedia*, vol. 24, pp. 2502–2514, 2022.

[4] M. Biasetton, U. Michieli, G. Agresti, and P. Zanuttigh, "Unsupervised domain adaptation for semantic segmentation of urban scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1211–1220.

[5] S. J. Park, H. J. Park, E. S. Kang, B. H. Ngo, H. S. Lee, and S. I. Cho, "Pseudo label rectification via co-teaching and decoupling for multisource domain adaptation in semantic segmentation," *IEEE Access*, vol. 10, pp. 91137–91149, 2022.

[6] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2512–2521.

[7] B. H. Ngo, J. H. Kim, Y. J. Chae, and S. I. Cho, "Multi-view collaborative learning for semi-supervised domain adaptation," *IEEE Access*, vol. 9, pp. 166488–166501, 2021.

[8] J. H. Kim, B. H. Ngo, J. H. Park, J. E. Kwon, H. S. Lee, and S. I. Cho, "Distilling and refining domain-specific knowledge for semi-supervised domain adaptation," in *Proc. 33rd Brit. Mach. Vis. Conf. (BMVC)* 2022, pp. 1–14.

[9] B. H. Ngo, J. H. Kim, S. J. Park, and S. I. Cho, "Collaboration between multiple experts for knowledge adaptation on multiple remote sensing sources," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4707815.

[10] B. H. Ngo, J. H. Park, S. J. Park, and S. I. Cho, "Semi-supervised domain adaptation using explicit class-wise matching for domain-invariant and class-discriminative feature learning," *IEEE Access*, vol. 9, pp. 128467–128480, 2021.
[11] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, "Maximum density divergence for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3918–3930, Nov. 2021.
[12] Y.-H. Liu and C.-X. Ren, "A two-way alignment approach for unsupervised multi-source domain adaptation," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108430.
[13] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, and Q. Yang, "Transfer learning with dynamic distribution adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 1, pp. 1–25, Feb. 2020.
[14] Y. Zhu, F. Zhuang, J. Wang, G. Ke, J. Chen, J. Bian, H. Xiong, and Q. He, "Deep subdomain adaptation network for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1713–1722, Apr. 2021.
[15] S. Yao, Q. Kang, M. Zhou, M. J. Rawa, and A. Albeshri, "Discriminative manifold distribution alignment for domain adaptation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 2, pp. 1183–1197, Feb. 2023.
[16] M. Chen, S. Zhao, H. Liu, and D. Cai, "Adversarial-learned loss for domain adaptation," in *Proc. AAAI*, 2020, pp. 3521–3528.
[17] 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, Apr. 2016.
[18] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2962–2971.
[19] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.
[20] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.
[21] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
[22] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, Feb. 2015, pp. 97–105.
[23] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 2208–2217.
[24] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 4119–4125.
[25] M. Kan, S. Shan, and X. Chen, "Bi-shifting auto-encoder for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3846–3854.
[26] S. Yang, K. Yu, F. Cao, H. Wang, and X. Wu, "Dual-representation-based autoencoder for domain adaptation," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7464–7477, Aug. 2022.
[27] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. AAAI*, 2018, pp. 1–8.
[28] J. Hoffman, E. Tzeng, T. Park, J. Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1989–1998.
[29] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.
[30] C. Yu, J. Wang, Y. Chen, and M. Huang, "Transfer learning with dynamic adversarial adaptation network," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 778–786.
[31] Y. Zhu, F. Zhuang, J. Wang, J. Chen, Z. Shi, W. Wu, and Q. He, "Multi-representation adaptation network for cross-domain image classification," *Neural Netw.*, vol. 119, pp. 214–221, Nov. 2019.
[32] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu, "Equivalence of distance-based and RKHS-based statistics in hypothesis testing," *Ann. Statist.*, vol. 41, no. 5, pp. 2263–2291, Oct. 2013.
[33] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1855–1862.
[34] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 193–200.
[35] 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 (CVPRW)*, Jun. 2020, pp. 3008–3017.
[36] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8049–8057.
[37] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C. L. Li, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.* 2020, pp. 596–608.
[38] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 8559–8570.
[39] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3964–3973.
[40] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proc. AAAI*, vol. 33, Jul. 2019, pp. 5989–5996.
[41] N. Venkat, J. Kundu, D. Singh, A. Revanur, and V. R. Babu, "Your classifier can secretly suffice multi-source domain adaptation," in *Proc. NIPS*, 2020, pp. 1–13.
[42] C. Chen, W. Xie, Y. Wen, Y. Huang, and X. Ding, "Multiple-source domain adaptation with generative adversarial nets," *Knowl.-Based Syst.*, vol. 199, Jul. 2020, Art. no. 105962.
[43] 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.
[44] H. Wang, M. Xu, B. Ni, and W. Zhang, "Learning to combine: Knowledge aggregation for multi-source domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 727–744.
[45] G. Kang, L. Jiang, Y. Wei, Y. Yang, and A. Hauptmann, "Contrastive adaptation network for single- and multi-source domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1793–1804, Apr. 2022.
[46] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Trans. Image Process.*, vol. 30, pp. 8008–8018, 2021.
[47] Y. Li, L. Yuan, Y. Chen, P. Wang, and N. Vasconcelos, "Dynamic transfer for multi-source domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10993–11002.
[48] K. Li, J. Lu, H. Zuo, and G. Zhang, "Multi-source contribution learning for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5293–5307, Oct. 2022.
[49] Y. Wang, Z. Zhang, W. Hao, and C. Song, "Attention guided multiple source and target domain adaptation," *IEEE Trans. Image Process.*, vol. 30, pp. 892–906, 2021.
[50] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. 11th Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 213–226.
[51] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5385–5394.
[52] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.
[53] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, 2007.
[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[55] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[56] 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*, 2017, pp. 1–4.
[57] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[58] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[59] S. Zhao, B. Li, P. Xu, X. Yue, G. Ding, and K. Keutzer, "MADAN: Multi-source adversarial domain aggregation network for domain adaptation," *Int. J. Comput. Vis.*, vol. 129, no. 8, pp. 2399–2424, Aug. 2021.
[60] C. Han, D. Zhou, Y. Xie, M. Gong, Y. Lei, and J. Shi, "Collaborative representation with curriculum classifier boosting for unsupervised domain adaptation," *Pattern Recognit.*, vol. 113, May 2021, Art. no. 107802.
[61] Q. Zhou, W. A. Zhou, S. Wang, and Y. Xing, "Unsupervised domain adaptation with adversarial distribution adaptation network," *Neural. Comput. Appl.*, vol. 33, pp. 7709–7721, Mar. 2021.
[62] J. Li, S. Lü, W. Zhu, and Z. Li, "Enhancing transferability and discriminability simultaneously for unsupervised domain adaptation," *Knowl.-Based Syst.*, vol. 247, Jul. 2022, Art. no. 108705.
[63] Y. Chen, C. Yang, Y. Zhang, and Y. Li, "Deep conditional adaptation networks and label correlation transfer for unsupervised domain adaptation," *Pattern Recognit.*, vol. 98, Feb. 2020, Art. no. 107072.
[64] X. Jia and F. Sun, "Unsupervised deep domain adaptation based on weighted adversarial network," *IEEE Access*, vol. 8, pp. 64020–64027, 2020.
[65] Z. Deng, K. Zhou, D. Li, J. He, Y.-Z. Song, and T. Xiang, "Dynamic instance domain adaptation," *IEEE Trans. Image Process.*, vol. 31, pp. 4585–4597, 2022.
[66] N. Xiao and L. Zhang, "Dynamic weighted learning for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15237–15246.
[67] M. Li, Y.-M. Zhai, Y.-W. Luo, P.-F. Ge, and C.-X. Ren, "Enhanced transport distance for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13933–13941.
[68] L. Abdi and S. Hashemi, "Unsupervised domain adaptation based on correlation maximization," *IEEE Access*, vol. 9, pp. 127054–127067, 2021.
[69] Y. Jin, X. Wang, M. Long, and J. Wang, "Minimum class confusion for versatile domain adaptation," in *Proc. ECCV*, 2020, pp. 464–480.
[70] K. Tanwisuth, X. Fan, H. Zheng, S. Zhang, H. Zhang, B. Chen, and M. Zhou, "A prototype-oriented framework for unsupervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17194–17208.
[71] Z. Han, H. Sun, and Y. Yin, "Learning transferable parameters for unsupervised domain adaptation," *IEEE Trans. Image Process.*, vol. 31, pp. 6424–6439, 2022.
[72] S. Lee, H. Jeon, and U. Kang, "Multi-EPL: Accurate multi-source domain adaptation," *PLoS ONE*, vol. 16, no. 8, Aug. 2021, Art. no. e0255754.
[73] M. Scalbert, M. Vakalopoulou, and F. Couzinié-Devy, "Multi-source domain adaptation via supervised contrastive learning and confident consistency regularization," in *Proc. BMVC*, 2021, pp. 1–20.
[74] Y. Wei, L. Yang, Y. Han, and Q. Hu, "Multi-source collaborative contrastive learning for decentralized domain adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 5, pp. 2202–2216, May 2023.
[75] Z. Deng, K. Zhou, Y. Yang, and T. Xiang, "Domain attention consistency for multi-source domain adaptation," in *Proc. BMVC*, 2021, pp. 1–19.

BA HUNG NGO (Member, IEEE) received the B.S. degree in control engineering and automation from the Hanoi University of Mining and Geology, Hanoi, Vietnam, in 2014, the M.S. degree in control engineering and automation from the Hanoi University of Science and Technology, Hanoi, in 2016, and the Ph.D. degree in engineering from Dongguk University, Seoul, Republic of Korea, in 2023. His research interests include computer vision and deep learning, especially deep transfer learning, domain adaptation, deep learning in medical imaging, remote sensing image processing, and remote sensing data and applications.

YEON JEONG CHAE (Member, IEEE) received the B.S. degree in multimedia engineering from Dongguk University, Seoul, South Korea, in 2021, where she is currently pursuing the M.S. degree. Her research interests include lane detection, semantic segmentation, and domain adaptation.

SO JEONG PARK received the B.S. and M.S. degrees in multimedia engineering from Dongguk University, in 2021 and 2023, respectively. Her research interests include semantic segmentation and domain adaptation.

JU HYUN KIM received the B.S. degree in electronic and electrical engineering from Dongguk University, Seoul, South Korea, in 2019, where he is currently pursuing the M.S. degree in multimedia engineering. From 2019 to 2021, he was an Engineer with LG Display. His research interests include object detection and domain adaptation.

SUNG IN CHO (Member, IEEE) received the B.S. degree in electronic engineering from Sogang University, Seoul, South Korea, in 2010, and the Ph.D. degree in electrical and computer engineering from the Pohang University of Science and Technology, Pohang, South Korea, 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, Gyeongsan, South Korea. He is currently an Associate Professor with the Department of AISW Convergence, Dongguk University, Seoul. His research interests include image analysis and enhancement, video processing, computer vision, and deep learning.