Learning CNN on ViT: A Hybrid Model to Explicitly Class-specific Boundaries for Domain Adaptation

Ba Hung Ngo[1,*], Nhat-Tuong Do-Tran[2,*], Tuan-Ngoc Nguyen[3], Hae-Gon Jeon[4], Tae Jong Choi[1,†]
[1]Graduate School of Data Science, Chonnam National University, South Korea
[2]Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
[3]Digital Transformation Center, FPT Telecom, VietNam, [4]AI Graduate School, GIST, South Korea
ngohung@chonnam.ac.kr tuongdotn.cs11@nycu.edu.tw tuannn55@fpt.com haegonj@gist.ac.kr ctj17@jnu.ac.kr

Abstract

*Most domain adaptation (DA) methods are based on either a convolutional neural networks (CNNs) or a vision transformers (ViTs). They align the distribution differences between domains as encoders without considering their unique characteristics. For instance, ViT excels in accuracy due to its superior ability to capture global representations, while CNN has an advantage in capturing local representations. This fact has led us to design a hybrid method to fully take advantage of both ViT and CNN, called **E**xplicitly **C**lass-specific **B**oundaries (**ECB**). ECB learns CNN on ViT to combine their distinct strengths. In particular, we leverage ViT's properties to explicitly find class-specific decision boundaries by maximizing the discrepancy between the outputs of the two classifiers to detect target samples far from the source support. In contrast, the CNN encoder clusters target features based on the previously defined class-specific boundaries by minimizing the discrepancy between the probabilities of the two classifiers. Finally, ViT and CNN mutually exchange knowledge to improve the quality of pseudo labels and reduce the knowledge discrepancies of these models. Compared to conventional DA methods, our ECB achieves superior performance, which verifies its effectiveness in this hybrid model. The project website can be found here.*

1. Introduction

Over the past few years, convolutional neural networks (CNNs) [13] have been the cornerstone of deep learning techniques in computer vision tasks. This progress is mainly attributed to a convolution layer, which efficiently captures local spatial hierarchies for robust image representations. This local feature extraction capability has enabled CNNs to achieve State-of-the-Art (SOTA) performance in a variety of vision tasks, from image classification to object detection. In spite of its powerful local feature extraction, CNNs are somewhat limited in capturing more global and comprehensive visual context. To overcome this limitation, vision transformer (ViT) [7] was introduced. ViT starts by dividing an image into patches, which are transformed into a sequence of tokens. Positional embeddings are incorporated into the tokens to retain the order of these patches. The model then uses transformer blocks to extract these tokens into features as image representations. Thanks to self-attention mechanisms in ViT, it is able to weigh the importance of different regions in an image irrespective of their spatial proximity, leading to comprehensive global representations with impressive accuracy. The advancements in ViT have led to a growing inclination in the machine learning realm over CNN-based approaches for a range of tasks.

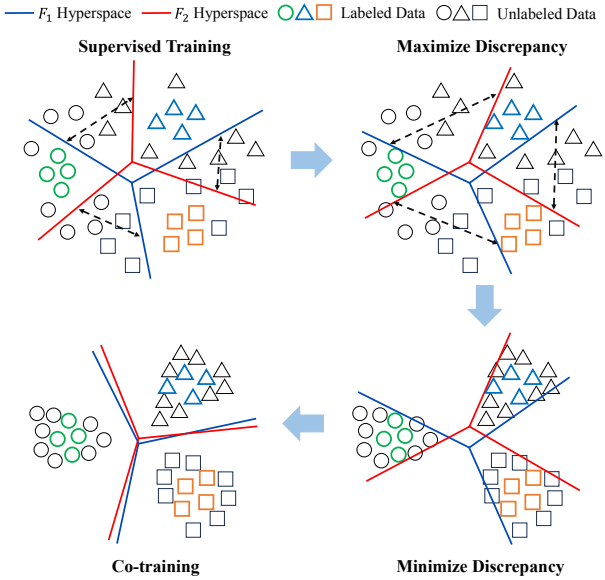


Figure 1. Illustration of the proposed hybrid network architecture that leverages the strengths of ViT and CNN models.

Instead of focusing solely on replacing CNN with ViT, our approach diverges from this trend. We believe that both

*Co-first author. †Corresponding author.

ViT and CNN architectures have their own strengths that can be harnessed when combined well. For instance, ViT has shown the capability to capture global representations and demonstrates robustness when trained on large datasets. Yet, because it is composed of multilayer perceptron (MLP) layers, ViT can overfit if the dataset is limited. On the other hand, CNNs perform well with relatively small datasets thanks to their spatial invariance and robustness in capturing local representations. Motivated by this, we propose a novel method that capitalizes on the distinct strengths of both architectures. As shown in Fig. 1, we exploit the superior accuracy of ViT in identifying more general class-specific boundaries by maximizing the discrepancy between the outputs of the two classifiers, enabling us to estimate the worst-case hyperspaces. Once these class-specific boundaries are defined, CNN minimizes the discrepancy by clustering target features closer to the source domain, aiming to minimize errors within the identified hyperspaces. However, the knowledge discrepancies between ViT and CNN still exist, so we applied additional co-training to bridge this gap while improving the quality of the pseudo labels. In the field of unsupervised domain adaptation (UDA), MCD [30] emphasizes the importance of maximizing the discrepancy between classifier outputs for target samples, which are far from the source domain's support, and then minimizing this discrepancy through a feature generator to align the target features closer to the source's support.

Although UDA has made significant advancements in domain adaptation (DA) tasks, the semi-supervised domain adaptation (SSDA) scenarios, as discussed in [11, 14–16, 28, 31, 34–36], are extensively employed to yield remarkable classification accuracy compared to the UDA setting [2, 4, 5, 19, 37]. 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 previous DA methods only take full advantage of the unique benefits of CNN as a feature extractor. Specifically, works in [12, 15, 21, 31] use a combination of a CNN encoder followed by an MLP classifier, but the decision boundary towards the source domain leads to data bias in DA. To address this bias, some previous works [27, 28, 37] introduce a multi-model by adding one more MLP classifier, which consists of a single CNN encoder and two MLP classifiers. Furthermore, the first DA approaches in [22, 34] use two CNN encoders and two MLP classifiers to boost the classification accuracy by leveraging a co-training strategy that ensures the consistency of unlabeled target data through knowledge exchange. However, these methodologies still follow the same rule, where the CNN model is selected as the encoder, and MLP is assigned as the classifier. Therefore, the capabilities of ViT in capturing global information remain unexploited. Inspired by the ideal, we make use of this multi-model architecture to build a new hybrid framework that leverages the advantages of ViT and CNN for DA settings. However, the proposed method does not require any additional complexity compared to previous works in the test phase. We summarize the contributions of this paper as follows:

- We introduce a hybrid model that can take advantage of ViT. Beyond simply replacing CNN with ViT, we can drive the feature space of ViT to CNN.
- Our approach demonstrates the successful integration of ViT and CNN, making a synergy with these two powerful frameworks.
- The proposed method outperforms the prior works to achieve SOTA performances on DA benchmark datasets.

2. Related Works

Co-training. In the realm of semi-supervised learning (SSL), co-training is a scheme to improve the robustness, first proposed by [1], and harnesses data from dual views, enabling two models to iteratively 'teach' each other. During this process, each model alternately makes predictions on unlabeled data, with the most confident predictions used to augment the training set of the other model. This mutual teaching strategy can significantly improve the performance of both models, particularly when labeled data is scarce. In the DA context, FixBi [22], DECOTA [34], and MVCL [23] offer innovative co-training strategies. Notably, FixBi and DECOTA utilize two distinct branches, each including a feature extractor and a classifier. They utilize MIXUP augmentation to reduce the gap in multiple intermediate domains between the source and target domain during co-training. However, they miss the potential benefits of strong augmentation. In contrast, while MVCL employs both weak and strong augmentation to bolster its co-training approach, it relies on a single CNN-based encoder and two classifiers to produce two views. This limits the representation of unlabeled target data, depriving its comprehensive global information. To address these limitations, we employ the ViT in conjunction with a CNN-based encoder. This combination generates two representational views that encapsulate both local and global information. Additionally, we integrate both weak and strong augmentations for unlabeled target samples, which enhances the interactivity and effectiveness of our co-training strategy.

Domain Adaptation Framework. In the realm of DA, various frameworks have been presented. Early methods such as MME [31], APE [15], and SENTRY [26] adopt the conventional approach of constructing deep learning frameworks that have a feature extractor coupled with a classifier. However, source and target domains share the same decision boundary, which leads to data bias toward the source domain. To alleviate data bias, MDD [37] and UODA [27] introduce two distinct classifiers. They demonstrate that using a dual-classifier setup can improve the performance

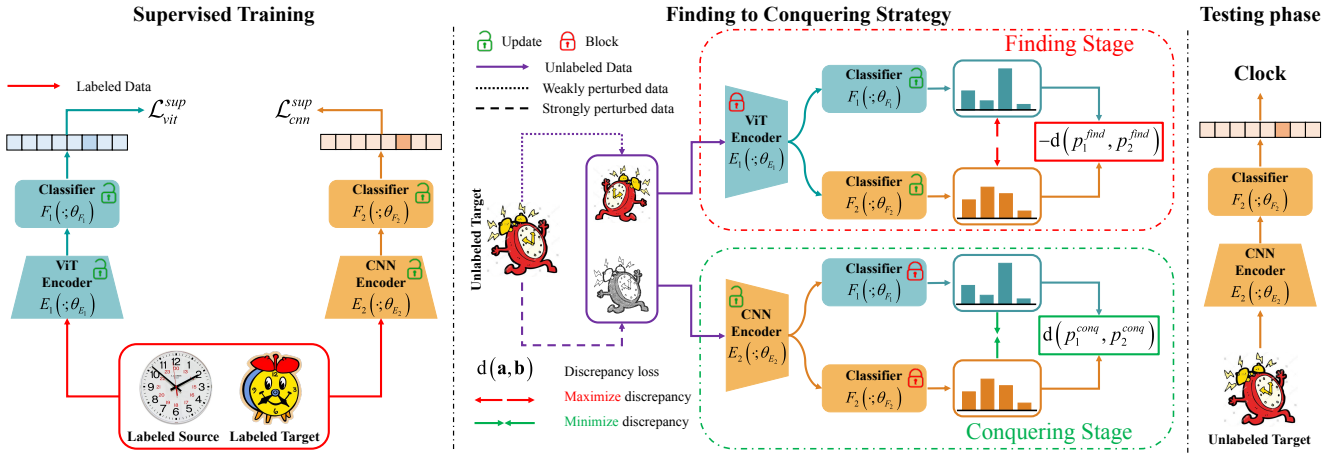


Figure 2. Illustration of a hybrid network with the proposed Finding to Conquering strategy. We use ViT to build E_1 that drives two classifiers F_1 and F_2 to expand class-specific boundaries comprehensively. Besides, we select CNN for the second encoder E_2 to cluster target features based on the boundaries identified by ViT. These encoders all use two classifiers F_1, F_2.

of classification tasks. These methods use a single feature extractor to train on both labeled source and target datasets. However, the source domain is more reliable than the target domain thanks to labeled source samples, which can lead the feature extractor to overly focus on source data representations. This results in a learning bias toward the source domain, accumulating errors during the training process. To solve this problem, DECOTA [34] proposes a novel architecture by decomposing two distinct branches, UDA and SSL, each consisting of a feature extractor and a single classifier. Specifically, the UDA branch is trained on labeled source data and unlabeled target data, while the SSL branch is trained on labeled and unlabeled target data. Thanks to the SSL branch, focusing solely on the target domain facilitates the alleviation of learning bias. In addition, the UDA branch is capable of mitigating learning bias by leveraging extra information from the SSL branch via the co-training strategy. Despite the notable advancements in the field, DECOTA still follows the framework of using CNN as the feature extractor. Therefore, it is impossible to completely improve the quality of pseudo labels for unlabeled target samples and exploit learning space based on the properties of ViT. As a result, the capabilities of ViT in capturing global information still need to be explored. Instead of solely replacing CNN with ViT, we propose a hybrid model that combines their strengths of both architectures.

3. Methodology

In DA scenarios, we deal with the following data setting:

- **Labeled source domain**: Denoted as $\mathcal{D}_{\mathcal{S}} = \{(x_i^{\mathcal{S}}, y_i^{\mathcal{S}})\}_{i=1}^{\mathcal{N}_{\mathcal{S}}}$ includes $\mathcal{N}_{\mathcal{S}}$ richly labeled samples.
- **Labeled target domain**: Denoted as $\mathcal{D}_{\mathcal{T}_l} = \{(x_i^{\mathcal{T}_l}, y_i^{\mathcal{T}_l})\}_{i=1}^{\mathcal{N}_{\mathcal{T}_l}}$ includes $\mathcal{N}_{\mathcal{T}_l}$ limited labeled target samples. *Notably, $\mathcal{D}_{\mathcal{T}_l}$ is empty in UDA.*
- **Unlabeled target domain**: Denoted as $\mathcal{D}_{\mathcal{T}_u} = \{(x_i^{\mathcal{T}_u}, y_i^{\mathcal{T}_u})\}_{i=1}^{\mathcal{N}_{\mathcal{T}_u}}$ includes $\mathcal{N}_{\mathcal{T}_u}$ target samples that do not have corresponding labels during the training phase.

In this setup, the sample $x_i^{\mathcal{S}}$ and $x_i^{\mathcal{T}_l}$ from the source domain and the labeled target domain are associated with corresponding ground-truth labels $y_i^{\mathcal{S}}$ and $y_i^{\mathcal{T}_l}$, respectively. There is an assumption that the label $y^{\mathcal{S}}$, $y^{\mathcal{T}_l}$, and $y^{\mathcal{T}_u}$ all belong to the same label space with K classes. Notably, $y^{\mathcal{T}_u}$, which denotes labels for the unlabeled target data, are only used during the testing phase. Furthermore, the number of labeled target samples $\mathcal{N}_{\mathcal{T}_l}$ is much smaller than both $\mathcal{N}_{\mathcal{S}}$ and $\mathcal{N}_{\mathcal{T}_u}$. Moreover, we implement two varied stochastic data transformations: $Aug_w(\cdot)$ and $Aug_{str}(\cdot)$. The function $Aug_w(\cdot)$ is a weak augmentation method employing light perturbations such as random horizontal flipping and random cropping, whereas $Aug_{str}(\cdot)$ stands as a strong augmentation method, using RandAugment [3], which involves 14 transformation techniques. Specifically, both $Aug_w(\cdot)$ and $Aug_{str}(\cdot)$ are applied to the unlabeled set $\mathcal{D}_{\mathcal{T}_u}$, transforming a sample $x_i^{\mathcal{T}_u}$ to two versions $x_{i,w}^{\mathcal{T}_u}$ and $x_{i,str}^{\mathcal{T}_u}$, respectively. Subsequently, we train the model on the labeled set $\mathcal{D}_l = \mathcal{D}_{\mathcal{S}} \cup \mathcal{D}_{\mathcal{T}_l}$ and the unlabeled set $\mathcal{D}_{\mathcal{T}_u}$, and evaluate the trained model on $\mathcal{D}_{\mathcal{T}_u}$.

To improve the performance on $\mathcal{D}_{\mathcal{T}_u}$, we propose a hybrid model consisting of ViT and CNN branches. The ViT branch is made up of a ViT encoder $E_1(\cdot;\boldsymbol{\theta}_{E_1})$ and a classifier $F_1(\cdot;\boldsymbol{\theta}_{F_1})$, while the CNN branch includes a CNN encoder $E_2(\cdot;\boldsymbol{\theta}_{E_2})$ and a classifier $F_2(\cdot;\boldsymbol{\theta}_{F_2})$. Our strategy training proceeds in three steps:

1. **Supervised Training**: We train both ViT and CNN branches on $\mathcal{D}_l$ whose knowledge is adapted to the $\mathcal{D}_{\mathcal{T}_u}$ as illustrated in Fig. 2.
2. **Finding to Conquering**: In Fig. 2, we find class-specific boundaries based on fixed E_1 by maximizing discrepancy between F_1 and F_2. Subsequently, the E_2 clusters the

target features based on those class-specific boundaries by minimizing discrepancy.

3. **Co-training**: To exchange effectively knowledge between two branches on $\mathcal{D}_{\mathcal{T}_u}$, the ViT branch generates a pseudo label for weakly version $x_{i,w}^{\mathcal{T}_u}$ to teach the CNN branch with strongly version $x_{i,str}^{\mathcal{T}_u}$. Then, the reverse process is also applied as depicted in Fig. 3.

3.1. Supervised Training

In this phase, we employ the standard cross-entropy loss to train a model's two branches: **ViT branch** and **CNN branch**. In the ViT branch, we aim to minimize the empirical loss of labeled data, which is defined as follows:

$$\mathcal{L}_{vit}^{sup}(\boldsymbol{\theta}_{E_1}, \boldsymbol{\theta}_{F_1}) = \frac{1}{\mathcal{N}_l} \sum_{i=1}^{\mathcal{N}_l} H(y_i^l, p_1^l(x_i^l)), \tag{1}$$

where $H(\cdot)$ denotes the standard cross-entropy loss. When $p_1^l(x_i^l) = \sigma(F_1(E_1(x_i^l)))$, σ represents the `softmax` function, which transforms the logits into probabilities across K categories. Initially, ViT encoder E_1 maps the labeled sample x_i^l into a d-dimensional feature space. The classifier F_1 then categorizes the input feature space $E_1(x_i^l)$ into the logits. Finally, we minimize the cross-entropy loss between the predicted labeled probability by $p_1^l(x_i^l)$ and the provided ground-truth label y_i^l, ensuring accurate predictions on labeled data.

Similarly, the CNN branch is also trained using labeled data to minimize the empirical loss as follows:

$$\mathcal{L}_{cnn}^{sup}(\boldsymbol{\theta}_{E_2}, \boldsymbol{\theta}_{F_2}) = \frac{1}{\mathcal{N}_l} \sum_{i=1}^{\mathcal{N}_l} H(y_i^l, p_2^l(x_i^l)), \tag{2}$$

where $p_2^l(x_i^l) = \sigma(F_2(E_2(x_i^l)))$ is obtained by extracting features from the labeled sample x_i^l using the CNN encoder E_2. These features are then mapped to logits through the classifier F_2. Finally, the `softmax` function σ transforms these logits into probabilities across K categories. The main goal of this process is to minimize the cross-entropy loss to ensure a close alignment between the predicted probabilities $p_2^l(x_i^l)$ and the ground-truth labels y_i^l.

3.2. Finding to Conquering

Discrepancy Loss Definition. In this work, we use the absolute difference between the probabilistic outputs **a** and **b** from two distinct classifiers F_1 and F_2, respectively, for each class to guarantee that the discrepancy loss is always non-negative as follows:

$$d(\mathbf{a}, \mathbf{b}) = \frac{1}{K} \sum_{k=1}^{K} |a_k - b_k|, \tag{3}$$

where a_k and b_k correspond to the probability outputs of the two classifiers for the k-th class. Normalizing with the total number of categories assures that the discrepancy loss is scale-invariant and bounded between 0 and 1. The discrepancy loss serves as a measure of divergence between the probabilistic outputs of two classifiers, **a** and **b**. A discrepancy loss of zero indicates the perfect agreement between them across all classes. On the other hand, a higher discrepancy loss points to a more significant divergence in the two classifiers's predictions. This loss is beneficial in scenarios where we want to expand class-specific boundaries.

Finding Stage. In this stage, we aim to expand the class-specific boundaries of the classifiers (F_1, F_2) in relation to the fixed ViT encoder E_1 by maximizing the discrepancy loss between their outputs. This means we have a possibility to estimate worst-case hyperspaces, which identifies target samples that fall outside the broader source distribution's support, rather than using CNN encoder E_2. To avoid this issue, we add a supervised loss on the labeled samples as follows:

$$\mathcal{L}_{find}(\boldsymbol{\theta}_{F_1}, \boldsymbol{\theta}_{F_2}) = \mathcal{L}_{vit}^{sup} + \mathcal{L}_{cnn}^{sup}$$
$$- \frac{1}{\mathcal{N}_{\mathcal{T}_u}} \sum_{i=1}^{\mathcal{N}_{\mathcal{T}_u}} d\big(p_1^{find}(x_i^{\mathcal{T}_u}), p_2^{find}(x_i^{\mathcal{T}_u})\big), \tag{4}$$

where $p_1^{find}(x_i^{\mathcal{T}_u}) = \sigma(F_1(E_1(x_i^{\mathcal{T}_u})))$ and $p_2^{find}(x_i^{\mathcal{T}_u}) = \sigma(F_2(E_1(x_i^{\mathcal{T}_u})))$ denote the probability outputs of F_1 and F_2 with ViT encoder E_1 on the unlabeled target data $x_i^{\mathcal{T}_u}$, respectively. By integrating the supervised loss on the labeled samples, we ensure that the classifiers not only distinguish different classes, but also push to deviate as far as possible from the currently learned class-specific boundaries while all labeled samples are correctly distinguished. In essence, the Finding Stage is a delicate balance between maximizing the discrepancy to explore the class-specific boundaries and minimizing the supervised loss to maintain classification accuracy.

Conquering Stage. In this stage, we leverage the class-specific boundaries established by the ViT encoder E_1 as a reference to guide the optimization of the CNN encoder E_2, while keeping the classifiers F_1 and F_2 fixed. The primary is to minimize the discrepancy between the outputs of two classifiers, F_1 and F_2, allowing the features extracted by CNN encoder E_2 to be accurately classified. The objective function is formulated as follows:

$$\mathcal{L}_{conq}(\boldsymbol{\theta}_{E_2}) = \frac{1}{\mathcal{N}_{\mathcal{T}_u}} \sum_{i=1}^{\mathcal{N}_{\mathcal{T}_u}} d\big(p_1^{conq}(x_i^{\mathcal{T}_u}), p_2^{conq}(x_i^{\mathcal{T}_u})\big), \tag{5}$$

where $p_1^{conq}(x_i^{\mathcal{T}_u}) = \sigma(F_1(E_2(x_i^{\mathcal{T}_u})))$ and $p_2^{conq}(x_i^{\mathcal{T}_u}) = \sigma(F_2(E_2(x_i^{\mathcal{T}_u})))$ denote the probability outputs of F_1 and F_2 with CNN encoder E_2 on the unlabeled target data, $x_i^{\mathcal{T}_u}$. By focusing on minimizing the discrepancy loss, the features extracted by the CNN encoder E_2 are clustered on the

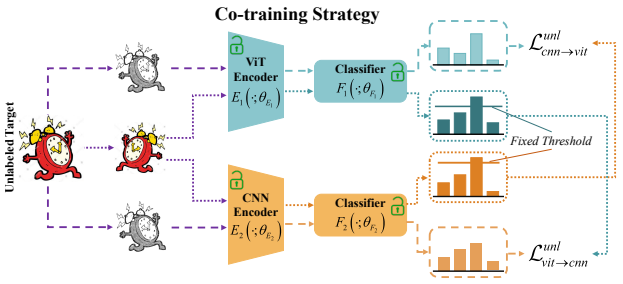


Figure 3. Illustration of co-training strategy.

class-specific boundaries of the ViT encoder E_1, even when estimating the worst-case hyperspaces. This alignment is crucial as it guarantees that the unlabeled samples are not only correctly classified but also robust to variations in the class-specific boundaries.

3.3. Co-training

Following the Finding-to-Conquering (FTC) strategy, we recognize a significant gap in the knowledge discrepancy between the ViT and CNN branches that needs to be minimized for the optimal performance. To resolve this, we have adopted a co-training strategy with dual objectives as illustrated in Fig. 3. The first objective focuses on reducing the gap between the two branches by enabling mutual enhancement and leveraging each branch's strength to improve the quality of pseudo labels. The second objective is to specifically improve the performance of the CNN branch, thanks to the potential to capture complex patterns and relationships in the data of the ViT branch.

Initially, we employ pseudo labels generated by the ViT branch to teach the learning process of the CNN branch. This is achieved by setting a fixed threshold, denoted as τ_{vit}, and applying it as follows:

$$\mathcal{L}^{unl}_{vit\to cnn}(\boldsymbol{\theta}_{E_2}, \boldsymbol{\theta}_{F_2}) =$$

$$\frac{1}{\mathcal{N}_{\mathcal{T}_u}} \sum_{i=1}^{\mathcal{N}_{\mathcal{T}_u}} \mathbb{1}[\max(\mathbf{q}_i^v) \geq \tau_{vit}] H\big(\hat{q}_i^v, p^c(x_{i,str}^{\mathcal{T}_u})\big), \quad (6)$$

where $\mathbb{1}[\cdot]$ is the binary indicator, returning 1 if the condition inside $[\cdot]$ is satisfied, and 0 otherwise. In this context, $\mathbf{q}_i^v = \sigma(F_1(E_1(x_{i,w}^{\mathcal{T}_u})))$ indicates the ViT branch's prediction for a weakly augmented version of an unlabeled target sample. Then, the highest prediction, exceeding the fixed threshold τ_{vit}, is converted into pseudo label $\hat{q}_i^v = \texttt{argmax}(\mathbf{q}_i^v)$. Finally, the output prediction $p^c(x_{i,str}^{\mathcal{T}_u}) = \sigma(F_2(E_2(x_{i,str}^{\mathcal{T}_u})))$ of the CNN branch on the strongly augmented target sample is adjusted to closely align with the pseudo label $\hat{q}_i^v$ using the cross-entropy loss. This process ensures that the CNN branch effectively learns from the ViT branch's predictions, making the consistency and alignment between the two branches better.

Similarly, we also leverage the pseudo labels from the CNN branch to guide the learning process of the ViT branch. This is achieved by setting another fixed threshold, denoted as τ_{cnn}, and using it as follows:

$$\mathcal{L}^{unl}_{cnn\to vit}(\boldsymbol{\theta}_{E_1}, \boldsymbol{\theta}_{F_1}) =$$

$$\frac{1}{\mathcal{N}_{\mathcal{T}_u}} \sum_{i=1}^{\mathcal{N}_{\mathcal{T}_u}} \mathbb{1}[\max(\mathbf{q}_i^c) \geq \tau_{cnn}] H\big(\hat{q}_i^c, p^v(x_{i,str}^{\mathcal{T}_u})\big), \quad (7)$$

where $\mathbf{q}_i^c = \sigma(F_2(E_2(x_{i,w}^{\mathcal{T}_u})))$ is the CNN branch's prediction for a weakly augmented target sample. The pseudo label $\hat{q}_i^c$ is generated from the highest predictions of the CNN branch that is higher than the threshold τ_{cnn}. Meanwhile, $p^v(x_{i,str}^{\mathcal{T}_u}) = \sigma(F_1(E_1(x_{i,str}^{\mathcal{T}_u})))$ indicates the output prediction of the ViT branch on the strongly augmented target sample and is adjusted to closely align with the pseudo label $\hat{q}_i^c$, using the cross-entropy loss. Through the co-training process, we enhance the model's ability to generalize and perform accurately on unlabeled target data.

3.4. Testing Phase

For a fair comparison with the previous DA methods [12, 16, 26, 31, 36], we select the CNN encoder E_2 associated with its classifier F_2 in the testing phase as illustrated in Fig. 2 as follows:

$$\hat{y}_i^{\mathcal{T}_u} = \texttt{argmax}\left(F_2(E_2(x_i^{\mathcal{T}_u}))\right), \quad (8)$$

where $\hat{y}_i^{\mathcal{T}_u}$ is the predicted class of unlabeled target sample.

4. Experiments

4.1. Experiment Setup

Datasets. We conduct extensive evaluations on standard DA benchmark datasets: *Office-Home* [29] and *DomainNet* [25]. On *Office-Home* dataset, we perform experiments on all possible combinations of these 4 domains: Real (R), Clipart (C), Art (A), and Product (P) with 65 categories. Consistent with prior SSDA methods [11, 16, 17, 31, 36], we conduct an evaluation on a subset of *DomainNet* that includes 126 categories in 4 domains: Real (rel), Clipart (clp), Painting (pnt), and Sketch (skt) using 7 different mixtures of these domains.

Implementation Details. We use the ViT/B-16 [7] for the ViT encoder E_1, while the ResNet [10] is adopted for the CNN encoder E_2. For UDA, we used ResNet-50 as the backbone network following previous works [2, 12, 30, 37]. Similarly, we follow the evaluation protocol of previous SSDA methods [9, 17, 28, 31], ResNet-34 is applied for this scenario on the *DomainNet* dataset. Each model is initially pre-trained on the ImageNet-1K [6]. Given the disparate nature of the backbone architectures, we will have

Method	$A \rightarrow C$	$A \rightarrow P$	$A \rightarrow R$	$C \rightarrow A$	$C \rightarrow P$	$C \rightarrow R$	$P \rightarrow A$	$P \rightarrow C$	$P \rightarrow R$	$R \rightarrow A$	$R \rightarrow C$	$R \rightarrow P$	Mean
DANN [8]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
MCD [30]	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
BNM [4]	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
MDD [37]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
MCC [12]	55.1	75.2	79.5	63.3	73.2	75.8	66.1	52.1	76.9	73.8	58.4	83.6	69.4
GVB [5]	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
DCAN [18]	54.5	75.7	81.2	67.4	74.0	76.3	67.4	52.7	80.6	74.1	59.1	83.5	70.5
DALN [2]	57.8	79.9	82.0	66.3	76.2	77.2	66.7	55.5	81.3	73.5	60.4	85.3	71.8
FixBi [22]	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
DCAN+SCDA [19]	60.7	76.4	82.8	69.8	77.5	78.4	68.9	59.0	82.7	74.9	61.8	84.5	73.1
ATDOC [20]	60.2	77.8	82.2	68.5	78.6	77.9	68.4	58.4	83.1	74.8	61.5	87.2	73.2
EIDCo [38]	63.8	80.8	82.6	71.5	80.1	80.9	72.1	61.3	84.5	78.6	65.8	87.1	75.8
ECB (CNN)	**68.5**	**85.4**	**88.3**	**79.2**	**86.8**	**89.0**	**79.3**	**66.4**	**88.5**	**81.0**	**71.1**	**90.4**	**81.2**

Table 1. **Accuracy (%) on Office-Home** of UDA methods across different domain shifts. **ECB (CNN)** represents the performance of our method when applied to ResNet-50. The top and second-best accuracy results are highlighted in **bold** and underline for easy identification.

Method	$rel \rightarrow clp$		$rel \rightarrow pnt$		$pnt \rightarrow clp$		$clp \rightarrow skt$		$skt \rightarrow pnt$		$rel \rightarrow skt$		$pnt \rightarrow rel$		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}
ENT [9]	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
MME [31]	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
S^3D [35]	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
ATDOC [20]	74.9	76.9	71.3	72.5	72.8	74.2	65.6	66.7	68.7	70.8	65.2	64.6	81.2	81.2	71.4	72.4
MAP-F [24]	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
DECOTA [34]	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 [16]	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 [28]	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
CDAC+SLA [36]	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	76.9
ProML [11]	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 [23]	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
G-ABC [17]	80.7	82.1	76.8	76.7	79.3	81.6	72.0	73.7	75.0	76.3	73.2	74.3	83.4	83.9	77.2	78.4
ECB (CNN)	**83.8**	**87.4**	**85.4**	**85.6**	**86.4**	**87.3**	**79.7**	**80.6**	**83.4**	**85.6**	**79.5**	**81.7**	**88.7**	**90.3**	**83.8**	**85.5**

Table 2. **Accuracy (%) on DomainNet** of SSDA methods in both 1-shot and 3-shot settings using ResNet-34.

two different classifiers, F_1 and F_2, consisting of two fully-connected layers followed by the softmax function. We optimize our models using stochastic gradient descent with a 0.9 momentum and 0.0005 weight decay. Considering the different architectures of the ViT and the CNN branches, we set the initial learning rates for ViT and CNN at $1e-4$ and $1e-3$, respectively. The size of the mini-batch is set to 32 for both $\mathcal{D}_l$ and $\mathcal{D}_{\mathcal{T}_u}$. The confidence threshold values for pseudo-label selection are set to $\tau_{vit} = 0.6$ and $\tau_{cnn} = 0.9$. The warmup phase for both branches on $\mathcal{D}_l$ is finetuned for $100,000$ iterations. Then, we update the learning rate scheduler and proceed with $50,000$ iterations of training for our method.

4.2. Comparison Results

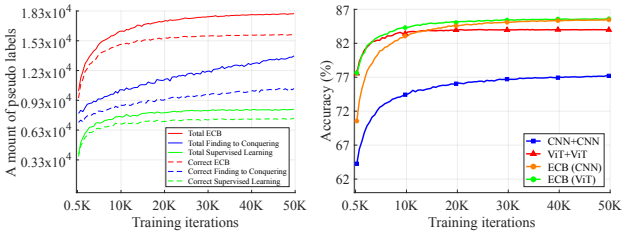
We evaluate our ECB method and the previous SOTA methods on *Office-Home* and *DomainNet* under UDA and SSDA settings, respectively. For a fair comparison with other DA methods, we rely on classification outcomes from the CNN branch, employing ResNet-50 as the backbone for UDA and ResNet-34 for SSDA. As a result, our method does not add any testing complexity compared to previous DA approaches. *Besides, we have included further details about the experimental results in Supplementary Materials.*

Results on *Office-Home* under UDA setting. The implementation of our ECB method has significantly boosted classification efficiency in all domain shift tasks, consistently outperforming the comparison methods, as detailed in Tab. 1. Remarkably, our approach has recorded accuracy enhancements of +7.7%, +8.1%, and +7.2% for the $C \rightarrow A$, $C \rightarrow R$, and $P \rightarrow A$ tasks, respectively, surpassing the results of the second-best. In addition, our method has achieved an impressive average classification accuracy of 81.2%, showing a remarkable margin of +5.4% over the nearest-competitor EIDCo [38].

Results on *DomainNet* under SSDA setting. The results on the *DomainNet* dataset are presented for both 1-shot and 3-shot settings in Tab. 2, where the CNN branch of our ECB method outperforms all prior methods. In comparison to the nearest-competitor method, G-ABC [17], the ECB (CNN) achieves an impressive maximum performance increase of +9.3% in the $skt \rightarrow pnt$ task for 3-shot learning. Even in the

Method	*rel→clp*		*rel→pnt*		*pnt→clp*		*clp→skt*		*skt→pnt*		*rel→skt*		*pnt→rel*		Mean	
	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN	ViT	CNN
vit→cnn	73.3	79.0	78.8	81.0	75.1	79.2	71.6	74.7	78.6	80.8	67.2	72.0	88.1	88.8	76.1	79.4
cnn→vit	74.2	61.9	76.8	66.8	76.1	67.4	69.5	57.2	74.9	64.6	67.4	54.8	86.0	76.1	75.0	64.1
co-training	87.4	87.4	85.8	85.6	87.3	87.3	80.7	80.6	85.8	85.6	81.7	81.7	90.9	90.3	85.7	85.5

Table 3. **Ablation study on DomainNet** between co-training and one-direction teaching under 3-shot settings.



(a) (b)

Figure 4. (a) The quality and quantity of the pseudo labels are generated by the CNN branch on *DomainNet* under the 3-shot setting of the *rel→clp* task using ResNet-34. (b) Comparison between backbone settings on *DomainNet* under the 3-shot setting. Displayed is the mean accuracy across all domain shift tasks.

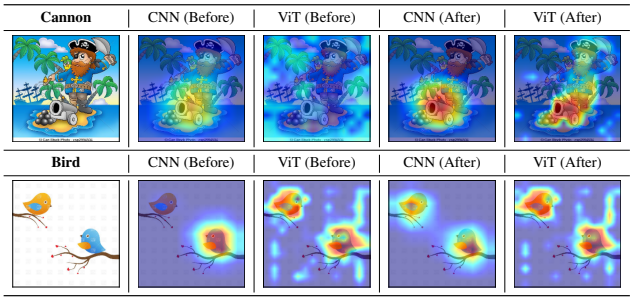


Table 4. **Visualize the feature maps** for the '*Cannon*' and '*Bird*' examples to investigate the learning behaviors of CNN and ViT with and without using the proposed method ECB.

more restrictive 1-shot learning, the ECB method demonstrates robust performance, showing a minimum increase of +3.1% in the *rel→clp* task. On average, the ECB method validates a performance improvement of +6.6% in the 1-shot setting and +7.1% in the 3-shot setting.

5. Analyses

Ablation studies. A single classifier is concurrently trained with both labeled source and target data. It is thus easily dominated by the labeled source data. This imbalance prevents the classifier from using the extra-labeled target data effectively, resulting in a misalignment between the learned and true class-specific boundaries. This misalignment is called data bias. Therefore, we perform an ablation study to evaluate the effectiveness of each stage in the proposed method. The emphasis is on addressing the data bias issue within the ViT branch, leveraging the CNN branch's capability to generate high-quality pseudo labels during the *rel→clp* task on *DomainNet* under, as illustrated in Fig. 4a. Initially, during the Supervised Training phase, we are only able to generate around 8,000 total pseudo labels. Following the integration of the FTC strategy, the total pseudo labels smoothly increase up to 14,000 with steady maintenance of 10,000 correct pseudo labels. This plays a crucial role in substantially mitigating data bias by generating numerous pseudo-labels, which improves the diversity and representativeness of the unlabeled target dataset. The final experiment is implemented to highlight that co-training is necessary for reducing the remaining gap following the FTC stage, which yields a total of 18,000 pseudo labels and 16,000 correct pseudo labels.

Effectiveness of co-training. We further investigate a variant of ECB termed one-direction teaching on the *DomainNet* under the 3-shot setting. In this approach, we employ either $\mathcal{L}^{unl}_{vit\rightarrow cnn}$ or $\mathcal{L}^{unl}_{cnn\rightarrow vit}$ to generate pseudo labels for the remaining branch, while maintaining the standard configurations through the use of Supervised Training followed by FTC strategy. As depicted in Tab. 3, there is a performance drop when using one-direction teaching. Specifically, in the *vit→cnn* scenario, the ViT branch generates pseudo labels for the unlabeled target data to teach the CNN branch through minimizing $\mathcal{L}^{unl}_{vit\rightarrow cnn}$. It is crucial to highlight that the ViT branch does not receive pseudo labels from the CNN branch, resulting in the CNN branch outperforming the ViT branch by +3.3%. In contrast, in the *cnn→vit* scenario, where $\mathcal{L}^{unl}_{cnn\rightarrow vit}$ is minimized, the average performance on the ViT branch is +10.9% more outstanding than the CNN branch. However, it is worth mentioning that the performance on the ViT branch suffers from -1.1% reduction over the *vit→cnn* scenario. This performance drop is attributed to the introduction of noise by the CNN branch during the learning process of the ViT branch. On the other hand, when employing $\mathcal{L}^{unl}_{vit\rightarrow cnn}$, the CNN branch provides a significant boost in the average performance by +15.3%, highlighting the superior generalization capabilities of the ViT branch compared to the CNN branch. These scenarios try to leverage the unique strengths of CNN and ViT through the co-training strategy, showcasing the mutual exchange of knowledge between two branches and its potential for generalizing to unlabeled target data. As a result of co-training,

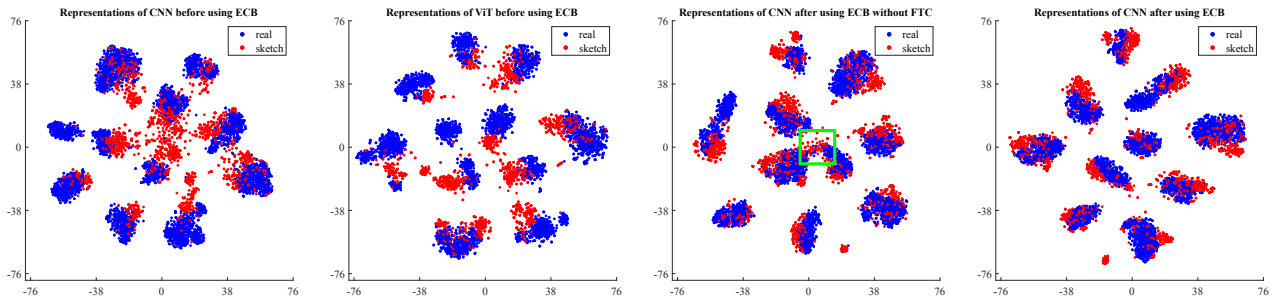


Figure 5. We visualize feature spaces for the $rel \rightarrow skt$ task on *DomainNet* in the 3-shot scenario using t-SNE [33]. Figures (a) and (b) illustrate the features obtained by CNN and ViT branches before adaptation, respectively. Figures (c) and (d) showcase the distribution changes of the CNN branch depending on the presence of the FTC strategy when implementing our ECB method.

the average performance on the target domain of the ViT and CNN branches reaches the optimal results of 85.7% and 85.5%, respectively.

Architecture analysis. We examine the effectiveness of the unique strengths of ViT and CNN over variations of architectures such as "CNN + CNN" and "ViT + ViT" on *DomainNet* under 3-shot settings using ResNet-34 for CNN and ViT/B-16 for ViT as shown in Fig. 4b. On average, the results show that "CNN + CNN" only achieves around 77.0% due to the lack of global information when just using CNN. In addition, the introduction of ViT with the primary goal of improving performance drives the "ViT + ViT" architecture to achieve a significant increase of +7.0% compared to the "CNN + CNN" architecture. This architecture makes it possible to establish more general class-specific boundaries, yet is unfair when compared with the previous SSDA methods [20, 30, 31, 34, 36] using ResNet-34 as the backbone, and the lack of local representation is extremely important. As a result, the evidence shows that combining "CNN + ViT" with the bridge of co-training achieves the highest accuracy of about 85.5% for both branches.

Attention map visualization analysis. To demonstrate the effectiveness of our designing framework methodology, we provide the attention map visualization results by using Grad-CAM [32], as visualized in Tab. 4. *CNN supports ViT*: Before applying ECB, ViT was background-sensitive in the "*Cannon*" class, whereas CNN still captures the correct object. However, after applying ECB, ViT can explicitly recognize the target object and back to enhance the robustness of CNN. *ViT supports CNN*: In the "*Bird*" class before applying ECB, CNN only obtains the part of the target, while ViT can cover the whole input object. Then, CNN is complemented to focus on the object accurately by ViT using ECB. Consequently, these findings validate that both branches offer distinct expertise and enhance each other rather than one overshadowing the other.

Feature Visualization. We present a detailed visualization of the feature space for the *DomainNet* dataset within the $rel \rightarrow skt$ task under the 3-shot setting. This visualization distinctly highlights the source domain (blue-colored) and the target domain (red-colored). Figs. 5a and 5b show the domain alignment of both the CNN and ViT branches before adaptation. In particular, Fig. 5a visualizes the feature space extracted by the CNN branch before applying ECB, reflecting a scatter that indicates weak classifier performance. In contrast, Fig. 5b displays the well-defined clusters in the ViT branch for the same unlabeled target data, which emphasizes the robustness of ViT in identifying more general class-specific boundaries compared to the CNN branch. Figs. 5c and 5d show the distribution changes of the CNN branch depending on the presence of the FTC strategy when implementing ECB. Initially, Fig. 5c indicates that the target representations overlap (highlighted with green box) when implementing ECB without the FTC strategy. On the other hand, Fig. 5d shows the well-aligned source and target domain representations with clusters to distinct separation when applying the FTC strategy, which demonstrates the effectiveness of our proposed method.

6. Conclusion

In this work, we have developed a novel method for learning CNN on ViT with ECB strategy, taking advantage of ViT and CNN. This innovative approach focuses on reducing data bias and significantly improves the precision of pseudo labels generated, which aligns between source and target domains. Our method is also fair when evaluated on the CNN branch and outperforms the previous SOTA methods on various DA benchmark datasets.

Discussions. We found that identifying an optimal threshold pair $\{\tau_{vit}$ and $\tau_{cnn}\}$ was quite time-consuming. Therefore, we leave an open task for future research that uses a dynamic threshold algorithm for domain adaptation instead of the fixed threshold.

Acknowledgement. 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).

References

[1] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, 1998. 2

[2] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7181–7190, 2022. 2, 5, 6

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE Conf. Comput. Vis. Pattern Recog. Works.*, 2020. 3

[4] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3941–3950, 2020. 2, 6

[5] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12455–12464, 2020. 2, 6

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, page 248–255, 2009. 5

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Thomas Unterthiner Xiaohua Zhai, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021. 1, 5

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17 (1):2096–2030, 2016. 6

[9] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Adv. Neural Inform. Process. Syst.*, 17, 2004. 5, 6

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, page 770–778, 2016. 5

[11] Xinyang Huang, Chuang Zhu, and Wenkai Chen. Semi-supervised domain adaptation via prototype-based multi-level learning. *Proc. IJCAI*, 2023. 2, 5, 6

[12] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *Eur. Conf. Comput. Vis.*, pages 464–480. Springer, 2020. 2, 5, 6

[13] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *IEEE Conf. Comput. Vis. Pattern Recog.*, page 14274–14285, 2020. 1

[14] Ju Hyun Kim, Ba Hung Ngo, Jae Hyeon Park, Jung Eun Kwon, Ho Sub Lee, and Sung In Cho. Distilling and refining domain-specific knowledge for semi-supervised domain adaptation. In *British Machine Vision Conference, BMVC*, page 606, 2022. 2

[15] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *Eur. Conf. Comput. Vis.*, pages 591–607, 2020. 2

[16] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2505–2514, 2021. 2, 5, 6

[17] Jichang Li, Guanbin Li, and Yizhou Yu. Adaptive betweenness clustering for semi-supervised domain adaptation. *IEEE Trans. Image Process.*, 2023. 5, 6

[18] Shuang Li, Chi Liu, Qiuxia Lin, Binhui Xie, Zhengming Ding, Gao Huang, and Jian Tang. Domain conditioned adaptation network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11386–11393, 2020. 6

[19] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *Int. Conf. Comput. Vis.*, pages 9102–9111, 2021. 2, 6

[20] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16632–16642, 2021. 6, 8

[21] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105. PMLR, 2015. 2

[22] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1094–1103, 2021. 2, 6

[23] Ba Hung Ngo, Ju Hyun Kim, Yeon Jeong Chae, and Sung In Cho. Multi-view collaborative learning for semi-supervised domain adaptation. *IEEE Access*, volume 9:166488–166501, 2021. 2, 6

[24] Ba Hung Ngo, Jae Hyeon Park, So Jeong Park, and Sung In Cho. Semi-supervised domain adaptation using explicit class-wise matching for domain-invariant and class discriminative feature learning. *IEEE Access*, volume 9:128467–128480, 2021. 6

[25] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Int. Conf. Comput. Vis.*, pages 1406–1415, 2019. 5

[26] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Int. Conf. Comput. Vis.*, pages 8558–8567, 2021. 2, 5

[27] Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu. Contradictory structure learning for semi-supervised domain adaptation. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 576–584, 2021. 2

[28] Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu. Semi-supervised domain adaptive structure

learning. *IEEE Trans. Image Process.*, 31:7179–7190, 2022. 2, 5, 6

[29] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Deep hashing network for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 5

[30] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 5, 6, 8

[31] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Int. Conf. Comput. Vis.*, pages 8050–8058, 2019. 2, 5, 6, 8

[32] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Int. Conf. Comput. Vis.*, pages 618–626, 2017. 8

[33] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, volume 9:770–778, 2008. 8

[34] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q. Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. In *Int. Conf. Comput. Vis.*, pages 8906–8916, 2021. 2, 3, 6, 8

[35] Jeongbeen Yoon, Dahyun Kang, and Minsu Cho. Semi-supervised domain adaptation via sample-to-sample self-distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1978–1987, 2022. 6

[36] Yu-Chu Yu and Hsuan-Tien Lin. Semi-supervised domain adaptation with source label adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 2, 5, 6, 8

[37] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019. 2, 5, 6

[38] Yixin Zhang, Zilei Wang, Junjie Li, Jiafan Zhuang, and Zihan Lin. Towards effective instance discrimination contrastive loss for unsupervised domain adaptation. In *Int. Conf. Comput. Vis.*, pages 11388–11399, 2023. 6