Cross-Domain Adaptive Clustering for Semi-Supervised Domain Adaptation

Jichang Li[1] Guanbin Li[2*] Yemin Shi[3] Yizhou Yu[1*]
[1]The University of Hong Kong [2]Sun Yat-sen University [3]Deepwise AI Lab
csjcli@connect.hku.hk, liguanbin@mail.sysu.edu.cn, shiyemin@deepwise.com, yizhouy@acm.org

Abstract

In semi-supervised domain adaptation, a few labeled samples per class in the target domain guide features of the remaining target samples to aggregate around them. However, the trained model cannot produce a highly discriminative feature representation for the target domain because the training data is dominated by labeled samples from the source domain. This could lead to disconnection between the labeled and unlabeled target samples as well as misalignment between unlabeled target samples and the source domain. In this paper, we propose a novel approach called Cross-domain Adaptive Clustering to address this problem. To achieve both inter-domain and intra-domain adaptation, we first introduce an adversarial adaptive clustering loss to group features of unlabeled target data into clusters and perform cluster-wise feature alignment across the source and target domains. We further apply pseudo labeling to unlabeled samples in the target domain and retain pseudo-labels with high confidence. Pseudo labeling expands the number of "labeled" samples in each class in the target domain, and thus produces a more robust and powerful cluster core for each class to facilitate adversarial learning. Extensive experiments on benchmark datasets, including DomainNet, Office-Home and Office, demonstrate that our proposed approach achieves the state-of-the-art performance in semi-supervised domain adaptation.

1. Introduction

Semi-supervised domain adaptation (SSDA) is a variant of the unsupervised domain adaptation (UDA) problem. With a small number of labeled samples in the target domain, SSDA has the potential to significantly boost performance in comparison to UDA. In general, domain adaptation needs to reduce inter-domain gap (*i.e.*, differences in feature distributions between two domains) and decrease intra-domain gap (*i.e.*, differences among class-wise sub-distributions in the target domain) in order to achieve inter-domain adaptation and intra-domain adaptation simultaneously [26].

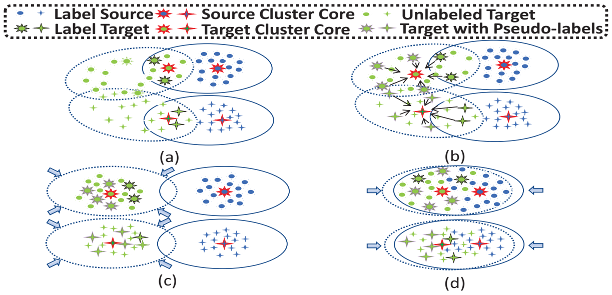

Figure 1. Overview of our Cross-domain Adaptive Clustering (CDAC) approach. (a) Supervision on labeled data from both source and target domains to ensure partial cross-domain feature alignment. (b) Pseudo labeling for giving pseudo-labels on unlabeled target samples to form enhanced target cluster cores with higher robustness and power. (c) Minimization of the adversarial adaptive clustering loss for grouping features from the target domain into clusters. (d) Maximization of the adversarial adaptive clustering loss to facilitate cross-domain cluster-wise feature alignment.

Many existing domain adaptation approaches start with inter-domain adaptation, and guide their models to learn cross-domain sample-wise feature alignment [34, 6, 4], or distribution-wise feature alignment [12, 22, 18]. In the semi-supervised learning setting, adversarial learning is employed in [32, 25] to improve sample-wise feature alignment for inter-domain adaptation. However, such previous work ignores extra information indicated by class-wise sub-distributions in the target domain, and thus results in cross-domain feature mismatch during model training, thereby reducing model generalization performance on novel test data in the target domain.

Whereafter, much work on domain adaptation has turned to intra-domain adaptation [16, 13]. By optimizing class-wise sub-distributions within the target domain, the generalization performance of adaptation models can be improved. In the context of semi-supervised domain adaptation, the presence of few labeled target samples is utilized to help features of unlabeled target samples from different

*Corresponding Authors.

classes be guided to aggregate in the corresponding clusters to form perfect class-wise sub-distributions in the target domain, which reduces the possibility of feature mismatch across domains. However, a model trained with supervision on few labeled target samples and labeled source data just can ensure partial cross-domain feature alignment because it only aligns the features of labeled target samples and their correlated nearby ones with the corresponding feature clusters in the source domain [17]. Also, the trained model cannot produce a highly discriminative feature representation for the target domain because the learned feature representation is biased to the sample discrimination of the source domain due to the existence of a much larger scale of labeled samples than those of the target domain [32]. These could lead to disconnection between the labeled and unlabeled target samples as well as misalignment between unlabeled target samples and the source domain.

In this paper, we propose a novel approach called **C**ross-**d**omain **A**daptive **C**lustering (**CDAC**), as Figure 1 shows, to address the aforementioned problem. It first groups features of unlabeled target data into clusters and further performs cluster-wise feature alignment across the source and target domains rather than sample-wise or distribution-wise feature alignment. In this way, our approach achieves both inter-domain adaptation and intra-domain adaptation simultaneously. More specifically, our proposed approach performs minimax optimization over the parameters of a feature extractor and a classifier. For intra-domain adaptation, the features of unlabeled target samples are guided by labeled target samples to form clusters corresponding to the classes of labeled samples by minimizing an adversarial adaptive clustering loss with respect to the parameters of the feature extractor. For inter-domain adaptation, the classifier is trained to maximize the same loss defined on unlabeled target samples so that cluster-wise feature distribution in the target domain is aligned with the corresponding feature distribution in the source domain.

In addition, we apply pseudo labeling to unlabeled samples in the target domain and retain pseudo-labels with high confidence. In the SSDA setting, since only a very small number (typically one or three) of target samples from each class are labeled, it is hard for such few samples to form a stable and accurate cluster core. Pseudo labeling expands the number of “labeled" samples in each class in the target domain, and thus produces a more robust and powerful cluster core for each class. Such an enhanced cluster core can attract unlabeled samples from the corresponding class towards itself in the target domain using the adversarial adaptive clustering loss. Therefore, our pseudo labeling technique assists adversarial learning, and helps our SSDA model reach higher performance.

In summary, our main contributions of the proposed Cross-domain Adaptive Clustering (CDAC) approach are as follows.

- We introduce an adversarial adaptive clustering loss to perform cross-domain cluster-wise feature alignment so as to solve the SSDA problem.

- We integrate an adapted version of pseudo labeling to enhance the robustness and power of cluster cores in the target domain to facilitate adversarial learning.

- Extensive experiments on benchmark datasets, including *DomainNet* [28], *Office-Home* [39] and *Office* [31], demonstrate that our proposed CDAC approach achieves the state-of-the-art performance in semi-supervised domain adaptation.

2. Related Work

2.1. Adversarial Learning for UDA

Most domain adaptation algorithms attempt to achieve feature distribution alignment between domains by minimizing the domain shift between the source domain and the target domain, so that the knowledge learned from the source data can be transferred to the target domain and improve its classification performance [27, 10]. Adversarial learning is one of the mainstream solutions [44, 38, 3]. Saito *et al.* [33] proposed to train task-specific classifiers and maximize their output discrepancy to detect target samples that are far from the support of the source distribution, then learn to generate target features near the support to fool the classifiers. [40, 26] introduce entropy-based adversarial training to enhance high-confident predictions in the target domain. Moreover, in order to overcome the issue of mode collapse caused by the separate design of task and domain classifiers, Tang *et al.* [36] proposed discriminative adversarial learning to promote the joint distribution alignment within both feature-level and class-level.

Different from previous sample-wise adversarial learning based domain adaptation methods, we first propose adaptive cluster-wise feature alignment to achieve both inter-domain and intra-domain adaptation. This method can greatly alleviate the situation that the model produces feature representations with bias towards the source domain caused by the dominance of most labeled source samples during model training, and can reduce the difficulty of exploring the decision boundary of the classifier by improving the cohesion of unlabeled samples in the target domain, so as to improve the performance of the model in a two-pronged manner.

2.2. Pseudo Labeling on UDA

Pseudo labeling, a.k.a self-training, is often used in semi-supervised learning, aiming to give reliable pseudo-labels

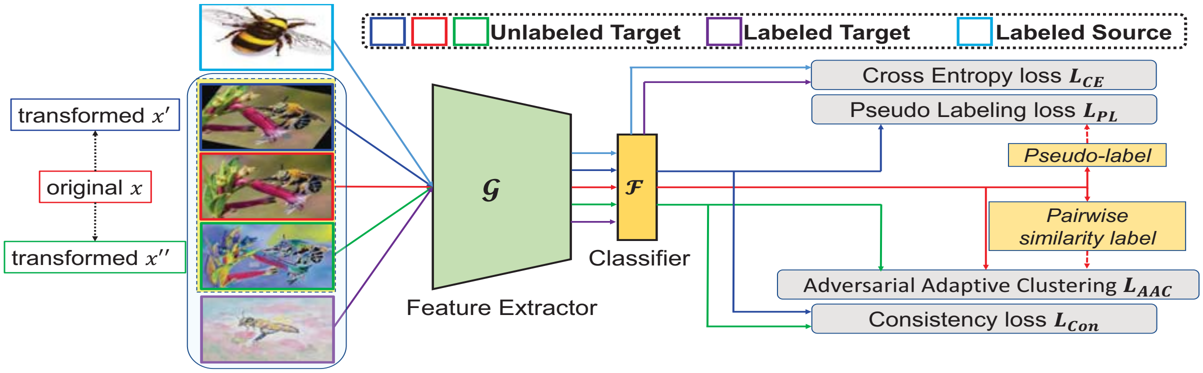

Figure 2. Outline of our model architecture and training procedure. Arrows with various colors represent data flows for different types of samples from both source and target domains. The feature extractor $\mathcal{G}$ uses Alexnet or Resnet34 as the backbone network and the classifier $\mathcal{F}$ is an unbiased linear network with a normalized layer, which is shared by both domains. As shown, an image x from unlabeled target data is first fed to the feature extractor and the classifier and then its prediction is constructed to pairwise similarity label and pseudo-label, which are employed as targets for its two different transformed versions, x' and x'', to train the model with the adversarial adaptive clustering loss and the proposed pseudo labeling loss, respectively.

to unlabeled data through an ensemble of output predictions from multiple models and assist model training to improve performance and its generalization [20, 11]. In the field of semi-supervised image classification, the reliability of pseudo-labels is usually improved by integrating the output predictions of one model with multiple augmented inputs [2, 35], outputs of different models [41], or multiple predictions of the same model in different training stages [43, 1, 7]. In previous researches, pseudo labeling is also proved to be effective in domain adaptation, *e.g.*, [26] proposed entropy-based ranking function to separate the target domain data into an easy and hard split followed by employing self-supervised adaptation from easy to hard for decreasing intra-domain gap. To avoid introducing noise from pseudo labeling, [13] constructed a robust Gaussian-Uniform mixture model in spherical feature space to guarantee the correctness of given pseudo-labels from unlabeled target data.

In this work, pseudo labeling is employed to give pseudo-labels for unlabeled target data with high probabilistic confidence and thus expand the number of “labeled” samples in each class of the target domain, resulting in a more robust and powerful cluster core for each class to facilitate adversarial learning.

2.3. Semi-supervised Domain Adaptation

Semi-supervised domain adaptation (SSDA) is a relatively promising form of transfer learning, which intents to leverage a small number of labeled samples (e.g, one or few samples per class) in the target domain and give full play to their potential to greatly improve the performance of domain adaptation. Recently, SSDA has recently attracted wide attentions [32, 29, 15, 21, 17, 42] from researchers. [32, 29] first proposed to solve SSDA by aligning the features from both domains by means of adversarial learning. [15] proposed to reduce intra-domain discrepancy within the target domain to attract unaligned target sub-distributions towards the corresponding source sub-distributions so as to improve feature alignment across domains. In addition, [26] proposed to decompose SSDA into a semi-supervised learning (SSL) problem in the target domain and an unsupervised domain adaptation (UDA) problem across domains, and then train two classifiers using Mixup and Co-training methods, so as to bridge the gap and exchange expertise between the source and target domains. Furthermore, [21] proposed to explore the optimal initial weights for the adaptation model using online meta-learning. Most of the previous approaches solve SSDA based on sample-wise feature alignment. In this work, we take an attempt to use adaptive cluster-wise feature alignment affiliated with pseudo labeling to achieve both inter-domain and intra-domain adaptation.

3. Methodology

In this section, we first introduce the background and notations of SSDA, and then present our proposed Cross-domain Adaptive Clustering (CDAC) approach, which contains an adversarial adaptive clustering loss and a pseudo labeling loss. Finally, we summarize the overall loss used in our work. An outline of our model architecture and training procedure is shown in Figure 2.

3.1. Semi-supervised Domain Adaptation

Semi-supervised domain adaptation seeks a classifier for a target domain when given labeled data $\mathcal{S} = (x_i^s, y_i^s)_{i=1}^{N_s}$ from a source domain as well as both unlabeled data $\mathcal{U} = \{(x_i^u)\}_{i=1}^{N_u}$ and labeled data $\mathcal{L} = \left\{\left(x_i^l, y_i^l\right)\right\}_{i=1}^{N_l}$ from the target domain. $\mathcal{S}$,$\mathcal{U}$ and $\mathcal{L}$ represent three subsets of available data in this problem, and they contain N_s, N_u and N_l instances, respectively. In the semi-supervised setting, N_l is much smaller than N_s and N_u, and only contains one shot or few shots per class. Each data point $x_i^s(x_i^l)$ from $\mathcal{S}(\mathcal{L})$ has its associated label $y_i^s(y_i^l)$, while any data point x_i^u from $\mathcal{U}$ has none. Our work aims to make our SSDA model trained using $\mathcal{S}, \mathcal{U}$ and $\mathcal{L}$ perform well on test data from the target domain.

Our network consists of two components, *i.e.*, a feature extractor $\mathcal{G}$, parameterized by $\theta_{\mathcal{G}}$, and a classifier $\mathcal{F}$, parameterized by $\theta_{\mathcal{F}}$, as in existing work [32, 15, 17]. The classifier $\mathcal{F}$ is an unbiased linear network with a normalization layer, which maps features from the feature extractor $\mathcal{G}$ into a spherical feature space. This similarity-based feature space is more suitable for decreasing the feature variance of samples sharing the same class label [32, 17]. These are commonly used model settings for the SSDA problem [32, 15, 17].

The feature of an input image x, $\mathcal{G}(x)$, is fed into the classifier $\mathcal{F}$ to obtain the probabilistic prediction as follows:

$$p\left(x\right) = \sigma(\mathcal{F}(\mathcal{G}(x))), \tag{1}$$

where $\sigma\left(\cdot\right)$ is the softmax function. For convenience, we often abbreviate $p\left(x\right)$ as $\mathbf{p}$, *i.e.*, $\mathbf{p} = p\left(x\right)$.

To train our model with supervision from all labeled data from both source and target domains, we follow the practices of existing work on SSDA [32, 29, 15, 21, 17, 42], and include the following standard cross-entropy loss in the training loss,

$$\boldsymbol{L_{CE}} = -\sum_{(x,y)\in\mathcal{S}\cup\mathcal{L}} y\log\left(p\left(x\right)\right). \tag{2}$$

3.2. Adversarial Adaptive Clustering

The key idea in our work is the introduction of an adversarial adaptive clustering loss into semi-supervised domain adaptation to group features in the target domain into clusters and further perform cross-domain cluster-wise feature alignment to achieve inter-domain adaptation and intra-domain adaptation simultaneously. Underlying assumptions are that features of sample images form clusters and samples from the same cluster should have similar features and share the same class label. This loss first computes pairwise similarities among features of unlabeled samples in the target domain, then forces the class labels predicted by the classifier for such samples with pairwise feature similarities to be consistent. The latter is achieved by training the model with a binary cross-entropy loss, where binary pairwise feature similarities are used as groundtruth labels. This loss can force the features from the target domain to form clusters.

In detail, the above approach requires setting up connections on the basis of a similarity measure between sample pairs (x_i^u,x_j^u) from the same mini-batch. According to the above assumption, for a pair of similar samples, we set a pairwise pseudo-label $s_{ij} = 1$ (*i.e.*, pairwise connection between paired samples); otherwise, $s_{ij} = 0$ for dissimilar samples. According to [14], pairwise feature similarity can be measured using the indices of feature elements rank ordered according to their magnitudes. If two samples share the same top-k indices in their respective lists of rank ordered feature elements, the paired samples belong to the same class with a high confidence and thus $s_{ij} = 1$; otherwise, $s_{ij} = 0$. Therefore, we can formulate pairwise similarity label as follows,

$$s_{ij} = \mathbb{1}\{\text{top}k\left(\mathcal{G}\left(x_i^u\right)\right) = \text{top}k\left(\mathcal{G}\left(x_j^u\right)\right)\}, \tag{3}$$

where top$k\left(\cdot\right)$ denotes the top-k indices of rank ordered feature elements and we set $k = 5$. And $\mathbb{1}\{\cdot\}$ is an indicator function.

Then we establish pairwise comparisons among unlabeled target data using the binary cross-entropy loss, which utilize the above pairwise feature similarity labels of sample pairs in a mini-batch as targets, *i.e.*, our adversarial adaptive clustering loss $\boldsymbol{L_{AAC}}$ can be written as follows,

$$\begin{aligned}\boldsymbol{L_{AAC}} = -\sum_{i=1}^{M}\sum_{j=1}^{M} & s_{ij}\log(\mathbf{p}_i^{\mathsf{T}}\mathbf{p}_j') \\ & + (1-s_{ij})\log(1-\mathbf{p}_i^{\mathsf{T}}\mathbf{p}_j'),\end{aligned} \tag{4}$$

where M is the number of unlabeled target samples in each mini-batch and $\mathbf{p}_i = p\left(x_i^u\right) = \sigma(\mathcal{F}(\mathcal{G}(x_i^u)))$ represents the prediction of an image x_i^u in the mini-batch. Also, $\mathbf{p}_i' = p\left(x_i'\right) = \sigma(\mathcal{F}(\mathcal{G}(x_i')))$ indicates the prediction of a transformed image x_j', which is an augmented version of x_j^u using a data augmentation technique. The inner product $\mathbf{p}_i^{\mathsf{T}}\mathbf{p}_j'$ in Equation (4) is used as a similarity score, which predicts whether image x_i^u and the transformed version of image x_j^u share the same class label or not. Besides, as illustrated in [30], data augmentation techniques combined in the process of pairwise comparison can significantly strengthen the model performance.

What is the goal of Cross-domain Adaptive Clustering achieved using the $\boldsymbol{L_{AAC}}$ loss? Similar to [32], we also enforce supervision on labeled samples from the source and target domains and perform minimax training on unlabeled target domain samples to optimize the model, but we replace the conditional entropy loss with our adversarial adaptive clustering loss. In our work, directly minimizing the $\boldsymbol{L_{AAC}}$ loss makes features of similar samples in the target domain close but features of dissimilar ones distant so that features form clusters within the target domain. However, the learned feature representation in the target domain would be always biased towards the source domain because a large number of source labels dominate the supervision process. Thus direct minimization of $\boldsymbol{L_{AAC}}$ over unlabeled target domain data would make this worse and give rise to more severe overfitting. Therefore, we utilize a gradient reversal layer [8] to flip the gradients of $\boldsymbol{L_{AAC}}$ between the feature extractor and the classifier and, in this situation, the classifier is still enforced to ensure correct classification in the target domain. In other words, the maximization of $\boldsymbol{L_{AAC}}$ on unlabeled target domain data would decrease the bias of feature representations towards the source domain and encourage the model to produce more domain-invariant features so as to facilitate cross-domain feature alignment. Thus, a preliminary loss function for adversarial learning in our network can be summarized as follows,

$$\begin{aligned} \theta_{\mathcal{G}}^{*} &= \arg\min_{\theta_{\mathcal{G}}} \boldsymbol{L_{CE}} + \lambda \boldsymbol{L_{AAC}}, \\ \theta_{\mathcal{F}}^{*} &= \arg\min_{\theta_{\mathcal{F}}} \boldsymbol{L_{CE}} - \lambda \boldsymbol{L_{AAC}}, \end{aligned} \tag{5}$$

where λ is a scalar hyper-parameter that controls the balance between the cross-entropy loss and the proposed adversarial adaptive clustering loss.

3.3. Pseudo Labeling for Unlabeled Target Domain Data

Due to the small number of labeled target domain samples in the SSDA problem, it is hard for the adversarial adaptive clustering loss to form stable and accurate cluster cores in the target domain during model training, which may negatively affect cross-domain cluster-wise feature alignment. To solve this problem, we apply pseudo labeling to unlabeled target samples and retain pseudo-labels with high confidence to expand the number of "labeled" samples in the target domain, thereby forming more robust cluster cores for different classes. Pseudo labeling is a classic technique for semi-supervised learning [1, 35], and utilizes the prediction capability of a model to generate artificial hard labels for a subset of unlabeled samples and then train the model with a supervised loss involving these artificial labels. In our work, we choose the progressive pseudo labeling technique in [35].

In the proposed pseudo labeling process, we first feed an image x_j^u from a mini-batch of unlabeled images into the current model, and the prediction $\mathbf{p}_j = p\left(x_j^u\right) = \sigma(\mathcal{F}(\mathcal{G}(x_j^u)))$ from the model is then converted to a one-hot hard label $\hat{y}_j^u = \arg\max\left(\mathbf{p}_j\right)$, which is used as a pseudo label in a supervised loss. Afterwards, the prediction $\mathbf{p}_j'' = p\left(x_j''\right)$ produced from another transformed image x_j'' for the same image x_j^u is obtained to increase the input diversity of our model. Therefore, in this section, our model is trained using the standard cross-entropy loss as follows,

$$\boldsymbol{L_{PL}} = -\sum_{j=1}^{M} \mathbb{1}\{\max(\mathbf{p}_j) \geq \tau\} \cdot \hat{y}_j^u \log(\boldsymbol{p}\left(x_j''\right)), \tag{6}$$

where $\mathbf{p}_j'' = p\left(x_j''\right) = \sigma(\mathcal{F}(\mathcal{G}(x_j'')))$ denotes the model prediction of the transformed image x_j'', and τ is a scalar confidence threshold that determines the subset of pseudo labels that should be retained for model training.

Our $\boldsymbol{L_{PL}}$ loss is employed to enhance the adversarial adaptive clustering loss. Once pseudo-labels with high confidence are identified and used for model training, more robust cluster cores in the target domain can be established to make the feature clusters in the target domain better aligned with the source domain ones.

3.4. Overall Loss

The overall loss function for training our SSDA network can be summarized as follows,

$$\begin{aligned} \theta_{\mathcal{G}}^{*} &= \arg\min_{\theta_{\mathcal{G}}} \boldsymbol{L_{CE}} + \lambda \boldsymbol{L_{AAC}} + \boldsymbol{L_{PL}} + \boldsymbol{L_{Con}}, \\ \theta_{\mathcal{F}}^{*} &= \arg\min_{\theta_{\mathcal{F}}} \boldsymbol{L_{CE}} - \lambda \boldsymbol{L_{AAC}} + \boldsymbol{L_{PL}} + \boldsymbol{L_{Con}}, \end{aligned} \tag{7}$$

where

$$\boldsymbol{L_{Con}} = w\left(t\right) \sum_{j=1}^{M} ||\mathbf{p}_j' - \mathbf{p}_j''||^2, \tag{8}$$

and $w\left(t\right) = \nu e^{-5\left(1-\frac{t}{T}\right)^2}$ is a ramp-up function used in [19] with the scalar coefficient ν, the current time step t and the total number of steps T in the ramp-up process. In order to improve the input diversity of our model, we have created two different transformed versions of each unlabeled image in the target domain to implement the adversarial adaptive clustering loss and the pseudo labeling loss, respectively. Therefore, we employ a consistency loss, $\boldsymbol{L_{Con}}$, to keep the model predictions on these two transformed images consistent.

4. Experiments

4.1. Setups

Benchmark datasets: We evaluate the efficacy of our proposed CDAC approach on several standard SSDA image classification benchmarks, including the *DomainNet*[1] [28], *Office-Home*[2] [39] and *Office*[3] [31]. *DomainNet* is initially a multi-source domain adaptation benchmark, and MME [32] borrows its subset as one of the benchmarks for SSDA evaluation. Similar to the setting of MME, we only select 4 domains, which are Real, Clipart, Painting, and Sketch (abbr. **R**, **C**, **P** and **S**), each of which contains images of 126 categories. *Office-Home* is a widely used UDA benchmark and consists of Real, Clipart, Art and Product (abbr. **R**, **C**, **A** and **P**) domains with 65 classes. *Office* is a relatively small dataset contains three domains including

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

[2] http://hemanthdv.org/OfficeHome-Dataset/

[3] https://people.eecs.berkeley.edu/ jhoffman/domainadapt/

Table 1. Accuracy(%) on *DomainNet* under the settings of 1-shot and 3-shot using Alexnet and Resnet34 as backbone networks.

Net	Method	R→C		R→P		P→C		C→S		S→P		R→S		P→R		Mean	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
Alexnet	S+T	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	55.8	58.7	40.0	43.4
	DANN	43.3	46.1	41.6	43.8	39.1	41.0	35.9	36.5	36.9	38.9	32.5	33.4	53.5	57.3	40.4	42.4
	ENT	37.0	45.5	35.6	42.6	26.8	40.4	18.9	31.1	15.1	29.6	18.0	29.6	52.2	60.0	29.1	39.8
	MME	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	60.7	44.2	48.2
	Meta-MME	-	56.4	-	50.2		51.9	-	39.6	-	43.7	-	38.7	-	60.7	-	48.8
	BiAT	54.2	58.6	49.2	50.6	44.0	52.0	37.7	41.9	39.6	42.1	37.2	42.0	56.9	58.8	45.5	49.4
	APE	47.7	54.6	49.0	50.5	46.9	52.1	38.5	42.6	38.5	42.2	33.8	38.7	57.5	61.4	44.6	48.9
	CDAC	**56.9**	**61.4**	**55.9**	**57.5**	**51.6**	**58.9**	**44.8**	**50.7**	**48.1**	**51.7**	**44.1**	**46.7**	**63.8**	**66.8**	**52.1**	**56.2**
Resnet34	S+T	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
	DANN	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
	ENT	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
	MME	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
	UODA	72.7	75.4	70.3	71.5	69.8	73.2	60.5	64.1	66.4	69.4	62.7	64.2	77.3	80.8	68.5	71.2
	Meta-MME	-	73.5	-	70.3	-	72.8	-	62.8	-	68.0	-	63.8	-	79.2	-	70.1
	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
	APE	70.4	76.6	70.8	72.1	72.9	76.7	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
	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**

Table 2. Accuracy(%) on *Office-Home* under the setting of 3-shot using Alexnet and Resnet34 as backbone networks.

Net	Method	R→C	R→P	R→A	P→R	P→C	P→A	A→P	A→C	A→R	C→R	C→A	C→P	Mean
Alexnet	S+T	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57.0	54.3	37.5	57.9	50.0
	DANN	47.2	66.7	46.6	58.1	44.4	36.1	57.2	39.8	56.6	54.3	38.6	57.9	50.3
	ENT	44.9	70.4	47.1	60.3	41.2	34.6	60.7	37.8	60.5	58.0	31.8	63.4	50.9
	MME	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
	Meta-MME	50.3	-	-	-	48.3	40.3	-	44.5	-	-	44.5	-	-
	BiAT	-	-	-	-	-	-	-	-	-	-	-	-	56.4
	APE	51.9	74.6	51.2	61.6	47.9	42.1	**65.5**	44.5	60.9	58.1	44.3	64.8	55.6
	CDAC	**54.9**	**75.8**	**51.8**	**64.3**	**51.3**	**43.6**	65.1	**47.5**	**63.1**	**63.0**	**44.9**	**65.6**	**56.8**
Resnet34	S+T	55.7	80.8	67.8	73.1	53.8	63.5	73.1	54.0	74.2	68.3	57.6	72.3	66.2
	DANN	57.3	75.5	65.2	69.2	51.8	56.6	68.3	54.7	73.8	67.1	55.1	67.5	63.5
	ENT	62.6	85.7	70.2	79.9	60.5	63.9	79.5	61.3	79.1	76.4	64.7	79.1	71.9
	MME	64.6	85.5	71.3	80.1	64.6	65.5	79.0	63.6	79.7	76.6	67.2	79.3	73.1
	Meta-MME	65.2	-	-	-	64.5	66.7	-	63.3	-	-	**67.5**	-	-
	APE	66.4	**86.2**	**73.4**	**82.0**	65.2	66.1	**81.1**	63.9	80.2	76.8	66.6	79.9	74.0
	CDAC	**67.8**	85.6	72.2	81.9	**67.0**	**67.5**	80.3	**65.9**	**80.6**	**80.2**	67.4	**81.4**	**74.2**

Table 3. Accuracy(%) on *Office* under the settings of 1-shot and 3-shot on the Alexnet backbone network.

Net	Method	W→A		D→A		Mean	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
Alexnet	S+T	50.4	61.2	50.0	62.4	50.2	61.8
	DANN	57.0	64.4	54.5	65.2	55.8	64.8
	ADR	50.2	61.2	50.9	61.4	50.6	61.3
	CDAN	50.4	60.3	48.5	61.4	49.5	60.8
	ENT	50.7	64.0	50.0	66.2	50.4	65.1
	MME	57.2	67.3	55.8	67.8	56.5	67.6
	BiAT	57.9	68.2	54.6	68.5	56.3	68.4
	APE	-	67.6	-	69.0	-	68.3
	CDAC	**63.4**	**70.1**	**62.8**	**70.0**	**63.1**	**70.0**

DSLR, Webcam and Amazon (abbr. **D**, **W** and **A**) with 31 classes. For fair comparisons, the settings of our benchmark datasets refer to the existing SSDA approaches [32, 29, 17], including adaptation scenarios of each dataset, the number of labeled target data (typically 1-shot or 3-shot per class), sample selection strategies, etc.

Implementation details: Similar to previous SSDA work [32, 15], we choose Alexnet and Resnet34 as our backbone networks. Firstly, the feature extractor is initialized with a pre-trained model on ImageNet[4] and the linear classification layer is initialized randomly, which has the same setting as [32, 29, 17, 15], such as architecture, output feature size, and so on. To balance multiple loss terms, we set λ in Equation (7) to 1.0 and ν in Equation (8) to 30.0. Also, we set the confidence threshold $\tau = 0.95$ in Equation (6). We implement our experiments on the widely-used PyTorch[5] platform. Additionally, in each iteration, we first train our model with the standard cross-entropy loss only on labeled data from both source and target domains and then add our proposed losses on unlabeled target data to further optimize the model. Furthermore, we introduce RandAugment [5] as the data augmen-

[4] http://www.image-net.org/

[5] https://pytorch.org/

Table 4. Accuracy(%) of CDAC using Resnet34 as the backbone on *DomainNet* under the setting of 3-shot. In the UDA setting, the supervised cross-entropy loss $\boldsymbol{L_{CE}}$ refers to the model trained only with labeled source samples.

Net	Setting	$\boldsymbol{L_{CE}}$	$\boldsymbol{L_{AAC}}$	$\boldsymbol{L_{PL}}$	$\boldsymbol{L_{Con}}$	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Mean
Resnet34	UDA	✓				57.8	61.4	58.1	53.4	58.2	49.6	72.3	58.6
	UDA	✓	✓			64.6	64.8	64.8	59.9	62.2	58.8	73.0	64.0
	UDA	✓		✓		68.0	73.2	68.3	61.8	67.0	63.1	76.5	68.2
	UDA	✓	✓	✓		76.9	73.9	73.9	66.2	70.2	69.0	79.3	72.8
	UDA	✓	✓	✓	✓	77.1	74.4	73.2	67.0	70.4	69.6	79.6	73.0
	SSDA	✓				60.0	62.2	59.4	55.0	59.5	50.1	73.9	60.0
	SSDA	✓	✓			69.4	68.1	68.3	62.8	65.6	62.0	76.9	67.6
	SSDA	✓		✓		76.7	73.6	76.3	66.9	70.3	69.3	80.4	73.4
	SSDA	✓	✓	✓		78.7	74.9	78.5	69.7	73.2	71.1	81.6	75.3
	SSDA	✓	✓	✓	✓	79.6	75.1	79.3	69.9	73.4	72.5	81.9	76.0

tation techniques used in this work. Finally, for fair comparisons, other experimental settings in our proposed CDAC, such as the optimizer, learning rate, mini-batch size, are the same as MME [32]. Our code is publicly available at https://github.com/lijichang/CVPR2021-SSDA.

Baselines: We compare CDAC with previous state-of-the-art SSDA approaches, including "**MME**" [32], "**UODA**" [29], "**BiAT**" [15], "**Meta-MME**" [21], "**APE**" [17], "**S+T**", "**DANN**" [9] and "**Ent**" [11]. Specifically, the model of the "S+T" method is trained using labeled source and target data only. In addition, "DANN" and "Ent" are both representative UDA methods and we re-train the models of "DANN" and "Ent" with an additional supervision loss by adding a few labeled target data.

4.2. Comparisons with the state-of-the-arts

Results on *DomainNet*, *Office-Home* and *Office* under the settings of 1-shot and 3-shot with Alexnet and Resnet34 as backbone networks are reported in Table 1, 2 and 3, respectively. As illustrated, our proposed CDAC significantly outperforms the state of the art throughout all experiments.

On DomainNet: As shown in Table 1, our CDAC significantly outperforms the existing approaches in all adaptation scenarios on *DomainNet*. Using Alexnet as the backbone, our method surpasses the existing best performing approach by 6.6% and 6.8% on average w.r.t the 1-shot and 3-shot settings respectively. Compared with the competing approaches using Resnet34 as the backbone, CDAC also achieves the best results in all cases and surpasses the current best results by 6% and 4.3% in the settings of 1-shot and 3-shot. Note that "MiCo" proposed in [42] is an unpublished work concurrent with ours, and the average performance of our CDAC using ResNet34 as the backbone is 0.4% higher than "MiCo" under the 3-shot setting.

On Office-Home and Office: To be consistent with the previous methods and achieve a fair comparison, we just employ Alexnet as the backbone on the *Office* benchmark. As shown in Table 2 and Table 3, our CDAC outperforms all comparison methods w.r.t mean accuracy on both datasets. In addition, it is worth noting that our method using Alexnet as the backbone achieves superior performance for most adaptation scenarios on *Office-Home*, and consistently achieves the best performance on *Office* w.r.t the adaptation scenarios of both "W→A" and "D→A".

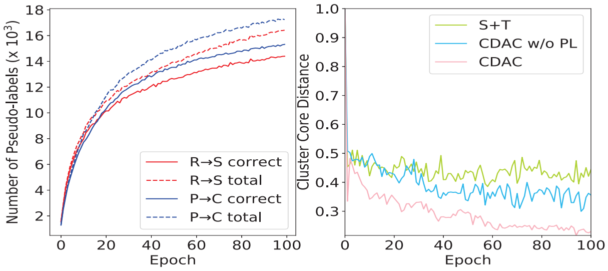

Figure 3. **Left:** The quantity and correctness of the proposed pseudo labeling technique on two adaptation scenarios of *DomainNet* (i.e., "R→S" and "P→C"), using Resnet34 as the backbone under the setting of 3-shot and 1-shot, respectively. **Right:** Variation of Cluster Core Distance among different approaches during model training while class "axe" is taken as an example.

4.3. Analysis

Ablation studies: We perform ablation studies on both SSDA and UDA settings to analyze the effectiveness of each loss term in our proposed CDAC, including $\boldsymbol{L_{CE}}$, $\boldsymbol{L_{AAC}}$, $\boldsymbol{L_{PL}}$ and $\boldsymbol{L_{Con}}$. All experiments are conducted on *DomainNet* using Resnet34 as the backbone under the 3-shot setting. As shown in Table 4, we regard the model trained with the cross-entropy loss $\boldsymbol{L_{CE}}$ only on labeled samples from both domains as the baseline for SSDA. And then, by combining both $\boldsymbol{L_{AAC}}$ and $\boldsymbol{L_{PL}}$ with $\boldsymbol{L_{CE}}$, the trained model achieves 25.3% higher average performance than the baseline, while the classification accuracy is on average 17.6% (+$\boldsymbol{L_{AAC}}$) or 23.4% (+$\boldsymbol{L_{PL}}$) higher than the baseline when only one of them is used together with the cross-entropy loss. Furthermore, the model trained with all loss functions reaches the best classification performance compared with the baseline. Moreover, each loss term proposed in our approach used for unlabeled target examples also shows similar roles in improving classification performance under the UDA setting.

Effectiveness of Adversarial Adaptive Clustering: To evaluate the effectiveness of the adversarial adaptive clustering loss, we refer to [37, 23] and employ Cluster Core Distance (CCD) to measure the distance between the source and target domain feature clusters within the same class. Generally speaking, the more aligned cross-domain feature clusters are, the smaller the CCDs are. We compare our CDAC model with "S+T" and "CDAC w/o PL" (a degraded version of CDAC, which is trained with only $\boldsymbol{L_{CE}}$ and $\boldsymbol{L_{AAC}}$). As shown in the right of Figure 3, it can be observed that the CCDs of all three methods decrease gradually during model training and it demonstrates that the source and target domain clusters within each class become closer. And both "CDAC w/o PL" and CDAC can result in better feature alignment than S+T. The CCD obtained from the model trained with CDAC finally converges to the minimum value, indicating that CDAC overall shows the best classification performance. This demonstrates the effectiveness of the proposed adversarial adaptive clustering loss in guiding the model towards learning better cluster-wise feature alignment.

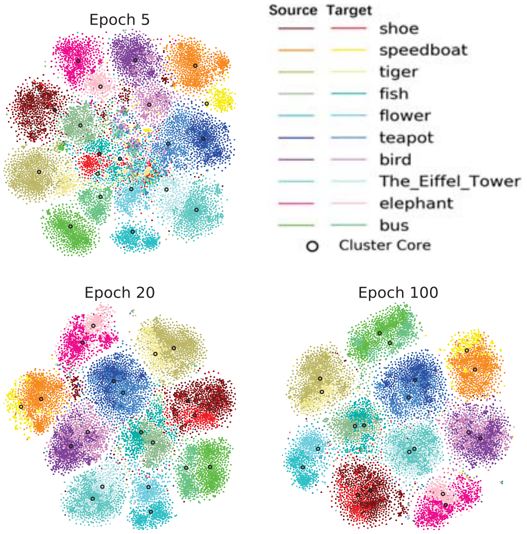

Figure 4. Visualization for feature distribution variations during model training with t-SNE. We choose 10 representative classes under the adaptation scenario "R→S" on *DomainNet* and their corresponding feature distributions from both source and target domains are displayed with different colors while black cycles represent cluster cores. We can observe that the final features have better cross-domain alignment than those at the beginning.

Effectiveness of Pseudo Labeling: The left subfigure in Figure 3 shows the quality and correctness of our proposed pseudo labeling technique in the model training process under two adaptation scenarios on *DomainNet* (i.e., "R→S" and "P→C" using Resnet34 as the backbone under the setting of 3-shot and 1-shot, respectively). It displays that a large proportion of unlabeled data is given correct pseudo-labels (up to 59.9% and 63.8% of total training examples per epoch at the best performance, respectively), which demonstrates the effectiveness of the proposed pseudo labeling technique in CDAC.

Feature visualization: We report with t-SNE [24] to display the gradual process of cluster-wise feature alignment during model training using the adaptation scenario "R→S" of *DomainNet* under the setting of 3-shot with Resnet34 as the backbone. As shown in Figure 4, we visualize the variations of the cluster and the corresponding cluster core of each class in the model training process. It can be observed that as the model optimization progresses, target features gradually converge towards target cluster cores, and each cluster in the target domain also gradually moves closer to their corresponding source cluster cores, showing a cluster-wise feature alignment effect. We take the "bus" class as an example. In Epoch 5, the feature distributions from both source and target domains are relatively far away. Then, as the model iterates, they gradually approach and finally achieve a perfect match at the last epoch.

5. Conclusions

We have presented a novel approach called Cross-domain Adaptive Clustering (CDAC) to solve the SSDA problem. CDAC consists of an adversarial adaptive clustering loss to guide the model training towards grouping the features of unlabeled target data into clusters and further performing cluster-wise feature alignment across domains. Furthermore, an adapted version of pseudo labeling is integrated into CDAC to enhance the robustness and power of cluster cores in the target domain to facilitate adversarial learning. Extensive experimental results, as well as ablation studies, have validated the virtue of our proposed method.

Acknowledgements

This work was supported in part by National Key Research and Development Program of China (No.2020YFC2003900), in part by the Guangdong Basic and Applied Basic Research Foundation (No.2020B1515020048) and in part by National Natural Science Foundation of China (No.61976250 and No.U1811463). This work was also supported by Meituan.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 3, 5

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019. 3

[3] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[4] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 627–636, 2019. 1

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. 2020 ieee. In *CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3008–3017, 2020. 6

[6] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9944–9953, 2019. 1

[7] Zhengyang Feng, Qianyu Zhou, Guangliang Cheng, Xin Tan, Jianping Shi, and Lizhuang Ma. Semi-supervised semantic segmentation via dynamic self-training and class-balanced curriculum. *arXiv preprint arXiv:2004.08514*, 2020. 3

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 5

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Franois Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2017. 7

[10] Joumana Ghosn and Yoshua Bengio. Bias learning, knowledge sharing. *IEEE Transactions on Neural Networks*, 14(4):748–765, 2003. 2

[11] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 3, 7

[12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 1

[13] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9101–9110, 2020. 1, 3

[14] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *International Conference on Learning Representations*, 2020. 4

[15] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence IJCAI-PRICAI-20*, 2020. 3, 4, 6, 7

[16] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 1

[17] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *16th European Conference on Computer Vision, ECCV 2020*. ECCV, 2020. 2, 3, 4, 6, 7

[18] Atsutoshi Kumagai and Tomoharu Iwata. Unsupervised domain adaptation by matching distributions based on the maximum mean discrepancy via unilateral transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4106–4113, 2019. 1

[19] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *Proc. International Conference on Learning Representations (ICLR)*, 2017. 5

[20] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013. 3

[21] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3, 4, 7

[22] 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. 1

[23] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019. 8

[24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 8

[25] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6670–6680, 2017. 1

[26] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020. 1, 2, 3

[27] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 2

[28] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 2, 5

[29] Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu. Opposite structure learning for semi-supervised domain adaptation. *arXiv preprint arXiv:2002.02545*, 2020. 3, 4, 6, 7

[30] Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Lsd-c: Linearly separable deep clusters. *arXiv preprint arXiv:2006.10039*, 2020. 4

[31] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 2, 5

[32] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8050–8058, 2019. 1, 2, 3, 4, 5, 6, 7

[33] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 2

[34] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. 1

[35] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33, 2020. 3, 5

[36] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *AAAI*, pages 5940–5947, 2020. 2

[37] Chaofan Tao, Fengmao Lv, Lixin Duan, and Min Wu. Minimax entropy network: Learning category-invariant features for domain adaptation. *arXiv preprint arXiv:1904.09601*, 2019. 8

[38] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[39] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. 2, 5

[40] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2517–2526, 2019. 2

[41] Si Wu, Jichang Li, Cheng Liu, Zhiwen Yu, and Hau-San Wong. Mutual learning of complementary networks via residual correction for improving semi-supervised classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6500–6509, 2019. 3

[42] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Mico: Mixup co-training for semi-supervised domain adaptation. *arXiv preprint arXiv:2007.12684*, 2020. 3, 4, 7

[43] Song-Bo Yang and Tian-Li Yu. Pseudo-representation labeling semi-supervised learning. *arXiv*, pages arXiv–2006, 2020. 3

[44] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, volume 31, pages 8559–8570. Curran Associates, Inc., 2018. 2