IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 February 2024, accepted 1 March 2024, date of publication 5 March 2024, date of current version 13 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3374105

RESEARCH ARTICLE

Dual Dynamic Consistency Regularization for Semi-Supervised Domain Adaptation

BA HUNG NGO[1], BA THINH LAM[2], THANH HUY NGUYEN[3], QUANG VINH DINH[4], AND TAE JONG CHOI[1]

[1]Graduate School of Data Science, Chonnam National University, Gwangju 61186, Republic of Korea
[2]Faculty of Information Technology, University of Science, Ho Chi Minh City 700000, Vietnam
[3]Department of Biomedical Engineering, National Cheng Kung University, Tainan 70101, Taiwan
[4]School of Electrical Engineering and Computer Science, Vietnamese–German University, Ho Chi Minh City 72000, Vietnam

Corresponding author: Tae Jong Choi (ctj17@jnu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. RS-2023-00214326 and No. RS-2023-00242528).

ABSTRACT The Vision Transformer (ViT) model serves as a powerful model to capture and comprehend global information, particularly when trained on extensive datasets. Conversely, the Convolutional Neural Network (CNN) model is beneficial to training with small datasets for retaining essential local information. Inspired by these properties of ViT and CNN models, we introduce a hybrid framework that smoothly increases the cross-domain generalization in Semi-supervised Domain Adaptation (SSDA). To achieve this goal, we first train the ViT model on abundant labeled source data, while the CNN model is trained on a few labeled target samples. Then, these models dynamically exchange their knowledge for potential generalization to unlabeled target data via the proposed method, named Dual Dynamic Consistency Regularization (D^2CR). Specifically, the ViT model provides its pseudo labels to update the global perspective for the CNN model. Similarly, the CNN model offers pseudo labels to complement the local perspective for the ViT model. The previous methods use a fixed threshold algorithm for the pseudo-labeling process. However, we utilize the dynamic threshold strategy to create pseudo labels for the bi-directional consistency regularization learning between the ViT and CNN models. We verify our approach across several SSDA benchmark datasets. The outstanding experimental results provide strong evidence of the effectiveness and superiority of our approach over previous state-of-the-art SSDA methods.

INDEX TERMS Domain adaptation, semi-supervised learning, vision transformer (ViT), convolutional neural network (CNN).

I. INTRODUCTION

Domain adaptation is a crucial aspect of machine learning, focused on enhancing model efficacy in a target domain by exploiting knowledge insights from a related source domain. Various practical applications, including classification [1], [2], object detection [3], [4] and semantic segmentation [5], [6] have demonstrated success through domain adaptation techniques. Recent studies [7], [8] successfully employed domain adaptation for recommendation systems. These studies involved mapping user-side and item-side information into a shared space for improving the model performance.

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

Based on the number of labeled target data available during training, domain adaptation can be divided into two groups: semi-supervised domain adaptation (SSDA) and unsupervised domain adaptation (UDA).

The semi-supervised domain adaptation scenario, as depicted in [9], [10], [11], [12], [13], [14], [15], [16], [17], and [18], is extensively employed to yield remarkable classification accuracy compared to the unsupervised domain adaptation setting [19], [20], [21], [22], [23]. This is because a model trained under UDA is only accessed to labeled source data, while a model trained with the SSDA setting benefits from the extra target information with a few labeled target samples besides labeled source data. However, the amount of label information available from

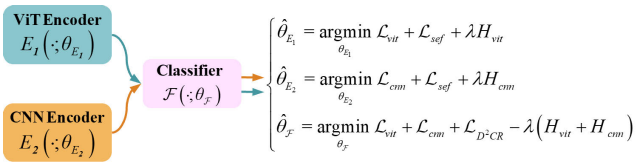


FIGURE 1. Illustration of the proposed hybrid network architecture that leverages the strengths of ViT and CNN models.

the source data is relatively significant compared to the label information from the target data. As a result, the model often prioritizes the abundant label information from source data, potentially paying less attention to target label information and introducing bias in learning. To address this problem, [9] and [14] divide SSDA into two different tasks, such as UDA [24], [25]–(*labeled source+unlabeled target*) and semi-supervised learning (SSL) [27], [28]–(*labeled target+unlabeled target*). Each task consists of the different labeled sets used to train the different models. Then, they exchange their knowledge through the co-training strategy to make consistency on the unlabeled target data. Many SSDA methods [9], [10], [11], [12], [13], [15], [17], [18] follow a similar methodology in designing network architecture for data representation extraction. They employ convolutional neural networks (CNNs) as encoders, followed by multilayer perceptrons (MLPs) as classifiers. However, the CNN model only benefits training with a small dataset [36].

Furthermore, SSDA [9], [11], [33] methods show outstanding classification results by exploiting the unlabeled target information. The core technique of these methods is to combine consistency regularization [28] and pseudo-labeling [38]. In addition, these approaches share the same point using a fixed threshold value to retain reliable samples with a high confidence prediction and simultaneously discard unreliable samples under the pre-defined threshold value. However, we recognize that each method provides distinct fixed threshold values. This is because their frameworks are designed in diverse manners; thus, they require different quantity and quality pseudo labels. As a result, it is not easy to reuse the threshold values of previous works for continuous potential research. Furthermore, due to the difference in characteristics of variant domain adaptation (DA) benchmark datasets, applying the optimal threshold value on a specific DA dataset for other DA datasets is challenging.

A. MOTIVATION

In this study, we further introduce a methodology to design a novel framework for SSDA that finds solutions to the following questions: can we bridge the different network architectures to build a framework that smoothly increases cross-domain generalization? and how to design a flexible consistency regularization strategy with the dynamic threshold?. For handling the first inquiry, we build a hybrid framework that leverages the properties of the ViT and CNN models, as illustrated in Figure 1. To be specific, we train the ViT model using the rich labeled source data, leveraging its strong generalization for global representations. Conversely, the CNN model is trained on scarce labeled target data, preserving crucial local details. Subsequently, these models share their knowledge to enhance potential generalization and achieve consistency in the target domain. To achieve this goal, we leverage the dynamic threshold approach [37] to propose a novel dynamic consistency regularization method called Dual Dynamic Consistency Regularization (D^2CR). In D^2CR, the ViT model updates the global information to the CNN model using pseudo labels. Similarly, the CNN model offers its pseudo labels to provide the local information for the ViT model. The dynamic threshold algorithm determines both pseudo labels created by the ViT and CNN models. By doing so, we can find a solution to the second question by leveraging the dynamic threshold to enhance consistency between CNN and ViT models.

B. CONTRIBUTIONS

The contributions of our method can be listed as follows:

- In this work, we provide a new perspective that smoothly combines the learning behaviors of ViT and CNN backbone networks to improve cross-domain generalization.
- We point out that the consistency regularization process in previous SSDA methods still has much room for improvement with the fixed threshold values. To address this problem, we introduce a novel consistency regularization learning utilizing the dynamic threshold algorithm.
- The experimental results demonstrate that the proposed D^2CR method effectively handles bias learning in SSDA.
- The proposed D^2CR method surpasses the prior works to achieve a new SoTA method on several challenging SSDA benchmark datasets.

C. ORGANISATION

The remainder of the paper is structured in the following manner. Section II discusses related work, highlighting various concepts for designing semi-supervised domain adaptation (SSDA) frameworks. It also provides an overview of consistency regularization methods commonly used in SSDA. Section III delves into the details of the proposed dual dynamic consistency regularization method. Section IV presents descriptions of the datasets used in the experiments, the settings employed for implementation, and the evaluation metrics utilized. Section V provides results comparing the proposed method with previous SSDA approaches to demonstrate its effectiveness. Section VI includes ablation studies and analyses that evaluate the impact of each component in the proposed method and the influence of the dynamic threshold algorithm. Section VII offers visualization results to illustrate how the proposed method operates intrinsically. Finally, Section VIII summarizes remarkable points and suggests potential directions for future research.

TABLE 1. Comparison of our method with the previous methods in terms of components in training and testing phases.

Method	Training Phase		Testing Phase	
	Encoder	Classifier	Encoder	Classifier
MME	CNN	MLP	CNN	MLP
Con^{2}DA	CNN	MLP	CNN	MLP
APE	CNN	MLP	CNN	MLP
S^{3}D	CNN	MLP	CNN	MLP
STar	CNN	MLP	CNN	MLP
CLDA	CNN	MLP	CNN	MLP
ECACL	CNN	MLP	CNN	MLP
CDAC	CNN	MLP	CNN	MLP
CDAC+SLA	CNN	MLP	CNN	MLP
ASDA	CNN	MLP+MLP	CNN	MLP+MLP
MVCL	CNN	MLP+MLP	CNN	MLP+MLP
DECOTA	CNN+CNN	MLP+MLP	CNN+CNN	MLP+MLP
D^{2}CR	ViT+CNN	MLP	CNN	MLP

II. RELATED WORK

This section discusses various concepts for designing the SSDA frameworks and the consistency regularization technique.

A. METHODOLOGY DESIGN THE SSDA FRAMEWORK

To efficiently delve into training data representations and mitigate empirical risk, distinct SSDA methods introduce varied frameworks. For instance, earlier approaches like MME [29], STar [39], and APE [30] adopt the conventional approach of constructing deep learning frameworks. These frameworks comprise a feature extractor coupled with a classifier. UODA [32] and ASDA [12] demonstrate the effectiveness of dual classifiers in improving classification ability. Specifically, ASDA [12] uses two classifiers for alleviating confirmation bias to select the hard pseudo label. These methods use a single feature extractor to train on both labeled source and target datasets. Nevertheless, the larger number of labeled source samples compared to a few labeled target samples causes the feature extractor to primarily focus on representations from the source data, leading to a bias in learning that favors the source data. To solve this problem, DeCoTa [14] propose a novel network architecture by unifying two different models, each consisting of a feature extractor and a single classifier. One model is trained on labeled source data, while the other focuses on labeled target domain data. The aforementioned SSDA approaches differ only in the number of components within their frameworks. However, the fundamental elements remain consistent and still follow the rule: employing the CNN model for the feature extractor and the MLP for the classifier.

B. VARIANTS OF SSDA FRAMEWORKS

Table 1 illustrated the comparison between our method and the previous methods in terms of components operating in the training and testing phases. Specifically, [1], [11], [13], [15], [29], [30], [31], [33], and [39] utilized a combination of a CNN encoder followed by the MLP classifier in both stages. To alleviate bias confirmation, [9] and [12] introduced a new framework by adding one more MLP classifier, which consists of a single CNN encoder and two MLP classifiers. Reference [14] was the first SSDA approach that used two CNN encoders and two MLP classifiers with the co-training strategy to boost the classification accuracy. However, these above-mentioned SSDA methods utilized the different components in their frameworks, but the methodology in designing still followed the same rule, where the CNN model was selected as the encoder, and MLP was assigned as the classifier. We proposed an approach that leverages the strengths of ViT [34] and CNN [36] architectures to build a novel hybrid framework for the SSDA setting.

C. CONSISTENCY REGULARIZATION FOR SSDA

Prior SSDA methods [1], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] successfully combine consistency regularization [28] and pseudo labeling [38] to leverage unlabeled target data, yielding impressive classification outcomes. A common feature of these above-mentioned methods is that they use the fixed threshold to filter out the high confidence score for generating pseudo labels. However, we found that each method provides different threshold values. For instance, DeCoTa [14] and ECACL [13] set the threshold $\tau = 0.5$ and $\tau = 0.8$, respectively, for selecting their pseudo labels. Besides, CDAC [1], MVCL [9], and ASDA [12] choose $\tau = 0.95$ for all experiments over various DA datasets. In contrast, Con^{2}DA [33] offers $\tau = 0.9$ and $\tau = 0.8$ associated with AlexNet [44] and ResNet [45] to deploy on the *DomainNet* [42] dataset, while $\tau = 0.95$ is set to implement on the *Office-Home* [41] and *Office-31* [40] datasets. Variability of fixed threshold values arises due to divergent optimization processes in different approaches, requiring distinct quantities and qualities of pseudo labels. Consequently, prior threshold values can not be readily reproduced for continuous potential research. Moreover, dissimilar characteristics among diverse domain adaptation benchmark datasets make applying an optimal threshold value from one dataset to another challenging.

III. METHODOLOGY

In this section, we first introduce the problem setting in SSDA. We then present the proposed method Dual Dynamic Consistency Regularization (D^{2}CR) approach. We explicitly divide the training procedure of the proposed method into three steps: supervised learning on the labeled set, dual dynamic consistency regularization on the unlabeled target data, and minimizing divergence across domains, respectively.

A. PROBLEM DEFINITION OF SSDA

In the image classification task under the SSDA setting, we are provided the labeled set consisting of the rich labeled source samples $\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$ with additional

TABLE 2. Notations and descriptions.

Notations	Descriptions
τ	The fixed threshold
$\mathcal{D}_S$	The set of source samples.
x_i^S	The i-th sample in the source domain.
y_i^S	The label of the i-th sample in the source domain.
n_S	The number of source samples.
$\mathcal{D}_{T_l}$	The set of labeled target samples.
$x_i^{T_l}$	The i-th labeled sample in the target domain.
$y_i^{T_l}$	The label of the i-th sample in the target domain.
n_{T_l}	The number of labeled target samples.
$\mathcal{D}_{T_u}$	The set of unlabeled target samples.
$x_i^{T_u}$	The i-th unlabeled target sample.
n_{T_u}	The number of unlabeled target samples.
$\mathcal{D}_l$	The set of labeled samples.
$E_1(\cdot;\theta_{E_1})$	The ViT encoder.
$E_2(\cdot;\theta_{E_2})$	The CNN encoder.
$\mathcal{F}(\cdot;\theta_{\mathcal{F}})$	The shared classifier.
$\mathbf{q}_i^{vit}$	The prediction of the ViT branch on the weakly augmented version of $x_i^{T_u}$.
$\mathbf{p}_i^{cnn}$	The prediction of the CNN branch on the strongly augmented version of $x_i^{T_u}$.
$\mathbf{q}_i^{cnn}$	The prediction of the CNN branch on the weakly augmented version of $x_i^{T_u}$.
$\mathbf{p}_i^{vit}$	The prediction of the ViT branch on the strongly augmented version of $x_i^{T_u}$.
t	The training step.
$f(\theta,\xi^{T_u})$	The cross-entropy loss.
ρ_t	The dynamic threshold at the step t.
$\hat{\rho}$	The averaged loss from the labeled training set.
ρ_t^{vit}	The dynamic threshold value for the ViT branch at step t.
$\hat{\rho}_{vit}$	The averaged loss of the ViT branch at the step t.
ρ_t^{cnn}	The dynamic threshold value for the CNN branch at step t.
$\hat{\rho}_{cnn}$	The averaged loss of the CNN branch at the step t.
K	The number of classes.

information of a few labeled data $\mathcal{D}_{T_l} = \{(x_i^{T_l}, y_i^{T_l})\}_{i=1}^{n_{T_l}}$ from the target domain. The source and target samples, x_i^S and $x_i^{T_l}$, are attached with their ground-truth labels corresponding to y_i^S and $y_i^{T_l}$, respectively. n_S and n_{T_l} are the size of labeled samples in source and target domains, respectively. Let $\mathcal{D}_{T_u} = \{(x_i^{T_u})\}_{i=1}^{n_{T_u}}$ denotes the unlabeled target set without ground-truth class information during training, where n_{T_u} is the number of unlabeled target samples. In this research, we only consider the closed-set domain adaptation setting, where the y^S, y^{T_l}, and y^{T_u} share the same label space with K categories, where y^{T_u} is the label of the unlabeled target data, which only is used in the testing phase. Besides, the number of labeled target samples n_{T_l} is less than the number of labeled source n_S and unlabeled target samples n_{T_u}, respectively. Notations frequently used in this paper are listed in Table 2.

Our final objective is to optimize a model on datasets $\mathcal{D}_S$, $\mathcal{D}_{T_l}$, and $\mathcal{D}_{T_u}$, and demonstrates effective performance on $\mathcal{D}_{T_u}$.

To achieve this goal, we proposed a novel framework including three components: a ViT encoder $E_1(\cdot;\theta_{E_1})$, a CNN encoder $E_2(\cdot;\theta_{E_2})$ and a shared classifier $F(\cdot;\theta_F)$. As illustrated in Figure 2, the proposed framework is divided into two branches such as ViT and CNN branches. The ViT branch consists of E_1 and F, while the CNN branch is built from E_2 and F.

B. MODEL-BASED REPRESENTATION LEARNING

We train the ViT branch on the labeled source data by using the standard cross-entropy loss, which is defined as follows:

$$\mathcal{L}_{vit}^{src} = -\frac{1}{n_S}\sum_{i=1}^{n_S}\sum_{c=1}^{C} y_{i,c}^S \log(\sigma(F(E_1(x_{i,c}^S)))), \tag{1}$$

where $(x_{i,c}^S, y_{i,c}^S)$ is a pair of the input image corresponding to its ground-truth in the class $c \in [1, 2, \ldots, C]$. E_1 maps the i-th source sample into the feature space having dimension d as $E_1(x_i^S) \in \mathbb{R}^d$, and σ is the `softmax` function. Then, the shared classifier F categorizes the input feature $E_1(x_i^S)$ into the K-way classification. Finally, we minimize the distributions between the source probability predicted by $F(E_1(x_i^S))$, and the given source ground-truth label, y_i^S, using the cross-entropy loss.

The process to train the CNN branch is conducted similarly but on the labeled target samples as follows:

$$\mathcal{L}_{cnn}^{tar} = -\frac{1}{n_{T_l}}\sum_{i=1}^{n_{T_l}}\sum_{c=1}^{C} y_{i,c}^{T_l} \log(\sigma(F(E_2(x_{i,c}^{T_l})))), \tag{2}$$

where $E_2(x_i^{T_l})$ is the representations of the i-th target sample extracted the CNN encoder. $F(E_2(x_i^{T_l}))$ is the output prediction of the CNN branch driven by the labeled target ground-truth label, $y_i^{T_l}$.

C. BACKGROUND

This section provides the background and analyses the difference between fixed and dynamic threshold algorithms.

1) FIXED THRESHOLD

Existing SSDA methods [1], [9], [12], [13], [17] successfully leverage the unlabeled target information by using the pseudo labels [38] with the fixed threshold for the consistency regularization loss [28], which is defined as follows:

$$\mathcal{L}_{fix}^{unl} = -\frac{1}{n_{T_u}}\sum_{i=1}^{n_{T_u}} \mathbb{1}\big[\max(q_i) \geq \tau\big] \cdot \hat{y}_i \log(p_i), \tag{3}$$

where $q_i = p(\alpha(x_i^{T_u}))$ is the prediction vector of "*weak*" augmentation image with the weak perturbation function, α, that simply translates or flips an input image without changing its appearance; then, the highest probability value of q_i that overs a fixed threshold τ is converted by a one-hot

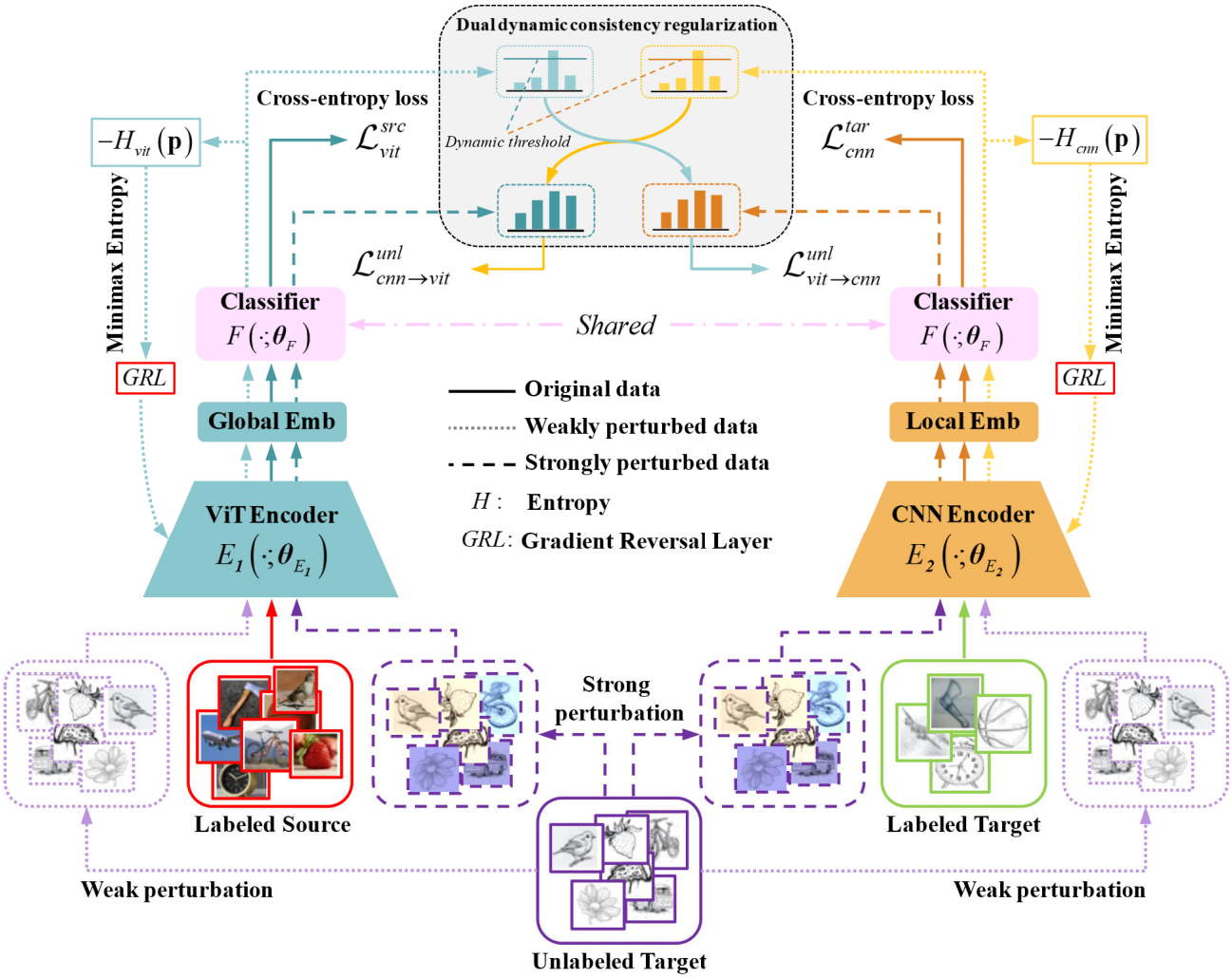


FIGURE 2. Illustration of dual dynamic consistency regularization (D^2CR) framework. We use ViT to build E_1 that can effectively exploit the global information in rich labeled source data. On the contrary, we select CNN for the second encoder E_2 to extract the local information on a few labeled target samples. These encoders share a common classifier F.

encoder into a hard label, $\hat{y}_i = \texttt{argmax}(q_i)$, where $\hat{y}_i$ is called "*pseudo-label*". Finally, the consistency regularization loss is formulated by using the cross-entropy loss to drive the prediction $p_i = p(\beta(x_i^u))$ of a heavily perturbed image to be closed to the pseudo-label $\hat{y}_i$, where β is the strong perturbation function [46].

2) DYNAMIC THRESHOLD

As shown in [37], the dynamic threshold ρ at step t is defined by:

$$\rho_t = C\gamma^{-(t-1)}\hat{\rho}, \tag{4}$$

where $C = 1.0001$, $\gamma = \{1.01, 1.1, 1.2, 1.3\}$, t is the training step, and $\hat{\rho}$ is the averaged loss from the labeled training set $\mathcal{D}_l$. $\hat{\rho}$ is approximated as follows:

$$\hat{\rho} \approx \frac{1}{|\mathcal{D}_l|}\sum_{\xi_i \in \mathcal{D}_l} f(\theta, \xi_i), \tag{5}$$

where θ are the learned parameters of the deep learning model trained on $\xi_i = (x_i, y_i) \in \mathcal{D}_l$, where f is the cross-entropy loss. As represented in Equation (4), the variation of the dynamic threshold ρ_t depends on the training time t and averaged loss of the trained model on the labeled dataset, $\hat{\rho}$.

3) DIFFERENCE BETWEEN FIXED AND DYNAMIC THRESHOLDS

As presented in Equations (3), (4), and (5), we recognize that there are two critical points to distinguish the fixed threshold [28] and dynamic threshold [37] algorithms.

- The first is the difference in flexibility between the fixed and dynamic threshold. Specifically, the fixed threshold is set manually and unchanged throughout the whole training process. Furthermore, the various datasets can require different fixed threshold values [33]; thus, it introduces additional processes. In contrast, the dynamic threshold can be automatically updated during training progress based on the behavior of the trained model.

- The second is the mechanism for assigning pseudo labels of the fixed and dynamic threshold algorithms. In the fixed threshold algorithm, an unlabeled target sample is assigned a pseudo label based on the output prediction of the trained model. Specifically, $x_i^{T_u}$ directly receives $\hat{y}_i = \texttt{argmax}(p(x_i^{T_u}))$ as its pseudo label when $\texttt{max}(p(x_i^{T_u})) \geq \tau$. Conversely, the dynamic threshold algorithm relies on the cross-entropy loss to determine pseudo labels. To be specific, $\hat{y}_i = \texttt{argmax}(p(x_i^{T_u}))$ is recognized the reliable pseudo label of $x_i^{T_u}$ if and only if the cross-entropy loss $f(\theta, \xi_i^{T_u}) \leq \rho_t$, where $\xi_i^{T_u} = (x_i^{T_u}, \hat{y}_i)$, and ρ_t is a dynamic threshold at step t computed as in Equation (4).

D. DUAL DYNAMIC CONSISTENCY REGULARIZATION

In Figure 2, the proposed method consists of ViT and CNN branches. The ViT branch is trained on the labeled source domain, while the CNN branch is trained on the labeled target domain. Therefore, they should be followed by different threshold values. To be specific, the dynamic threshold value for the ViT branch is defined as follows:

$$\rho_t^{vit} = C\gamma^{-(t-1)}\hat{\rho}_{vit}, \tag{6}$$

where $\hat{\rho}_{vit}$ is the averaged loss of the ViT branch on the labeled source data, which is computed as follows:

$$\hat{\rho}_{vit} \approx \frac{1}{|\mathcal{D}_S|} \sum_{\xi_i^S \in \mathcal{D}_S} f(\theta_{vit}, \xi_i^S), \tag{7}$$

with $\xi_i^S = (x_i^S, y_i^S) \in \mathcal{D}_S$, and $\theta_{vit} = (\theta_F, \theta_{E_1})$.

Similarly, the dynamic threshold for the CNN branch is defined as follows:

$$\rho_t^{cnn} = C\gamma^{-(t-1)}\hat{\rho}_{cnn}, \tag{8}$$

where $\hat{\rho}_{cnn}$ is the averaged loss of the CNN branch on the labeled target data, calculated as follows:

$$\hat{\rho}_{cnn} \approx \frac{1}{|\mathcal{D}_{T_l}|} \sum_{\xi_i^{T_l} \in \mathcal{D}_{T_l}} f(\theta_{cnn}, \xi_i^{T_l}), \tag{9}$$

with $\xi_i^{T_l} = (x_i^{T_l}, y_i^{T_l}) \in \mathcal{D}_{T_l}$, and $\theta_{cnn} = (\theta_F, \theta_{E_2})$.

We first use the pseudo labels extracted by the ViT branch to teach the CNN branch by using the dynamic threshold ρ_t^{vit} in Equation (6) as follows:

$$\mathcal{L}_{vit\to cnn}^{unl} = -\frac{1}{n_{T_u}} \sum_{i=1}^{n_{T_u}} \mathbb{1}\left[f(\theta_{vit}, \xi_{i,vit}^{T_u}) \leq \rho_t^{vit}\right] \cdot \hat{y}_i^{vit} \log(p_i^{cnn}), \tag{10}$$

where $\mathbb{1}[\cdot]$ is the indicator function, which returns 1 when $[\cdot]$ is true and 0 otherwise. $\xi_{i,vit}^{T_u} = (x_i^{T_u}, \hat{y}_i^{vit})$ with $\hat{y}_i^{vit} = \texttt{argmax}(q_i^{vit})$ is the pseudo label of the unlabeled target sample $x_i^{T_u}$ generated by the ViT branch, where $q_i^{vit} = \sigma(F(E_1(\alpha(x_i^{T_u}))))$ is the prediction vector of the ViT branch on the weakly augmented version of the unlabeled target sample with the weak perturbation function, α—(*random horizontal flip and random crop*). Then, the highest prediction that has cross-entropy loss under ρ_t^{vit} in Equation (6) to be converted to a one-hot label.

Finally, the output prediction $p_i^{cnn} = \sigma(F(E_2(\beta(x_i^{T_u}))))$ of the CNN branch on the strongly augmented target data is pushed to nearby the pseudo label $\hat{y}_i^{vit}$ using the cross-entropy loss, where β is the strong perturbation function [46].

Similarly, we use the generated pseudo labels of the CNN branch to train the ViT branch as follows:

$$\mathcal{L}_{cnn\to vit}^{unl} = -\frac{1}{n_{T_u}} \sum_{i=1}^{n_{T_u}} \mathbb{1}\left[f(\theta_{cnn}, \xi_{i,cnn}^{T_u}) \leq \rho_t^{cnn}\right] \cdot \hat{y}_i^{cnn} \log(p_i^{vit}), \tag{11}$$

where $\xi_{i,cnn}^{T_u} = (x_i^{T_u}, \hat{y}_i^{cnn})$, and $\hat{y}_i^{cnn} = \texttt{argmax}(q_i^{cnn})$ is the pseudo label of $x_i^{T_u}$ created by the CNN branch with $q_i^{cnn} = \sigma(F(E_2(\alpha(x_i^{T_u}))))$. Then, $\hat{y}_i^{cnn}$ is converted to a one-hot label if and only if its cross-entropy loss is less than the dynamic threshold ρ_t^{cnn} in Equation (8). $p_i^{vit} = \sigma(F(E_1(\beta(x_i^{T_u}))))$ is the output prediction of the ViT branch on the strongly augmented target image.

Equations (10) and (11) represent the mutual and dynamic consistency regularization that operates in both directions between the CNN and ViT branches. These regularization terms promote the exchange of local and global representations between these two branches for teaching each other. The ultimate objective is to enable them to generate similar predictions for the same unlabeled target sample. The computation of the dual dynamic consistency regularization loss is as follows:

$$\mathcal{L}_{D^2CR} = \mathcal{L}_{vit\to cnn}^{unl} + \mathcal{L}_{cnn\to vit}^{unl}. \tag{12}$$

E. MINIMIZING DISTRIBUTION DIVERGENCE

1) A QUICK RECAP OF THE MINIMAX ENTROPY STRATEGY

The classification function to predict the unlabeled target sample in [29] is presented as follows:

$$p(y_i|x_i^{T_u}) = \sigma\left[\frac{1}{T} \cdot \frac{W^\top E(x_i^{T_u})}{||E(x_i^{T_u})||}\right], \tag{13}$$

where $p(y_i|x_i^{T_u}) \in \mathbb{R}^K$ is the probability of the i-th unlabeled target sample, which is the $\texttt{softmax}$ values of the cosine similarity between $\mathbf{W}$ and $E(x_i^{T_u})$. $E(x_i^{T_u}) \in \mathbb{R}^d$ denotes the representations of the i-th unlabeled target sample extracted by the feature extractor E. d is the embedding feature dimension, which varies following the different backbone networks selected as the encoders. $W = [w_1, \ldots, w_K]$ consists of K weight vectors corresponding to K columns in the classifier F, where $W \in \mathbb{R}^{d\times K}$, K is the number of classes, and T is the temperature.

In our framework, the ViT branch is proposed to operate on the labeled source and unlabeled target data, which exists the inter-domain discrepancy [9]. In contrast, the CNN branch works with the labeled and unlabeled target data, which

Algorithm 1 Dual Dynamic Consistency Regularization

Require: The labeled source data $\mathcal{D}_S = \{x^S, y^S\}$, $\xi_i^S = (x_i^S, y_i^S)$; the labeled target data $\mathcal{D}_{T_l} = \{x^{T_l}, y^{T_l}\}$, $\xi_i^{T_l} = (x_i^{T_l}, y_i^{T_l})$; the unlabeled target data $\mathcal{D}_{T_u} = \{x^{T_u}\}$. $\mathcal{D}_l = \mathcal{D}_S \cup \mathcal{D}_{T_l}$.
Require: The number of training step N. Two hyper-parameters $C = 1.0001$ and $\gamma = 1.1$ to compute dynamic thresholds.
Require: Network components E_1, E_2 and F with parameters θ_{E_1}, θ_{E_2} and θ_F, respectively.
for $t = 1, 2, \ldots, N$ **do**
  # Model-based Representation Learning
  ▷ *Use the standard cross-entropy*
  Update θ_{E_1} by minimizing $\mathcal{L}_{vit}^{src}$ as in Equation (1).
  Update θ_{E_2} by minimizing $\mathcal{L}_{cnn}^{tar}$ as in Equation (2).
  Update θ_F by minimizing $\mathcal{L}_{vit}^{src} + \mathcal{L}_{cnn}^{tar}$.
  # D^{2}CR: Dual Dynamic Consistency Regularization
  $\hat{\rho}_{vit} \leftarrow \frac{1}{|\mathcal{D}_S|} \sum_{\xi_i^S \in \mathcal{D}_S} f(\theta_{vit}, \xi_i^S)$.
  $\hat{\rho}_{cnn} \leftarrow \frac{1}{|\mathcal{D}_{T_l}|} \sum_{\xi_i^{T_l} \in \mathcal{D}_{T_l}} f(\theta_{cnn}, \xi_i^{T_l})$.
  $\rho_t^{vit} \leftarrow C\gamma^{-(t-1)}\hat{\rho}_{vit}$.
  $\rho_t^{cnn} \leftarrow C\gamma^{-(t-1)}\hat{\rho}_{cnn}$.
  ▷ *The ViT branch teaches the CNN branch*
  Update θ_{E_2} and θ_F by minimizing $\mathcal{L}_{vit\to cnn}^{unl}$ as in Equation (10).
  ▷ *The CNN branch teaches the ViT branch*
  Update θ_{E_1} and θ_F by minimizing $\mathcal{L}_{cnn\to vit}^{unl}$ as in Equation (11).
  # Minimizing Distribution Divergence
  ▷ *Use the minimax entropy strategy*
  Compute entropy losses for ViT and CNN branches:
  $H_{vit} \leftarrow$ Equation (14).
  $H_{cnn} \leftarrow$ Equation (15).
  Update θ_{E_1} by minimizing λH_{vit} on $\mathcal{D}_{T_u}$.
  Update θ_{E_2} by minimizing λH_{cnn} on $\mathcal{D}_{T_u}$.
  Update θ_F by maximizing $\lambda(H_{vit} + H_{cnn})$ on $\mathcal{D}_{T_u}$.
end for; *convergence or maximum training iterations are reached.*

remains the intra-domain discrepancy [30]. We are motivated by [29] to reduce the inter- and intra-domain discrepancies to improve the generalization of the ViT and CNN branches on the target domain. In particular, we define the entropy calculated by the ViT branch as follows:

$$H_{vit} = -\frac{1}{n_{T_u}} \sum_{i=1}^{n_{T_u}} \left[p_{vit}(y|x_i^{T_u}) \log(p_{vit}(y|x_i^{T_u})) \right], \tag{14}$$

where $p_{vit}(y|x_i^{T_u})$ is the probability of the i-th unlabeled target sample predicted by the ViT branch. Then, we train E_1 to minimize H_{vit}, while F is trained to maximize H_{vit}. By doing so, the ViT branch can simultaneously discriminate unlabeled target representations and minimize the distributions between source and target domains.

Similarly, the entropy calculated by the CNN branch is defined as follows:

$$H_{cnn} = -\frac{1}{n_{T_u}} \sum_{i=1}^{n_{T_u}} \left[p_{cnn}(y|x_i^{T_u}) \log(p_{cnn}(y|x_i^{T_u})) \right], \tag{15}$$

with $p_{cnn}(y|x_i^{T_u})$ is the prediction of the i-th unlabeled target sample to be categorized by the CNN branch.

The overall objective loss to update the parameters of ViT encoder E_1 can be written as follows:

$$\hat{\theta}_{E_1} = \underset{\theta_{E_1}}{\texttt{argmin}}\, \mathcal{L}_{vit}^{src} + \mathcal{L}_{cnn\to vit}^{unl} + \lambda H_{vit}, \tag{16}$$

where λ is a hyper-parameter that enables us to control a trade-off of the model's behavior influenced by the classification loss with labeled samples and the minimax entropy loss with unlabeled target samples. The CNN encoder E_2 is trained with the loss function as follows:

$$\hat{\theta}_{E_2} = \underset{\theta_{E_2}}{\texttt{argmin}}\, \mathcal{L}_{cnn}^{tar} + \mathcal{L}_{vit\to cnn}^{unl} + \lambda H_{cnn}. \tag{17}$$

The cost function used to train the shared classifier F is computed as follows:

$$\hat{\theta}_F = \underset{\theta_F}{\texttt{argmin}}\, \mathcal{L}_{vit}^{src} + \mathcal{L}_{cnn}^{tar} + \mathcal{L}_{D^2CR} - \lambda(H_{vit} + H_{cnn}). \tag{18}$$

The training procedure of the proposed dual dynamic consistency regularization is summarized in Algorithm 1.

IV. EXPERIMENTS

In this section, we first describe the semi-supervised domain adaptation benchmark datasets. Then, we provide the implementation settings in detail. Finally, we introduce the evaluation metrics used to assess the effectiveness of our method.

A. EXPERIMENT SETUP

1) DATASETS

We evaluated the proposed D^{2}CR method on two challenging SSDA benchmark datasets, including *DomainNet* [42],

TABLE 3. Statistics 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	
DomainNet	Real	175,327	345
	Clipart	48,837	
	Painting	75,759	
	Sketch	70,386	
	Infograph	53,201	
	Quickdraw	172,500	

Office-Home [41], and *Office-31* [40]. Following previous SSDA works [1], [9], [29], [30], we selected 4 domains in *DomainNet*, including *Real* (rel), *Painting* (pnt), *Clipart* (clp), and *Sketch* (skt) with 126 classes in each domain. *Office-Home* is a challenging SSDA benchmark, which contains 4 domains: *Art* (A), *Clipart* (C), *Product* (P), and *Real* (R); each domain has 65 categories. We implemented all experiments on these datasets under 1-shot and 3-shot settings. *Office-31* is a small dataset that includes 3 domains *DSLR* (D), *Webcam* (W), and *Amazon* (A) with 31 classes. The details of these datasets are described in Table 3.

2) IMPLEMENTATION DETAILS

We used the PyTorch [48] framework running on a single NVIDIA RTX 4090 GPU for all experiments. We selected ViT [34] and ResNet34 [45] as the encoders for the ViT and CNN branches, which were pre-trained on the ImageNet-1K [43] dataset, respectively. We followed the settings of the recent SSDA works [1], [9], [29], [30] for our implementation environments. For the classifier, we followed [29], in which the classifier consists of two fully connected layers followed by the `softmax` function. ViT and CNN branches used the same optimizer, Stochastic Gradient Descent (SGD)—with a momentum of 0.9 and weight decay of 0.0005. The ViT and CNN branches shared the same batch size ($B = 24$), scheduler, weight-decay, and optimizer (SGD). However, the initial learning rates were 0.001 and 0.01 for ViT and ResNet34 encoders, respectively. [37] demonstrated that the dynamic threshold was not sensitive with γ in the certain range $\gamma = \{1.01, 1.1, 1.2, 1.3\}$. Therefore, we randomly chose $\gamma = 1.1$ for all our experiments. Similar to [29], the temperature in Equation (13) was set to 0.05.

B. EVALUATION METRICS

1) ACCURACY

The classification accuracy extracted by the ViT and CNN branches on $\mathcal{D}_{T_u}$ was defined as follows:

$$Acc_{vit} = \frac{1}{n_{T_u}} \sum_{i=1}^{n_{T_u}} \mathbb{1}(\texttt{argmax}(F(E_1(x_i^{T_u}))) = y_i^{T_u}), \quad (19)$$

$$Acc_{cnn} = \frac{1}{n_{T_u}} \sum_{i=1}^{n_{T_u}} \mathbb{1}(\texttt{argmax}(F(E_2(x_i^{T_u}))) = y_i^{T_u}), \quad (20)$$

respectively, where y^{T_u} only is used in the testing phase.

2) EXPECTATION GAP

Alongside the classification results, we also introduced the "*expectation gap*" metric to evaluate the effectiveness of the dual dynamic consistency regularization method in sharing knowledge between the ViT and CNN branches, which was defined as follows:

$$Gap(ViT; CNN) = |Acc_{vit} - Acc_{cnn}|. \quad (21)$$

V. COMPARISON RESULTS

We evaluated the proposed D^{2}CR method by comparing it with previous SoTA methods, including MME [29], APE [30], STar [39], PAC [17], MAP-F [16], S^{3}D [11], Con^{2}DA [33], CLDA [15], CDAC [1], UODA [32], ProML [26], MCL [10], DeCoTa [14], ECACL [13], ASDA [12], MVCL [9], SLA [31]. Noticeably, to ensure a fair evaluation compared to these SSDA methods, we utilized the output classification results obtained from the CNN branch, where ResNet34 was selected as the encoder. In contrast, the output classification results extracted by the ViT branch were only used for analysis. Therefore, our framework did not introduce any complexity in the testing phase compared with prior SSDA methods.

A. RESULTS ON DOMAINNET

As reported in Table 4, the classification accuracy of 14 DA tasks on *DomainNet* under 1-shot and 3-shot settings. Our D^{2}CR surpassed the previous SSDA approaches in all DA tasks, where ResNet34 was selected for the CNN encoder. The mean accuracy was significantly boosted to 80.3% and 82.6% for 1-shot and 3-shot settings, respectively. The average classification results of our method achieved a gain of 4.1% and 4.5% compared to the second-best MVCL [9] under 1-shot and 3-shot settings, respectively.

B. RESULTS ON OFFICE-HOME

We listed the comparison results on *Office-Home* in Table 5. As shown in this table, the proposed D^{2}CR method achieved the highest classification performance in all domain adaptation tasks extracted by ResNet34, with 76.6% and 79.4% in averaging accuracy under 1-shot and 3-shot settings, respectively. The proposed method also improved by 3.7% and 3.1% on the average classification results compared to

TABLE 4. Accuracy (%) on *DomainNet* under 1-shot and 3-shot settings. (*To ensure fairness with prior studies, we utilize CNN-ResNet34 branch results for comparison, and results extracted by the ViT branch are only used for analysis.*)

Method	rel→clp		rel→pnt		pnt→clp		clp→skt		skt→pnt		rel→skt		pnt→rel		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
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
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
Con^{2}DA	71.3	74.2	71.8	72.1	71.1	75.0	60.0	65.7	63.5	67.1	65.2	67.1	75.7	78.6	68.4	71.4
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	67.8	71.7
S^{3}D	73.3	75.9	68.9	72.1	73.4	75.1	60.8	64.4	68.2	70.0	65.1	66.7	79.5	80.3	69.9	72.1
STar	74.1	77.1	71.3	73.2	71.0	75.8	63.5	67.8	66.1	69.2	64.1	67.9	80.0	81.2	70.0	73.2
PAC	74.9	78.6	73.0	74.3	72.6	76.0	65.8	69.6	67.9	69.4	68.7	70.2	76.7	79.3	71.4	73.9
MAP-F	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
CLDA	76.1	77.7	75.1	75.7	71.0	76.4	63.7	69.7	70.2	73.7	67.1	71.1	80.1	82.9	71.9	75.3
DECOTA	79.1	80.4	74.9	75.2	76.9	78.7	65.1	68.6	72.0	72.7	69.7	71.9	79.6	81.5	73.9	75.6
CDAC	77.4	79.6	74.2	75.1	75.5	79.3	67.6	69.9	71.0	73.4	69.2	72.5	80.4	81.9	73.6	76.0
ASDA	77.0	79.4	75.4	76.7	75.5	78.3	66.5	70.2	72.1	74.2	70.9	72.1	79.7	82.3	73.9	76.2
ECACL	75.3	79.0	74.1	77.3	75.3	79.4	65.0	70.6	72.1	74.6	68.1	71.6	79.7	82.4	72.8	76.4
MCL	77.4	79.4	74.6	76.3	75.5	78.8	66.4	70.9	74.0	74.7	70.7	72.3	82.0	83.3	74.4	76.5
CDAC+SLA	79.8	81.6	75.6	76.0	77.4	80.3	68.1	71.3	71.7	73.5	71.7	73.5	80.4	82.5	75.0	77.0
ProML	78.5	80.2	75.4	76.5	77.8	78.9	70.2	72.0	74.1	75.4	72.4	73.5	84.0	84.8	76.1	77.4
MVCL	78.8	79.8	76.0	77.4	78.0	80.3	70.8	73.0	75.1	76.7	72.4	74.4	82.4	85.1	76.2	78.1
D^{2}CR (CNN)	**82.2**	**83.7**	**80.1**	**83.2**	**84.9**	**85.5**	**71.7**	**76.8**	**81.0**	**81.3**	**74.9**	**78.8**	**87.3**	**88.6**	**80.3**	**82.6**
D^{2}CR (ViT)	83.9	84.3	83.3	84.6	85.2	85.8	72.7	78.0	82.1	82.6	76.4	80.1	89.5	90.2	81.9	83.7

TABLE 5. Accuracy (%) on *Office-Home* under 1-shot and 3-shot settings. (*To ensure fairness with prior studies, we utilize CNN-ResNet34 branch results for comparison, and results extracted by the ViT branch are only used for analysis.*)

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Mean
1-shot													
APE	53.9	76.1	75.2	63.6	69.8	72.3	63.6	58.3	78.6	72.5	60.7	81.6	68.9
DECOTA	42.1	68.5	72.6	60.3	70.4	70.7	60.0	48.8	76.9	71.3	56.0	79.4	64.8
MME	59.6	75.5	77.8	65.7	74.5	74.8	64.7	57.4	79.2	71.2	61.9	82.8	70.4
MME+SLA	62.1	76.3	78.6	67.5	77.1	75.1	66.7	59.9	80.0	72.9	64.1	83.8	72.0
CDAC	61.2	75.9	78.5	64.5	75.1	75.3	64.6	59.3	80.0	72.7	61.9	83.1	71.0
CDAC+SLA	63.0	78.0	79.2	66.9	77.6	77.0	67.3	61.8	80.5	72.7	66.1	84.6	72.9
D^{2}CR (CNN)	**66.2**	**80.8**	**81.5**	**73.9**	**84.3**	**80.4**	**75.2**	**63.8**	**81.2**	**78.7**	**67.1**	**85.5**	**76.6**
D^{2}CR(ViT)	71.5	86.7	81.3	74.0	89.2	88.0	81.8	70.8	90.5	84.8	70.9	91.1	81.7
3-shot													
APE	63.9	81.1	80.2	66.6	79.9	76.8	66.1	65.2	82.0	73.4	66.4	86.2	74.0
DECOTA	64.0	81.8	80.5	68.0	83.2	79.0	69.9	68.0	82.1	74.0	70.4	87.7	75.7
MME	63.6	79.0	79.7	67.2	79.3	76.6	65.5	64.6	80.1	71.3	64.6	85.5	73.1
MME+SLA	65.9	81.1	80.5	69.2	81.9	79.4	69.7	67.4	81.9	74.7	68.4	87.4	75.6
CDAC	65.9	80.3	80.6	67.4	81.4	80.2	67.5	67.0	81.9	72.2	67.8	85.6	74.8
CDAC+SLA	67.3	82.6	81.4	69.2	82.1	80.1	70.1	69.3	82.5	73.9	70.1	87.1	76.3
D^{2}CR (CNN)	**69.1**	**84.8**	**85.3**	**76.8**	**86.6**	**83.2**	**76.1**	**69.7**	**83.0**	**79.3**	**71.0**	**87.4**	**79.4**
D^{2}CR (ViT)	74.9	88.9	85.2	82.4	91.2	90.3	83.6	76.2	90.5	85.2	74.7	91.8	84.6

the second-best CDAC+SLA [31] with 1-shot and 3-shot settings, respectively.

C. RESULTS ON OFFICE-31

We report the comparison with available baseline results on *Office-31* in Table 6, using AlexNet backbone as the encoder of the CNN branch. Following [29], two adaptation scenarios were compared (*Webcam* to *Amazon*, *DSLR* to *Amazon*). Our D^{2}CR approach consistently outperforms the compared methods. The average classification results of our method achieved a gain of 15.0% and 11.8% compared to the second-best CLDA [15] under 1-shot and 3-shot settings, respectively.

VI. ABLATION STUDIES AND ANALYSES

This section comprises a comparative analysis of the proposed D^{2}CR method. We employed extensive experiments to evaluate the impact of each component in our approach. Besides, we provided results to demonstrate the effectiveness

TABLE 6. Accuracy (%) on *Office-31* under 1-shot and 3-shot settings. We utilize AlexNet as an encoder for the CNN branch.

Method	W→A		D→A		Mean	
	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
PAC	53.6	65.1	54.7	66.3	54.2	65.7
MME	57.2	67.3	55.8	67.8	56.5	67.6
APE	-	67.6	-	69.0	-	68.3
BiAT	57.9	68.2	54.6	68.5	56.3	68.4
STar	59.8	69.1	56.8	69.0	58.3	69.1
MVCL	56.7	69.0	59.3	69.1	58.0	69.1
Con^{2}DA	58.3	69.8	56.2	69.7	57.3	69.8
CDAC	63.4	70.1	62.8	70.0	63.1	70.0
CLDA	64.6	70.5	62.7	72.5	63.6	71.5
D^{2}CR (CNN)	**78.5**	**83.5**	**78.6**	**83.1**	**78.6**	**83.3**
D^{2}CR (ViT)	79.1	84.8	79.8	84.2	79.5	84.5

of the dynamic threshold algorithm and the proposed hybrid model.

A. EFFECTIVENESS OF EACH OBJECTIVE LOSS

We provided three scenarios to investigate the effectiveness of each loss in the proposed method. The experiments were conducted on *DomainNet* under the 3-shot setting, including $\mathcal{L}_{vit}^{src}+\mathcal{L}_{cnn}^{tar}$ for (*scenario 1*) under the supervised learning process, $\mathcal{L}_{vit\to cnn}^{unl}+\mathcal{L}_{cnn\to vit}^{unl}$ for D^{2}CR (*scenario 2*), and $H_{vit}+H_{cnn}$ for minimizing the distribution divergences (*scenario 3*). We implemented these scenarios on *DomainNet* under the 3-shot setting. As shown in Table 7, the average classification results of the ViT and CNN branches were 72.3% and 50.9%, respectively. The gap in the classification accuracy of these branches was significant by 21.4%. However, when we added the D^{2}CR strategy, the classification performance of ViT and CNN extensively increased up to 10.6% and 31.5%, respectively, and their gap dropped by 0.5%. The results of *scenario 2* indicated the success of D^{2}CR, where ViT and CNN branches mutually exchanged their knowledge to improve the generalization on the target data. Moreover, these results demonstrated that the proposed D^{2}CR method effectively alleviated the bias learning in SSDA when the ViT and CNN branches could provide a similar classification. Finally, the classification accuracy of ViT and CNN branches slightly increased by 0.8% and 0.2% on average, respectively, when the inter- and intra-domain discrepancies were reduced using the minimax entropy strategy in *scenario 3*.

B. EFFECTIVENESS OF DUAL DYNAMIC CONSISTENCY REGULARIZATION

Furthermore, as shown in Table 8, we could easily observe the effectiveness of the proposed D^{2}CR learning via two important aspects. Firstly, even though the ViT branch was trained on the unchanged labeled source data, the classification accuracy in this branch still increased following the rise of labeled target data used for the CNN branch. Secondly, the gap between the ViT and CNN branches, as computed in Equation (21), was reduced when we increased the number of labeled target samples. The above

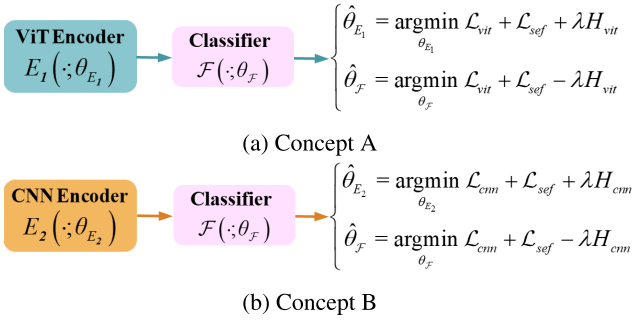


FIGURE 3. Visualization of different methodologies in designing frameworks. (a) and (b) adopt the single branch approach, which consists of a single classifier but utilizing distinct encoder models such as ViT and CNN, respectively.

observations demonstrated that the ViT and CNN branches of the proposed method were successful in exchanging their knowledge to improve cross-domain generalization.

C. EFFECTIVENESS OF THE DYNAMIC THRESHOLD ALGORITHM

DeCoTa [14], ECACL [13], and Con^{2}DA [33] set the fixed thresholds with $\tau = 0.5$, $\tau = 0.8$, and $\tau = 0.9$ for their implementations, respectively. Besides, CDAC [1], MVCL [9], and ASDA [12] choose $\tau = 0.95$ for all experiments. Therefore, we selected $\tau = 0.5, 0.8, 0.9$, and 0.95 to conduct experiments to demonstrate the effectiveness of D^{2}CR using the dynamic threshold. As shown in Table 9, the comparison classification results of the proposed method were conducted with fixed and dynamic threshold algorithms over three domain adaptation tasks on *DomainNet*. The classification accuracy extracted by the dynamic threshold was significantly higher than that extracted by the fixed threshold in all tasks, with 85.6% on ViT and 84.4% on CNN.

D. EFFECTIVENESS OF HYBRID MODEL-BASED

To demonstrate the effectiveness of the proposed hybrid model-based framework, we made a full comparison with two conventional architectures sharing the same conceptual design with an encoder and a single classifier, as visualized in Figure 3. However, in *concept A*, the ViT model [34] is selected for an encoder, while the CNN (ResNet34) [45] is used for *concept B*. As indicated in Table 10, the average classification accuracy of the ViT branch in the hybrid model further improved by 4.1% compared to the average classification derived from *concept A*. Likewise, the averaged outcomes of the CNN branch in the hybrid model demonstrated a 7.0% increase compared to the average classification extracted by *concept B*.

VII. VISUALIZATION

We used t-SNE [47] to visualize the representations of the target data in the challenging DA scenario "rel→skt" on *DomainNet* with the 3-shot setting. The visualizations depicted in Figure 4 were concordant with the ablation study found from the ablation study, as presented in Table 7.

TABLE 7. Ablation study on *DomainNet* to evaluate the impact of each object loss in our method under the 3-shot setting.

Scenario	Supervised	D^{2}CR	Entropy	rel→clp		rel→pnt		pnt→clp		clp→skt		skt→pnt		rel→skt		pnt→rel		Mean	
	$\mathcal{L}_{vit}^{src} + \mathcal{L}_{cnn}^{tar}$	$\mathcal{L}_{vit\to cnn}^{unl} + \mathcal{L}_{cnn\to vit}^{unl}$	$H_{vit} + H_{cnn}$	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN
1	✓			68.5	49.4	76.9	51.6	70.7	49.9	68.6	43.1	74.5	50.6	60.1	42.0	87.1	69.5	72.3	50.9
2	✓	✓		83.4	83.3	83.9	83.1	85.3	85.1	76.7	76.4	81.7	80.9	79.2	78.9	89.8	88.8	82.9	82.4
3	✓	✓	✓	84.3	83.7	84.6	83.2	85.8	85.5	78.0	76.8	82.6	81.3	80.1	78.8	90.2	88.6	83.7	82.6

TABLE 8. Ablation study on *DomainNet* to evaluate the effectiveness of the D^{2}CR learning.

Setting	rel→clp			rel→pnt			pnt→clp			clp→skt			skt→pnt			rel→skt			pnt→rel			Mean		
	ViT	CNN	Gap	ViT	CNN	Gap	ViT	CNN	Gap	ViT	CNN	Gap	ViT	CNN	Gap	ViT	CNN	Gap	ViT	CNN	Gap	ViT	CNN	Gap
1-shot	83.9	82.2	1.7	83.3	80.1	3.2	85.2	84.9	0.3	72.7	71.7	1.0	80.2	78.5	1.7	76.4	74.9	1.5	89.5	87.3	2.2	81.6	79.9	1.7
3-shot	84.3	83.7	0.6	84.6	83.2	1.4	85.8	85.5	0.3	78.0	76.8	1.2	82.6	81.3	1.3	80.1	78.8	1.3	90.2	88.6	1.6	83.7	82.6	1.1
5-shot	87.3	87.0	0.3	84.7	83.4	1.3	87.1	87.0	0.1	79.7	79.1	0.6	83.8	82.6	1.2	80.2	79.0	1.2	90.6	88.9	1.7	84.8	83.9	0.9
10-shot	87.9	87.6	0.3	86.0	84.6	1.4	88.6	88.6	0.0	81.2	80.6	0.6	85.5	84.6	0.9	80.8	80.3	0.5	91.7	90.2	1.5	86.0	85.2	0.7
20-shot	88.7	88.6	0.1	87.1	86.1	1.0	88.8	88.8	0.0	83.3	82.7	0.6	86.7	85.9	0.8	81.9	81.5	0.4	92.1	90.7	1.4	86.9	86.3	0.6

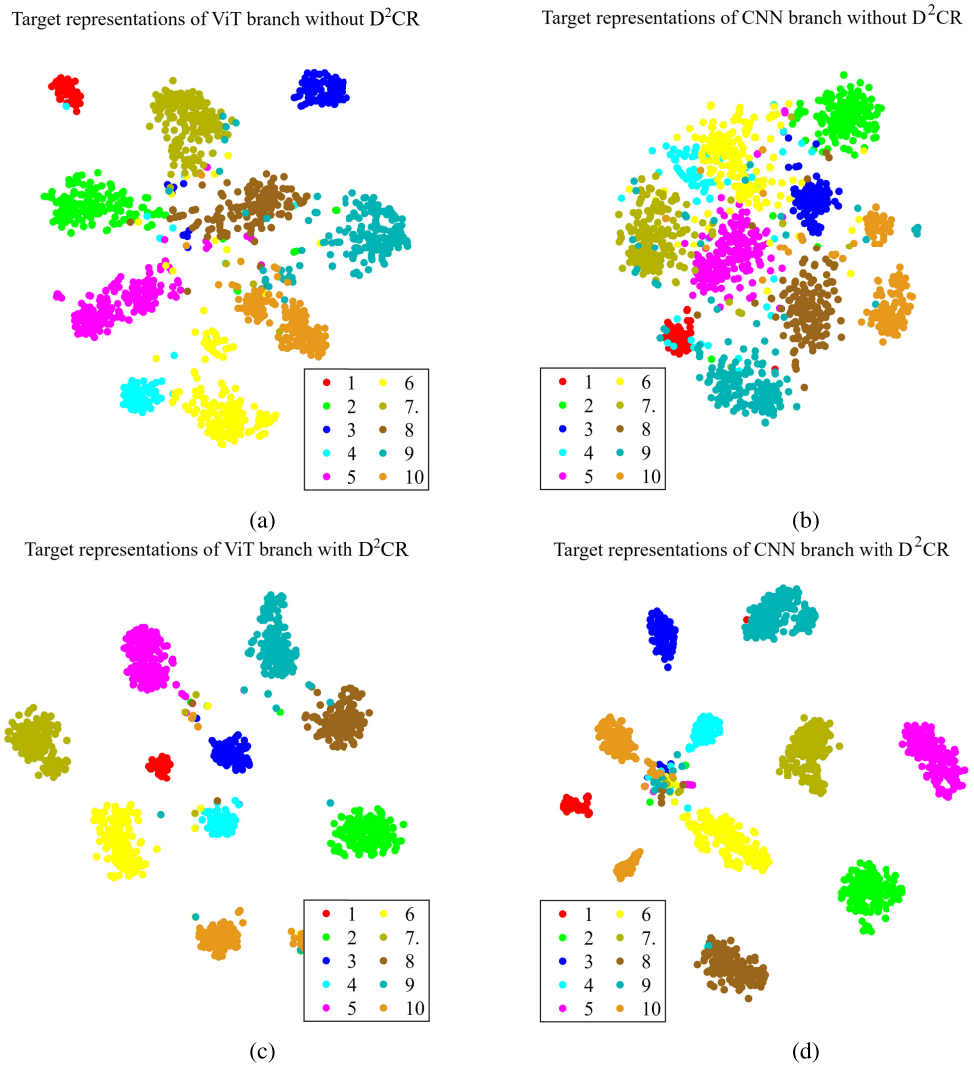


FIGURE 4. We visualized the representations of 10 difference classes in the target sketch domain under the "rel→skt" task on *DomainNet* using t-SNE [47]. The target representations obtained by the ViT and CNN branches with D^{2}CR were well discriminative compared to those obtained without utilizing D^{2}CR.

In particular, when D^{2}CR was not employed, the ViT and CNN branches failed to complement each other, resulting in a lack of information. Consequently, they yielded low classification accuracy as in *scenario 1*, along with poor

TABLE 9. Ablation study on *DomainNet* to investigate the efficacy of the dynamic and fixed thresholds with 1-shot.

Threshold		pnt→clp		skt→pnt		pnt→rel		Mean	
		ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN
Fixed	0.5	82.6	82.3	80.0	79.5	86.5	84.8	83.0	82.2
	0.8	82.7	82.7	73.6	78.7	87.2	84.8	81.2	82.1
	0.9	83.2	82.9	79.2	76.8	86.9	84.1	83.1	81.3
	0.95	81.6	80.5	80.2	76.5	88.1	83.9	83.3	80.3
Dynamic		**85.2**	**84.9**	**82.1**	**81.0**	**89.5**	**87.3**	**85.6**	**84.4**

TABLE 10. Ablation study (%) on *DomainNet* to analyze the performance of various frameworks under the 3-shot settings.

Method	rel→clp	rel→pnt	pnt→clp	clp→skt	skt→pnt	rel→skt	pnt→rel	Mean
Concept A (ViT)	76.1	79.5	77.6	72.9	80.8	78.8	91.6	79.6
Concept B (CNN)	76.8	76.8	76.9	72.1	71.2	71.4	84.3	75.6
D^2CR (CNN)	83.7	83.2	85.5	76.8	81.3	78.8	88.6	82.6
D^2CR (ViT)	84.3	84.6	85.8	78.0	82.6	80.1	90.2	83.7

representations, as shown in Figures 4a and 4b. In contrast, the target representations extracted by the ViT and CNN branches were well clustered within the same class and clearly discriminated among the different categories under the support of D^2CR, as shown in Figures 4c and 4d. These visualization results were robust evidence to demonstrate the effectiveness of the proposed D^2CR learning.

VIII. CONCLUSION

In this work, we successfully exploited the benefits of the ViT and CNN encoders to enjoy global and local representations for effectively alleviating the bias in learning issues in SSDA. Besides, we leveraged the dynamic threshold to propose the dual dynamic consistency regularization (D^2CR), enabling the ViT and CNN branches to exchange their knowledge to support each other as well as address their weaknesses effectively. Our method achieved outstanding classification results. The ViT model works as the auxiliary model to support the CNN model during training. Then, we only employ the CNN model in the testing time. Therefore, the proposed method did not require any additional complexity compared with previous works in the testing phase. To the best of our knowledge, the proposed method is the first SSDA approach to achieve outstanding results through the dynamic threshold algorithm. In future work, we intend to test our proposed method in diverse settings, including unsupervised domain adaptation and open-set domain adaptation setting.

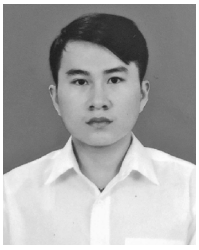
REFERENCES

[1] J. Li, G. Li, Y. Shi, and Y. Yu, "Cross-domain adaptive clustering for semi-supervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 2505–2514.
[2] 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, p. 114.
[3] P. J. Lu, C.-Y. Jui, and J.-H. Chuang, "FedDAD: Federated domain adaptation for object detection," *IEEE Access*, vol. 11, pp. 51320–51330, 2023.
[4] Y. Yu, X. Xu, X. Hu, and P.-A. Heng, "DALocNet: Improving localization accuracy for domain adaptive object detection," *IEEE Access*, vol. 7, pp. 63155–63163, 2019.
[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] C. Ruan, W. Wang, H. Hu, and D. Chen, "Category-level adversaries for semantic domain adaptation," *IEEE Access*, vol. 7, pp. 83198–83208, 2019.
[7] X. Yu, Y. Chu, F. Jiang, Y. Guo, and D. Gong, "SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features," *Knowl.-Based Syst.*, vol. 141, pp. 80–91, Feb. 2018.
[8] X. Yu, F. Jiang, J. Du, and D. Gong, "A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains," *Pattern Recognit.*, vol. 94, pp. 96–109, Oct. 2019.
[9] 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.
[10] Z. Yan, Y. Wu, G. Li, Y. Qin, X. Han, and S. Cui, "Multi-level consistency learning for semi-supervised domain adaptation," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 1–7.
[11] J. Yoon, D. Kang, and M. Cho, "Semi-supervised domain adaptation via sample-to-sample self-distillation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2022, pp. 1978–1987.
[12] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, and Y. Fu, "Semi-supervised domain adaptive structure learning," *IEEE Trans. Image Process.*, vol. 31, pp. 7179–7190, 2022.
[13] K. Li, C. Liu, H. Zhao, Y. Zhang, and Y. Fu, "ECACL: A holistic framework for semi-supervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8558–8567.
[14] L. Yang, Y. Wang, M. Gao, A. Shrivastava, K. Q. Weinberger, W.-L. Chao, and S.-N. Lim, "Deep co-training with task decomposition for semi-supervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8886–8896.
[15] A. Singh, "CLDA: Contrastive learning for semi-supervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 5089–5101.
[16] 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.
[17] V. Saligrama, K. Saenko, and S. Mishra, "Surprisingly simple semi-supervised domain adaptation with pretraining and consistency," in *Proc. BMVC*, 2021, pp. 1–20.
[18] P. Jiang, A. Wu, Y. Han, Y. Shao, M. Qi, and B. Li, "Bidirectional adversarial training for semi-supervised domain adaptation," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 934–940.
[19] L. Abdi and S. Hashemi, "Unsupervised domain adaptation based on correlation maximization," *IEEE Access*, vol. 9, pp. 127054–127067, 2021.
[20] S. Zang, Y. Cheng, X. Wang, Q. Yu, and G.-S. Xie, "Cross domain mean approximation for unsupervised domain adaptation," *IEEE Access*, vol. 8, pp. 139052–139069, 2020.
[21] T. Fu and Y. Li, "Unsupervised domain adaptation based on pseudo-label confidence," *IEEE Access*, vol. 9, pp. 87049–87057, 2021.
[22] Y. Zhang, N. Wang, S. Cai, and L. Song, "Unsupervised domain adaptation by mapped correlation alignment," *IEEE Access*, vol. 6, pp. 44698–44706, 2018.
[23] X. Jia and F. Sun, "Unsupervised deep domain adaptation based on weighted adversarial network," *IEEE Access*, vol. 8, pp. 64020–64027, 2020.
[24] 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.
[25] 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.
[26] X. Huang, C. Zhu, and W. Chen, "Semi-supervised domain adaptation via prototype-based multi-level learning," in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, Aug. 2023, pp. 884–892.
[27] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "MixMatch: A holistic approach to semi-supervised learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5050–5060.

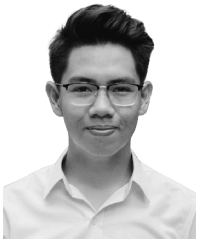
[28] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C. L. Li, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, p. 51.
[29] K. Saito, D. Kim, S. Sclaroff, T. 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.
[30] 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.*, 2020, pp. 591–607.
[31] Y.-C. Yu and H.-T. Lin, "Semi-supervised domain adaptation with source label adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 24100–24109.
[32] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, and Y. Fu, "Contradictory structure learning for semi-supervised domain adaptation," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2021, pp. 3723–3732.
[33] M. Prez-Carrasco, P. Protopapas, and G. Cabrera-Vives, "Con^{2}DA: Simplifying semi-supervised domain adaptation by learning consistent and contrastive feature representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–11.
[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.* 2021, p. 122.
[35] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your ViT? Data, augmentation, and regularization in vision transformers," 2021, *arXiv:2106.10270*.
[36] O. S. Kayhan and J. C. van Gemert, "On translation invariance in CNNs: Convolutional layers can exploit absolute spatial location," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14274–14285.
[37] Y. Xu, L. Shang, J. Ye, Q. Qian, Y. F. Li, B. Sun, H. Li, and R. Jin, "Dash: Semi-supervised learning with dynamic thresholding," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11525–11536.
[38] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn. (ICML)*, vol. 3, 2013, p. 896.
[39] A. Singh, N. Doraiswamy, S. Takamuku, M. Bhalerao, T. Dutta, S. Biswas, A. Chepuri, B. Vengatesan, and N. Natori, "Improving semi-supervised domain adaptation using effective target selection and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2703–2712.
[40] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*, 2010, pp. 213–226.
[41] 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.
[42] 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.
[43] 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.
[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, Dec. 2012, pp. 1097–1105.
[45] 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.
[46] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop (CVPRW)*, 2020, pp. 702–703.
[47] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[48] 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. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–4.

BA HUNG NGO received the B.S. degree in control engineering and automation from Hanoi University of Mining and Geology, Hanoi, Vietnam, in 2014, the M.S. degree in control engineering and automation from Hanoi University of Science and Technology, in 2016, and the Ph.D. degree from Dongguk University, Seoul, Republic of Korea.

He is currently a Postdoctoral Research Fellow with the Graduate School of Data Science, Chonnam National University. He has authored or coauthored multiple papers in top conferences, such as ICCV, WACV, and BMVC. His current research interests include computer vision and deep learning, especially deep transfer learning, domain adaptation, graph convolution networks (GCNs), graph learning, and knowledge distillation. He serves as a Reviewer for CVPR 2024.



BA THINH LAM received the B.S. degree from the University of Science, Vietnam National University Ho Chi Minh, in 2021. His research interests include computer vision, semi-supervised learning, and domain adaptation.



THANH HUY NGUYEN received the B.Sc. degree from the Mathematics Department, Ho Chi Minh University of Education, Vietnam, in 2022. His research interests include computer vision, medical image analysis, and semi-supervised learning.

QUANG VINH DINH received the B.S. degree in computer science from Nong Lam University, Ho Chi Minh City, Vietnam, in 2008, and the M.S. and Ph.D. degrees in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2013 and 2016, respectively. From 2016 to 2017, he was a Postgraduate Researcher with Sungkyunkwan University. From 2017 to 2020, he was also a Postgraduate Researcher with Gwangju Institute of Science and Technology. In 2020, he joined Vietnamese–German University, where he is a Lecturer with the School of Electrical Engineering and Computer Science. His current research interests include computer vision and deep learning.

TAE JONG CHOI received the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, South Korea. He is an Assistant Professor with Chonnam National University, South Korea. In 2014, he was a Visiting Scholar with the National Institute of Advanced Industrial Science and Technology (AIST), Japan; and New York University, USA, in 2019. From 2020 to 2022, he was an Assistant Professor with Kyungil University, South Korea. His research is focused on applying evolutionary computation, machine learning, and artificial intelligence to various fields.