Enhancing Semi-supervised Domain Adaptation via Effective Target Labeling

Jiujun He[1], **Bin Liu**[1*], **Guosheng Yin**[2]

[1]Center of Statistical Research, School of Statistics, Southwestern University of Finance and Economics, Chengdu, China
[2]Department of Statistics and Actuarial Science, The University of Hong Kong, Hong Kong, China
tk9ng_@smail.swufe.edu.cn, liubin@swufe.edu.cn, gyin@hku.hk

Abstract

Existing semi-supervised domain adaptation (SSDA) models have exhibited impressive performance on the target domain by effectively utilizing few labeled target samples per class (e.g., 3 samples per class). To guarantee an equal number of labeled target samples for each class, however, they require domain experts to manually recognize a considerable amount of the unlabeled target data. Moreover, as the target samples are *not* equally informative for shaping the decision boundaries of the learning models, it is crucial to select the most informative target samples for labeling, which is, however, impossible for human selectors. As a remedy, we propose an EFfective Target Labeling (EFTL) framework that harnesses active learning and pseudo-labeling strategies to automatically select some informative target samples to annotate. Concretely, we introduce a novel sample query strategy, called non-maximal degree node suppression (NDNS), that iteratively performs maximal degree node query and non-maximal degree node removal to select representative and diverse target samples for labeling. To learn target-specific characteristics, we propose a novel pseudo-labeling strategy that attempts to label low-confidence target samples accurately via clustering consistency (CC), and then inject information of the model uncertainty into our query process. CC enhances the utilization of the annotation budget and increases the number of "labeled" target samples while requiring no additional manual effort. Our proposed EFTL framework can be easily coupled with existing SSDA models, showing significant improvements on three benchmarks.

Introduction

Domain adaptation (DA) aims to ease the cross-domain discrepancies, whereby transferring the knowledge learned from a label-rich source domain to a label-scarce but related target domain (Pan et al. 2010). Over the past decade, extensive research has been carried out in the unsupervised DA where the target domain is completely unlabeled (Long et al. 2013; Ganin and Lempitsky 2015). Despite advancements in these methods, the costless annotation scenarios yet making limited performance improvements on the target domain arise as a dilemma. In practice, it may be feasible to obtain a small amount of labeled target samples that can greatly facilitate the learning process. Hence, another branch of DA research falls into the semi-supervised learning setting and, more importantly, the semi-supervised DA (SSDA) approaches exhibit compelling performance improvements on the target domain when a small amount of labeled target data are provided in comparison with the unsupervised counterparts (Saito et al. 2019).

Existing works on SSDA (Saito et al. 2019; Yang et al. 2021a; Singh 2021; Li et al. 2021b) typically assume that the labeled target data are given, and each class has an equal number of labeled samples (e.g., 3 samples per class). However, this assumption is non-trivial in practice since labeling an equal number of target samples for each class, particularly for classification tasks with a large number of categories, requires a considerable amount of manual effort for domain experts, which is inconsistent with the idea of low annotation cost in DA. Moreover, the quality of the labeled target samples should also be guaranteed in order to maximize the utilization of the limited annotation expense. In practice, labeling target samples may produce worse performance (e.g., labeling outliers) or gain a very limited performance improvement (e.g., labeling "easy samples"). Unfortunately, these common but bad scenarios cannot be circumvented by human annotators. Therefore, it would be more reasonable that we design a learning framework that can automatically select a pre-defined number of target samples which, once labeled, can greatly facilitate the DA process.

Motivated by the above observations, we propose EFfective Target Labeling (EFTL), a DA framework that uses active learning and pseudo-labeling strategies to select some informative target samples for labeling. First, since only 1–3 samples per class are allowed for labeling in the SSDA setting, we aim to select *intra-class representative* and *inter-class diverse* target samples. To this end, we introduce a novel active strategy, called non-maximal degree node suppression (NDNS), to model sample's representativeness and diversity with graphs. Specifically, we first construct a directed graph by defining **a**ccepte**d** **n**eighbors (ADN) and **a**ccepti**e** **n**eighbors (AEN) for each unlabeled target sample. Benefiting from the definitions of ADN and AEN, those nodes with the locally maximal out-degree[1] can well repre-

[*]Corresponding Author

[1]The out-degree of a node indicates how many nodes are oriented by that node.

sent local clusters, and they should be selected for labeling. Hence, we iteratively query nodes with the maximal outdegree on the current sample graph and remove the k-th order neighbors of the queried node from the graph.

In addition, based on the fact that the "hard" samples are more informative as they contribute more to shaping the decision boundaries of the learning model (Xie et al. 2022b), we consider selecting target samples that are close to the decision boundaries for labeling. To this end, we aim to inject model uncertainty into our query process. Prior to sample selection, we observe that the target data tend to form some clear patterns in the feature space. Due to the domain discrepancy, however, these patterns are difficult to be recognized by the source-dominated learning model, resulting in clear but high-model uncertainty clusters. For more effective sample selection, we aim to detect the easy patterns for pseudo-labeling. Toward this goal, we propose a novel pseudo-labeling strategy called clustering consistency (CC). Unlike model-confidence-based pseudo-labeling, CC applies different clustering views to discover the intrinsic patterns of the target data and to decrease the noise of pseudo-labels. Empirical results demonstrate that the proposed pseudo-labeling strategy is capable of reducing noise and labeling low-confidence samples. CC increases the number of "labeled" target samples without additional manual effort.

As the proposed EFTL is a labeling framework, it can be easily coupled with existing SSDA approaches. Our main contributions are summarized as follows: 1) We present NDNS, a novel active strategy to select representative and diverse target samples for labeling. 2) We introduce a novel pseudo-labeling strategy that focuses on low-confidence samples and low-noise labels to increase the number of "labeled" target samples. 3) Our proposed approaches can be easily coupled with existing SSDA methods, showing significant performance improvements on three benchmarks. For example, EFTL enhances minimax entropy (MME) (Saito et al. 2019) with a 4% margin on the *DomainNet* benchmark, without introducing extra unsupervised techniques.

Related Work

Domain Adaptation

Unsupervised domain adaptation (UDA) has been well studied over the past decade. The UDA approaches generally focus on adversarial training (Ganin and Lempitsky 2015; Long et al. 2018; Zhang et al. 2019; Saito et al. 2018), statistical metrics minimizing (Long et al. 2015, 2017; Sun, Feng, and Saenko 2016; Chen et al. 2020) or intrinsic structure learning (Kang et al. 2019; Tang, Chen, and Jia 2020; Wu, Inkpen, and El-Roby 2020). Although the difference between SSDA and UDA is nuanced, Saito et al. (2019) claimed that the UDA methods do not perform well in the SSDA settings, where few labelled target samples per class are provided. Recently, SSDA has attracted enormous attention (Saito et al. 2019; Kim and Kim 2020; Jiang et al. 2020; Singh 2021; Li et al. 2021b; Yang et al. 2021a; Singh et al. 2021). For example, the minimax entropy (MME) approach is proposed to learn domain-invariant prototypes (Saito et al. 2019). With demonstration of the importance of the intra-domain discrepancy of SSDA, three strategies are proposed to alleviate this problem (Kim and Kim 2020). More recently, Yang et al. (2021a) explicitly split the SSDA task into two sub-tasks, i.e., semi-supervised learning (SSL) and unsupervised domain adaptation (UDA), and proposed to use the corresponding models (SSL and UDA) to teach each other. Some self-supervised training strategies, e.g., contrastive learning (Singh 2021) or pseudo labelling (Li et al. 2021a,b; Yan et al. 2022) have been incorporated into training. Observing an early misalignment phenomenon between the source and target data, Yu and Lin (2023) proposed to align the source data with the target data. Despite advancements in the aforementioned methods, they still require an equal number of the target samples to be given, which is a non-trivial requirement in practice.

Active Domain Adaptation

Active DA (ADA) aims to select target samples that, once labeled, will significantly facilitate domain adaptation (Prabhu et al. 2021). Over the past decade, many methods have been proposed for ADA (Rai et al. 2010; Saha et al. 2011; Long et al. 2013; Su et al. 2020; Singh et al. 2021; Prabhu et al. 2021; Xie et al. 2022a). Rai et al. (2010) proposed the so called DS-AODA that leverages a domain separator to select target domain-oriented instances. Recent works generally follow the criteria of *diversity* and *uncertainty* to select target samples. Su et al. (2020) utilized the targetness predicted by a domain discriminator and entropy to measure the *diversity* and *uncertainty*, respectively. Prabhu et al. (2021) grouped the target instances into some *diverse* clusters, where the entropy-based *uncertainty* is used to weigh samples. Singh et al. (2021) proposed STar, an approach for active SSDA, which uses the class-wise adaptation pace to measure the classifier adaptation *uncertainty* and groups the target data into more clusters by K-means for *diverse* selection. Moreover, Fu et al. (2021) harnessed committee consistency, committee margin, and targetness to unify a query function. Xie et al. (2022b) recently proposed a pure margin-based loss function and query function to select hard instances. However, Huang et al. (2023) discussed that most of the existing ADA methods have the potential limitation to generalize to other DA settings, e.g., SSDA scenarios. Indeed, our experiments confirm that some state-of-the-art ADA approaches, e.g., TQS (Fu et al. 2021) and SDM (Xie et al. 2022b), enhance existing SSDA models by a very limited margin and sometimes the performance may even deteriorate (e.g., in the 1-shot SSDA setting).

Method

Preliminary

A classifier $f : \mathcal{X} \mapsto \mathbb{R}^K$ receives a source domain dataset $D_{\mathcal{S}} = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled samples and a target domain dataset $D_{\mathcal{T}_u} = \{(x_i^t, \cdot)\}_{i=1}^{n_t}$ with n_t completely unlabeled samples, where $\mathcal{X}$ denotes the input space and K is the dimension of the output space. It is assumed that the source and target data are sampled from two different distributions P and Q, respectively, while they share the same

Algorithm 1: Overview of our proposed learning framework

```
Input: Unlabeled target data $D_{\mathcal{T}_u}$, labeled data $D_{\mathcal{L}} \leftarrow D_{\mathcal{S}}$,
    pretrained source model $f_s$, budget $\mathcal{B}$, query rounds $r$,
    learning rate $\eta$, query interval $\mathcal{I}$
Output: Parameters of the target learning model $f_t$
Split $\mathcal{B}$ evenly into $r$ sub-budgets $\mathcal{B} = \sum_{i=1}^{r} \mathcal{B}_i$
# pseudo-labeling strategy
Obtain pseudo-labeled samples $\hat{D}_{\mathcal{T}_l} = \{(x_i^t, \hat{y}_i^t)\}_{i=1}^{n_{pl}}$
    using CC algorithm; update: $D_{\mathcal{L}} \leftarrow D_{\mathcal{L}} \cup \hat{D}_{\mathcal{T}_l}$, $D_{\mathcal{T}_u} \leftarrow$
    $D_{\mathcal{T}_u} \backslash \hat{D}_{\mathcal{T}_l}$
Initialize a target learning model $f_t$ using the parameters
    of $f_s$, and a round counter $i \leftarrow 1$
for epoch $\leftarrow$ 0 to max_epoch do
    # active strategy
    if (epoch mod $\mathcal{I}$) == 0 and $\mathcal{B} > 0$ then
        Select $\mathcal{B}_i$ target samples from $D_{\mathcal{T}_u}$ and label them
            using Algorithm 2: $D_{\mathcal{T}_l} = \{(x_i^t, y_i^t)\}_{i=1}^{\mathcal{B}_i}$
        Update: $D_{\mathcal{L}} \leftarrow D_{\mathcal{L}} \cup D_{\mathcal{T}_l}$, $D_{\mathcal{T}_u} \leftarrow D_{\mathcal{T}_u} \backslash D_{\mathcal{T}_l}$
        Update: $\mathcal{B} \leftarrow \mathcal{B} - \mathcal{B}_i$, $i \leftarrow i + 1$
    end if
    # using existing SSDA approach
    Compute loss $\mathcal{L}(\theta_{f_t}, D_{\mathcal{T}_u}, D_{\mathcal{L}})$ using Eq. (5) or (6)
    Update $\theta_{f_t} \leftarrow \theta_{f_t} - \eta \nabla_{\theta_{f_t}} \mathcal{L}(\theta_{f_t}, D_{\mathcal{T}_u}, D_{\mathcal{L}})$
end for
return $\theta_{f_t}$
```

semantic label space $\mathcal{Y} = \{1, \dots, K\}$. We decompose the classifier f into the composite of a feature embedding function $F : \mathcal{X} \mapsto \mathbb{R}^m$ and a feature classifier $C : \mathbb{R}^m \mapsto \mathbb{R}^K$. That is, $f(x) = C(F(x))$, where F and C are parameterized by θ_F and θ_C, respectively.

Our goal is to select few unlabeled samples from $D_{\mathcal{T}_u}$ and label them to improve the model performance. The process of sample retrieval has a very limited annotation budget $\mathcal{B} = n_c \cdot K$, where n_c denotes the number of labeled samples for each class, e.g., $n_c = 3$ corresponds to 3-shot SSDA. The conventional SSDA approaches require n_c samples to be labeled per class (Saito et al. 2019; Li et al. 2021b), such that an expert needs to manually recognize a considerable amount of the unlabeled target data. To alleviate such burden, we propose an active learning DA framework which can automatically select $\mathcal{B}$ samples for labeling. In the paradigm of active learning, there usually involve r rounds of sample selection, such that each round may select about $\mathcal{B}/r$ target samples. The aim is to dynamically capture the informative target data as the training process proceeds. Prior to target sample selection, we first pretrain a source model $f_s(x) = C_s(F_s(x))$ on $D_{\mathcal{S}}$ and then use it to initialize a target learning model $f_t(x) = C_t(F_t(x))$. We summarize the proposed framework in Algorithm 1.

Active Strategy

To formulate our active strategy, non-maximal degree node suppression (NDNS), we follow the two widely used criteria in active learning methods to query target samples: intra-class representativeness and inter-class diversity. We first construct a directed graph using the target data and then iteratively retrieve the "significant" nodes with the locally maximal out-degree.

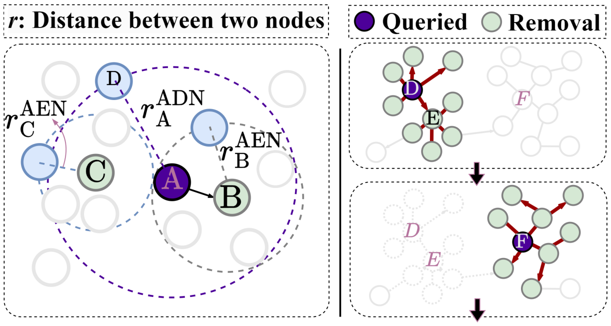

r: Distance between two nodes
● Queried ○ Removal
D
$r_{\mathrm{C}}^{\mathrm{AEN}}$
$r_{\mathrm{A}}^{\mathrm{ADN}}$
$r_{\mathrm{B}}^{\mathrm{AEN}}$
C
A
B
D
E
F
D
E
F

Figure 1: *Left panel:* An example of how we create the directed edges to construct a directed graph. In this case, $r_{\mathrm{A}}^{\mathrm{ADN}}$ is the distance between node A and its M_1-th nearest neighbor (node D); $r_{\mathrm{B}}^{\mathrm{AEN}}$ is the distance between node B and its M_2-th nearest neighbor and the same for $r_{\mathrm{C}}^{\mathrm{AEN}}$. Thus, node A may be accepted by those nodes inside the biggest dashed circle as their neighbors. Both nodes B and C satisfy this condition, but there is only a directed edge between node B and node A since node A is one of the AEN of node B. *Right panel:* An iteration of non-maximal degree node suppressio (NDNS). It first queries a node with maximal out-degree on the current graph, and then removes the k-order neighbor nodes of the queried node from the current graph ($k = 2$).

Directed graph construction. Let $\mathcal{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{n_t}]$ denote the feature embeddings of all unlabeled target data in $D_{\mathcal{T}_u}$, where $\mathbf{f}_i = F_t(x_i^t), i = 1, \dots, n_t$. Inspired by Yang et al. (2021b), we define two types of neighbors for each sample $x_i^t \in D_{\mathcal{T}_u}$,

$$\begin{aligned} \textbf{Accepte}\mathbf{d}\ \mathbf{N}\text{eighbors (ADN)} &: \mathcal{N}_{i\mathrm{ADN}}, \\ \text{Acceptiv}\mathbf{e}\ \mathbf{N}\text{eighbors (AEN)} &: \mathcal{N}_{i\mathrm{AEN}}, \end{aligned} \tag{1}$$

where $\mathcal{N}_{i*} = \mathrm{Topk}(\{\frac{\mathbf{f}_i^\top \mathbf{f}_j}{\|\mathbf{f}_i\| \cdot \|\mathbf{f}_j\|}, j = 1, \dots, n_t, j \neq i\})$, $* \in \{\mathrm{ADN}, \mathrm{AEN}\}$, and the $\mathrm{Topk}(\cdot)$ function returns a set of samples that have the K largest cosine similarity scores to x_i^t. We set $K = M_1$ for the accepted neighbors (ADN) and $K = M_2$ for the acceptive neighbors (AEN), and $M_2 \ll M_1$, then we have $\mathcal{N}_{i\mathrm{AEN}} \subset \mathcal{N}_{i\mathrm{ADN}}$. As shown by the left panel of Fig. 1, the principles underlying ADN and AEN are that sample i may be accepted by its ADN as their neighbors, and sample i only accepts its AEN as its neighbors; and that a large (small) value of M_1 (M_2) is suggested to create more edges with other nodes (decrease the number of noisy neighbors). Based on ADN and AEN, we define the adjacency matrix A of the directed graph as

$$A_{ij} = \begin{cases} 1, & j \in \mathcal{N}_{i\mathrm{ADN}} \text{ and } i \in \mathcal{N}_{j\mathrm{AEN}}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $i, j = 1, \dots, n_t$, and $A_{ij} = 1$ indicates there is a directed edge that orients from node i to node j. By summing the i-th row of A, we obtain the out-degree $d_i^- = \sum_{k=1}^{n_t} A_{ik}$ of node i, where d_i^- is node i's appearance frequency in the AEN of other nodes. For some node, AEN are its most affiliative neighbors since $M_2 \ll M_1$. If some node has a high

Algorithm 2: Non-maximal degree node suppression

```
Require: Current unlabeled target data $D_{\mathcal{T}_u}$, the current learning model $f_t$, annotation budget $\mathcal{B}_i$ in this round
Output: $\mathcal{B}_i$ labeled target samples
Initialize $\mathcal{Q} = \varnothing$ for storing the queried samples
Construct a directed graph using Eq. (2)
Compute $Q(x_i^t)$ for $x_i^t \in D_{\mathcal{T}_u}$ using Eq. (4)
while $|\mathcal{Q}| < \mathcal{B}_i$ do
    # node query
    Query a sample: $x_i^t \mapsto \arg\max_{i \in \{1,\dots,n_t\}} Q(x_i^t)$
    Append $x_i^t$ to $\mathcal{Q}$
    # node suppression
    Mask $Q(x_j^t) = -\infty$ for all $x_j^t$ in the $k$-order neighbors of the queried sample $x_i^t$.
end while
return $\mathcal{Q}$
```

out-degree, intuitively, it would be a "center point" for a local cluster. Hence, querying and labeling those nodes with the locally maximal out-degree can gain more information, which naturally motivates the design of NDNS.

Non-maximal degree node suppression. In fact, NDNS shares a similar idea with the non-maximum suppression (NMS) (Neubeck and Van Gool 2006). To avoid querying redundant nodes, as shown in the right panel of Fig. 1, we should remove those nodes which are close to the queried nodes (removal nodes in Fig. 1). As a result, the NDNS iteratively performs two steps—node query and node removal—to select informative and diverse samples to label, till reaching the prespecified budget in a round. **Node query:** we select node i with the maximal out-degree on the current graph (i.e., $\arg\max_i \; d_i^- = \sum_{k=1}^{n_t} A_{ik}$). **Node removal:** we remove node i and its k-order neighbor nodes from the current graph (k is fixed to be 2 for all experiments). For the succinctness of implementation, we directly set the j-th row and the j-th column of adjacency matrix A to be zero for each node j to be removed, which leads to each objective node being isolated on the current graph. For the subsequent query rounds, the 1-order neighbors of the selected target samples in the previous query rounds are marked to be isolate nodes, which further guarantees the diversity of queried samples.

Low-Confidence Target Labeling

As the "hard" samples are important for shaping the decision boundaries of f_t, we intend to select target samples that are close to the decision boundaries to annotate.

Model uncertainty injection. We first define the probabilistic margin m_i of the sample x_i^t yielded by f_t:

$$m_i = p_{1_*}(x_i^t) - p_{2_*}(x_i^t), \tag{3}$$

where $p_k(\cdot) = \sigma_k(C_t(F_t(\cdot)))$ denotes the prediction probability of the k-th class, $\sigma(\cdot)$ is the softmax function, and 1_* and 2_* represent the indices of the first and second largest elements of a vector, respectively. Hence, a lower margin implies an ambiguous prediction. Because labeling ambiguous target samples can greatly reshape the decision boundaries of the source-dominated model, it is natural to combine NDNS with model uncertainty to select more effective target samples. Thus, we modify the query criteria of NDNS as

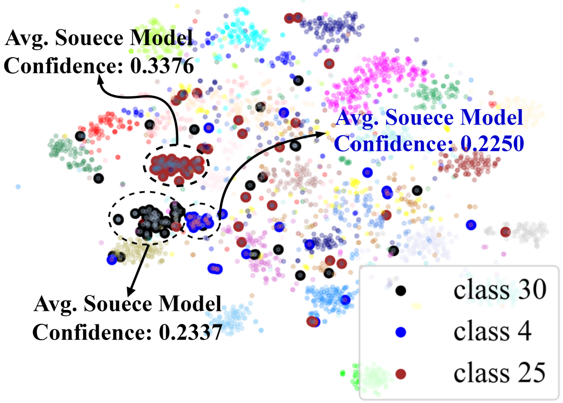
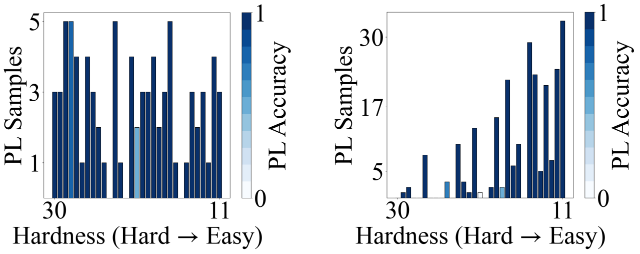


Figure 2: t-SNE visualization of the target features extracted by the source model F_s. Some target samples belonging to classes 4, 25, and 30 form clear patterns, but all of them have very low model confidence, resulting in clear but hard-to-recognize (by the source classifier C_s) patterns.

$$Q(x_i^t) = \alpha \hat{d}_i^- + (1 - m_i), \tag{4}$$

where $\hat{d}_i^- = d_i^- / \max_j \, (d_j^-)$ indicates the max-normalized out-degree and α is a trade-off factor. Hence, the *node query* step of NDNS is $i \mapsto \arg\max_i \; Q(x_i^t)$, and the rest of the procedures of NDNS remain the same. We summarize the active strategy in Algorithm 2.

Clustering consistency (CC) pseudo-labeling. We observe in Fig. 2 that there are some clear clusters with very low model confidence, in which some data are easy to be selected by NDNS for labeling because they are representative, diverse, and low-confidence. If these easy structures of the target data are captured, we can use a pseudo-labeling (PL) strategy for labeling so that the labeling budget $\mathcal{B}$ is not wasted. Hence, we utilize clustering algorithms to discover these patterns and leverage the consistency of different clustering views to filter out the noisy labels. Specifically, we implement the clustering algorithms in the embedding space $\mathbb{R}^m$, i.e., each sample is represented by $\mathbf{f}_i = F_s(x_i^t)$. We apply the clustering algorithm to obtain the class probability distribution for each sample. Let $P^1 = \{p_{ik}^1\}$ and $P^2 = \{p_{ik}^2\}$ denote the probabilistic matrices obtained by the Gaussian mixture model clustering and K-means clustering, respectively, where $i = 1, \dots, n_t$, $k = 1, \dots, K$, $\sum_{k=1}^{K} p_{ik}^* = 1$, p_{ik}^* denotes the probability of the i-th samples belonging to the k-th class, and $* \in \{1, 2\}$. For each class k, if $x_i^t \in \mathrm{Topk}(\{p_{ik}^1, i = 1, \dots, n_t\})$ and $x_i^t \in \mathrm{Topk}(\{p_{ik}^2, i = 1, \dots, n_t\})$, then sample x_i^t is labeled as class k, where $\mathrm{Topk}(\{p_{ik}^*, i = 1, \dots, n_t\})$ returns K samples with the largest prediction probability to class k. To avoid introducing additional hyperparameters, we fix K to



(a) CC

(b) CBPL

Figure 3: Visualization of PL samples obtained by the CC and CBPL algorithms (transfer task **W** → **A** on the Office-31 dataset), using ResNet-34 as the backbone. The y-axis denotes the number of pseudo-labeled samples for each class, and the x-axis represents the hardness of each class, i.e., class 30 and class 11 have the lowest and highest average model confidence, respectively. The depth of the color represents the accuracy of PL samples of each class.

be M_2 for all experiments. Let $\hat{D}_{\mathcal{T}_l} = \{x_i^t, \hat{y}_i^t\}_{i=1}^{n_{pl}}$ denote the pseudo-labeled data. Then, $\hat{D}_{\mathcal{T}_l}$ will be appended to the labeled dataset $D_{\mathcal{L}}$ and deleted from the unlabeled dataset $D_{\mathcal{T}_u}$ (see step 3 in Algorithm 1).

Fig. 3 shows the number of pseudo-labeled samples for each class, accuracy of pseudo-labels, and hardness of each class. The source model is trained on the source domain Webcam (**W**), and it will be employed to extract target features, i.e., $\mathbf{f}_i = F_s(x_i^t)$, to carry out the clustering consistency pseudo-labeling. The target domain Amazon (**A**) contains 2,817 samples and a total of 31 categories in our experiments. We return $M_2 = 6$ samples for each class for pseudo-labeling. Then, the CC strategy is applied to adaptively filter out inconsistent samples, which leads to 0–6 samples being labeled for each class. We also make a comparison with the widely used confidence-based pseudo-labeling (CBPL) like in Fixmatch (Sohn et al. 2020) (we set threshold=0.9).

As shown in Fig. 3 (b), CBPL exhibits a skewed distribution of pseudo-labels, i.e., the easy classes are always at an advantage. Due to the domain discrepancy, CBPL cannot mitigate it but may further aggravate the class imbalance problem. In contrast, CC not only circumvents this issue to some extent, but also shows highly accurate pseudo-labels, as shown in Fig. 3 (a).

Coupled with Existing SSDA Methods

After obtaining the labeled target data, we can immediately use current SSDA approaches to perform domain adaptation. Here, we consider two widely used baselines.

Minimax Entropy (MME). MME (Saito et al. 2019) is a pioneering work for the SSDA task, and its training objective is defined as follows:

$$\begin{aligned} \hat{\theta}_{C_t} &= \arg\min_{\theta_{C_t}} \mathcal{L}_{\mathrm{CE}} - \lambda \mathcal{L}_{\mathrm{ENT}}, \\ \hat{\theta}_{F_t} &= \arg\min_{\theta_{F_t}} \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{ENT}}, \end{aligned} \tag{5}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss computed on the labeled data, $\mathcal{L}_{\mathrm{ENT}}$ is the entropy loss computed on unlabeled target data, and λ is a trade-off hyperparameter fixed to be 0.1 (Saito et al. 2019).

	ADA Method	SSDA Method	W→A		D→A	
			1-shot	3-shot	1-shot	3-shot
SSDA Only	✗	S+T	69.2	73.2	68.2	73.3
	✗	MME	73.1	76.3	73.6	77.6
	✗	DECOTA	76.0	76.8	74.2	78.3
	✗	FixMME	75.3	78.7	76.8	78.5
ADA Only	TQS	✗	65.9	74.9	67.4	74.9
	SDM-A	✗	66.7	75.4	67.8	75.1
	EFTL (ours)	✗	**72.8**	**77.3**	**69.9**	**76.0**
ADA + SSDA	TQS	MME	70.3	75.9	69.4	73.2
	SDM-A	MME	70.1	76.8	69.2	73.9
	EFTL (ours)	MME	**78.6**	**81.0**	**78.2**	**82.3**
ADA + SSDA	TQS	FixMME	71.3	74.7	72.0	76.9
	SDM-A	FixMME	71.2	77.0	71.8	77.1
	EFTL (ours)	FixMME	**79.9**	**83.3**	**78.1**	**81.1**

Table 1: Compare SSDA with active domain adaptation (ADA) methods on *Office-31* benchmark using ResNet-34 as the backbone, where the loss of *ADA Only* approaches only allowed for computing on the labeled source and target data, the labeled target data used in *SSDA Only* methods are given by MME, and *ADA+SSDA* indicates the labeled target data of SSDA methods are selected by ADA methods.

FixMatch + MME (FixMME). Fixmatch (Sohn et al. 2020) is a strong baseline for semi-supervised learning, which is also widely used in SSDA (Li et al. 2021a; Yan et al. 2022; Li et al. 2021b). We combine FixMatch with MME as a strong baseline:

$$(\hat{\theta}_{C_t}, \hat{\theta}_{F_t}) = \arg\min_{(\theta_{C_t}, \theta_{F_t})} \mathcal{L}_{\mathrm{MME}} + \mathcal{L}_{\mathrm{PL}}, \tag{6}$$

where $\mathcal{L}_{\mathrm{MME}}$ is the MME loss defined in Eq. (5), and $\mathcal{L}_{\mathrm{PL}}$ is the pseudo-labeling loss defined as:

$$\mathcal{L}_{\mathrm{PL}} = -\mathbb{E}_{x \sim D_{\mathcal{T}_u}} \mathbb{I}\{p_{\hat{y}}(x) > \tau\} \log(p_{\hat{y}}(\tilde{x})), \tag{7}$$

where $p_k(\cdot) = \sigma_k(C_t(F_t(\cdot)))$ is defined in Eq. (3), $\hat{y} = \arg\max_{y \in \mathcal{Y}} p_y(x)$, $\tilde{x}$ is a strongly augmented version of x, τ is a probability threshold, and $\mathbb{I}(\cdot)$ is the 0-1 loss. The objective of $\mathcal{L}_{\mathrm{PL}}$ is clear: encourage the model to make consistent prediction on samples with a high probability.

Experiments

Experimental settings. We conduct thorough evaluations on three image-based benchmark datasets: *Office-31* (Saenko et al. 2010), *Office-Home* (Venkateswara et al. 2017), and *DomainNet* (Peng et al. 2019). Since the numbers of the AEN and ADN in the directed graph construction depend on the scale of the dataset, we choose $M_1 = \lfloor n_t/(3 \times K) \rfloor$ and $M_2 = \lfloor M_1/5 \rfloor$ to adaptively construct the directed graph across all datasets. The trade-off hyperparameter α in Eq. (4) is fixed as 0.1. For baseline FixMME, the threshold τ in Eq. (7) is set to be 0.85 for all datasets except

Setting	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	Avg.
1-shot	S+T (baseline)	52.1	78.6	66.2	74.4	48.3	57.2	69.8	50.9	73.8	70.0	56.3	68.1	63.8
	ENT	53.6	81.9	70.4	79.9	51.9	63.0	75.0	52.9	76.7	73.2	63.2	73.6	67.9
	CDAC	61.9	83.1	72.7	80.0	59.3	64.6	75.9	61.2	78.5	75.3	64.5	75.1	71.0
	DECOTA	56.0	79.4	71.3	76.9	48.8	60.0	68.5	42.1	72.6	70.7	60.3	70.4	64.8
	MME + SLA	64.1	83.8	72.9	80.0	59.9	66.7	76.3	62.1	78.6	75.1	67.5	77.1	72.0
	MCL	**67.0**	85.5	73.8	81.3	61.1	68.0	79.5	64.4	**81.2**	**78.4**	**68.5**	79.3	74.0
	MME	61.9	82.8	71.2	79.2	57.4	64.7	75.5	59.6	77.8	74.8	65.7	74.5	70.4
	+ EFTL (ours)	66.3	85.9	73.8	80.7	60.6	66.7	80.1	63.1	80.1	75.7	63.5	79.0	73.0 (+2.6)
	FixMME	64.7	83.6	72.0	80.3	59.0	65.5	76.0	64.0	78.7	75.5	65.3	75.2	71.7
	+ EFTL (ours)	66.6	**87.2**	**74.3**	**82.6**	**63.3**	**68.7**	**80.5**	**65.7**	80.8	77.5	65.6	**79.6**	**74.4** (+2.7)
3-shot	S+T (baseline)	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
	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
	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
	DECOTA	70.4	87.7	74.0	82.1	68.0	69.9	81.8	64.0	80.5	79.0	68.0	83.2	75.7
	MME + SLA	68.4	87.4	74.7	81.9	67.4	69.7	81.1	65.9	80.5	79.4	69.2	81.9	75.6
	MCL	70.1	88.1	75.3	83.0	68.0	69.9	83.9	67.5	82.4	**81.6**	**71.4**	84.3	77.1
	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
	+ EFTL (ours)	70.8	88.5	76.4	83.3	67.1	71.5	83.3	68.2	82.9	78.6	68.8	82.5	76.8 (+3.7)
	FixMME	69.0	88.0	72.3	81.3	66.6	69.1	81.5	65.8	81.0	78.9	67.2	81.4	75.2
	+ EFTL (ours)	**72.8**	**89.3**	**77.5**	**85.4**	**70.9**	**72.6**	**84.8**	**70.3**	**83.8**	81.5	70.6	**84.6**	**78.7** (+3.5)

Table 2: Classification accuracy (%) on the *Office-Home* benchmark, using ResNet-34 as the backbone.

Method	R→C		R→P		P→C		C→S		S→P		R→S		P→R		Avg.	
	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
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
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
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
DECOTA	79.1	80.4	**74.9**	75.2	76.9	78.7	65.1	68.6	72.0	72.7	69.7	71.9	79.6	81.5	73.9	75.6
MME + SLA	71.8	73.3	68.2	70.1	70.4	72.7	59.3	63.4	64.9	67.3	61.8	63.9	77.2	79.6	68.8	70.0
MCL	77.4	79.4	74.6	76.3	75.5	78.8	66.4	70.9	**74.0**	**74.7**	**70.7**	**72.3**	82.0	83.3	74.4	76.5
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
+ EFTL (ours)	74.1	77.4	70.1	72.8	74.0	77.0	65.0	68.1	67.3	69.5	64.7	67.6	78.0	80.5	70.5 (+4.1)	73.3 (+4.4)
FixMME	74.5	78.1	72.6	74.1	74.3	77.2	65.3	68.2	70.3	72.3	67.5	68.3	80.4	82.5	72.1	74.4
+ EFTL (ours)	**79.6**	**81.2**	**74.9**	**77.1**	**78.2**	**81.8**	**69.3**	**72.8**	71.8	74.4	69.9	71.5	**83.1**	**84.4**	**75.3** (+3.2)	**77.6** (+3.2)

Table 3: Classification accuracy (%) on the *DomainNet* benchmark, using ResNet-34 as the backbone.

	NDNS	CC	A→C		C→P	
			1-shot	3-shot	1-shot	3-shot
FixMME			64.0	65.8	75.2	81.4
	✓		64.0	69.9	78.7	**84.6**
	✓	✓	**65.7**	**70.3**	**79.6**	**84.6**
MME			59.6	63.6	74.5	79.3
	✓		62.9	67.9	78.6	82.1
	✓	✓	**63.1**	**68.2**	**79.0**	**82.5**

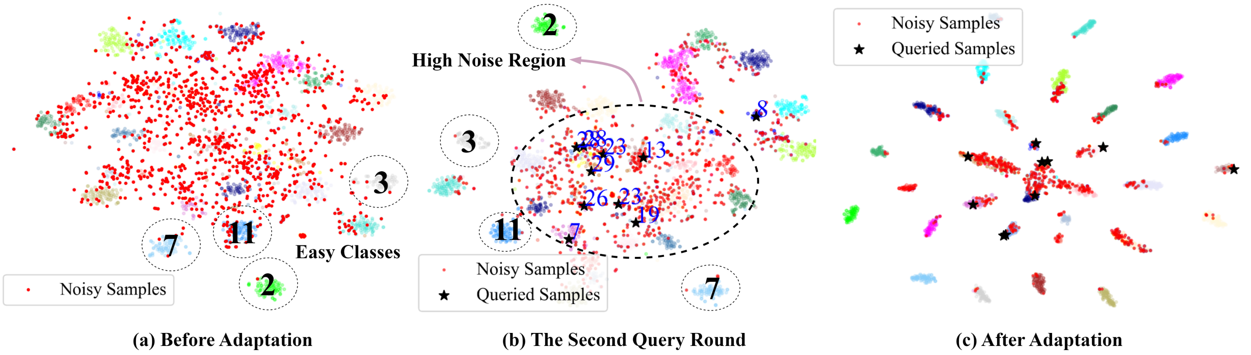
Table 4: Ablation study on the *Office-Home* benchmark.

0.8 for *DomainNet*. Following Li et al. (2021b), we exploit a label smoothing technique with parameter 0.1 to avoid overconfident predictions when using a cross-entropy loss. We run our experiments three times with different random seeds independently. For more details, please refer to our code: https://github.com/BetterTMrR/EFTL-Pytorch-main.

Results and Analysis

This section aims to answer the following two questions: (1) Can other labeling frameworks, i.e., active domain adaptation (ADA), be effectively combined with SSDA? (2) How effective is the proposed EFTL?

Compare existing ADA methods with SSDA models. We choose two recent published baselines for the ADA task: **TQS** (Fu et al. 2021) and **SDM-A** (Xie et al. 2022a) and four baselines for the SSDA task: **S+T**, **MME**, **DECOTA**, and **FixMME**. The results are reported in Table 1, from which we can draw the following conclusions. First, both **TQS** and **SDM-A** are originally designed for the ADA task and cannot be coupled well with SSDA models as expected. In the ADA+SSDA setting, they show worse performance than SSDA only setting (e.g., MME > MME+TQS). In contrast, our proposed EFTL significantly enhances the performance of SSDA methods. In addition, EFTL can still outperform

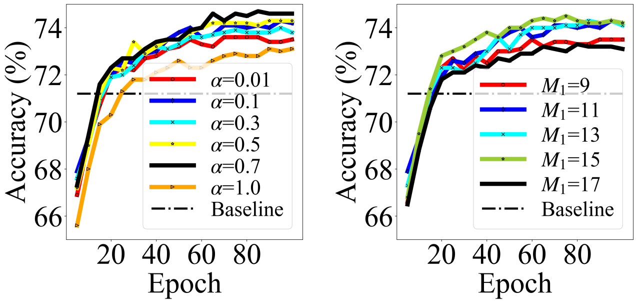


(a) Before Adaptation

(b) The Second Query Round

(c) After Adaptation

Figure 4: t-SNE visualization of the target features (1-shot transfer task W → A on *Office-31* benchmark and MME is employed), where each colored point indicates a target sample, different classes are displayed in different colors, and the red points represent that they are misclassified by the corresponding model. (a): The target features from the source model, where the numbers inside the circle indicate the class indices. (b): The target features from the target model at the second query round, where the notation ★ denotes the queried samples in the 1-2 query rounds (the number indicates its label). (c): The target features from the model after adaptation.



(a) Sensitivity to α

(b) Sensitivity to M_1

Figure 5: Sensitivity analysis of hyperparameters on the 1-shot transfer task R → A on the *Office-Home* benchmark. The results are obtained by varying the corresponding hyperparameter while fixing others.

ADA baselines in the ADA only setting.

Effectiveness of the proposed EFTL. We choose state-of-the-art baselines for comparisons: **S+T** (model trained on the labeled source and target data only), **ENT** (Grandvalet and Bengio 2004), **MME** (Saito et al. 2019), **CDAC** (Li et al. 2021a), **DECOTA** (Yang et al. 2021a), **MCL** (Yan et al. 2022), **MME+SLA** (Yu and Lin 2023), and **FixMME**. The results are reported in Tables 2 and 3. We can observe that EFTL significantly improves the performances of MME and FixMME. In the *DomainNet* benchmark, EFTL improves MME with a margin over 4% and FixMME with a margin over 3%. Notably, although the strong SSDA baselines achieve impressive performance on the target domain by using tedious unsupervised training techniques (e.g., FixMatch loss, inter and intra-domain losses in MCL), FixMME+EFTL can still outperform them on the most transfer tasks, demonstrating the importance of the quality of labeled data.

Ablation study. The results of ablation study are displayed in Table 4. When the active strategy NDNS is not applied, the labeled and unlabeled target data split is given by Saito et al. (2019). We observe that the active strategy NDNS significantly enhances the model performance, and the CC pseudo-labeling strategy can further improve the results.

Sensitivity analysis. Because the hyperparameter M_2 changes with M_1, we only show the sensitivity of M_1 and α in Eq. (4). According to the formula defined in the **Experimental settings** section, we obtain $M_1 = 11$ for the transfer task R → A in Table 2. We vary M_1 from 9 to 17 to observe the performance variations. The results of sensitivity to hyperparameters are displayed in Fig. 5, where we visualize the different learning processes of MME by varying the corresponding hyperparameters. Except for the extreme case (e.g., $\alpha = 1.0$ and $M_1 = 17$), we find that the model performance is *not* sensitive to M_1 or α. Moreover, under such a wide range of hyperparameters, they always outperform the baseline (i.e., without using an active strategy).

Feature visualization of query process. We visualize the query process of NDNS in Fig. 4. In Fig. 4 (a), some easy classes (e.g., classes 2, 3, 7, and 11) form clear clusters in the feature space of the source model. During the query process, as shown in Fig. 4 (b), NDNS mainly focuses on the high-noise region where the decision boundaries are not clear, enhancing the utilization of the labeled data. After being trained on the queried samples, our model yields clear decision boundaries, as shown in Fig. 4 (c).

Conclusion

In this work, we propose an effective target labeling framework that harnesses active learning and pseudo-labeling strategies to enhance the performance of the target learning model. To that end, we propose an active approach, NDNS, which aims to query the intra-class representative, inter-class diverse, and low-confidence target samples for labeling. Furthermore, we introduce a novel pseudo-labeling strategy that focuses on low-confidence samples and low-noise labels to increase the number of “labeled” target samples. Extensive experiments on three benchmarks demonstrate the effectiveness of our proposed approach.

Acknowledgements

The research of Guosheng Yin was partially supported by the Theme-based Research Scheme (TRS) from the Research Grants Council of Hong Kong, Institute of Medical Intelligence and XR (T45-401/22-N).

References

Chen, C.; Fu, Z.; Chen, Z.; Jin, S.; Cheng, Z.; Jin, X.; and Hua, X. 2020. HoMM: Higher-Order Moment Matching for Unsupervised Domain Adaptation. In *AAAI*, 3422–3429. AAAI Press.

Fu, B.; Cao, Z.; Wang, J.; and Long, M. 2021. Transferable query selection for active domain adaptation. In *CVPR*, 7272–7281.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, 1180–1189. PMLR.

Grandvalet, Y.; and Bengio, Y. 2004. Semi-supervised learning by entropy minimization. *NeurIPS*, 17.

Huang, D.; Li, J.; Chen, W.; Huang, J.; Chai, Z.; and Li, G. 2023. Divide and Adapt: Active Domain Adaptation via Customized Learning. In *CVPR*, 7651–7660.

Jiang, P.; Wu, A.; Han, Y.; Shao, Y.; Qi, M.; and Li, B. 2020. Bidirectional Adversarial Training for Semi-Supervised Domain Adaptation. In *IJCAI*, 934–940.

Kang, G.; Jiang, L.; Yang, Y.; and Hauptmann, A. G. 2019. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 4893–4902.

Kim, T.; and Kim, C. 2020. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV*, 591–607. Springer.

Li, J.; Li, G.; Shi, Y.; and Yu, Y. 2021a. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *CVPR*, 2505–2514.

Li, K.; Liu, C.; Zhao, H.; Zhang, Y.; and Fu, Y. 2021b. ECACL: A holistic framework for semi-supervised domain adaptation. In *ICCV*, 8578–8587.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *ICML*, 97–105. PMLR.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. *NeurIPS*, 31.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2013. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2200–2207.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep transfer learning with joint adaptation networks. In *ICML*, 2208–2217. PMLR.

Neubeck, A.; and Van Gool, L. 2006. Efficient non-maximum suppression. In *ICPR*, volume 3, 850–855. IEEE.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2010. Domain adaptation via transfer component analysis. *TNN*, 22(2): 199–210.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *ICCV*, 1406–1415.

Prabhu, V.; Chandrasekaran, A.; Saenko, K.; and Hoffman, J. 2021. Active domain adaptation via clustering uncertainty-weighted embeddings. In *ICCV*, 8505–8514.

Rai, P.; Saha, A.; Daumé III, H.; and Venkatasubramanian, S. 2010. Domain adaptation meets active learning. In *Workshop on ALNLP*, 27–32.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *ECCV*, 213–226. Springer.

Saha, A.; Rai, P.; Daumé, H.; Venkatasubramanian, S.; and DuVall, S. L. 2011. Active supervised domain adaptation. In *MLKDD*, 97–112. Springer.

Saito, K.; Kim, D.; Sclaroff, S.; Darrell, T.; and Saenko, K. 2019. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 8050–8058.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 3723–3732.

Singh, A. 2021. CLDA: Contrastive Learning for Semi-Supervised Domain Adaptation. *NeurIPS*, 34.

Singh, A.; Doraiswamy, N.; Takamuku, S.; Bhalerao, M.; Dutta, T.; Biswas, S.; Chepuri, A.; Vengatesan, B.; and Natori, N. 2021. Improving semi-supervised domain adaptation using effective target selection and semantics. In *CVPR*, 2709–2718.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 33: 596–608.

Su, J.-C.; Tsai, Y.-H.; Sohn, K.; Liu, B.; Maji, S.; and Chandraker, M. 2020. Active adversarial domain adaptation. In *WACV*, 739–748.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of Frustratingly Easy Domain Adaptation. In Schuurmans, D.; and Wellman, M. P., eds., *AAAI*, 2058–2065. AAAI Press.

Tang, H.; Chen, K.; and Jia, K. 2020. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, 8725–8735.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 5018–5027.

Wu, Y.; Inkpen, D.; and El-Roby, A. 2020. Dual mixup regularized learning for adversarial domain adaptation. In *ECCV*, 540–555. Springer.

Xie, B.; Yuan, L.; Li, S.; Liu, C. H.; Cheng, X.; and Wang, G. 2022a. Active learning for domain adaptation: An energy-based approach. In *AAAI*, volume 36, 8708–8716.

Xie, M.; Li, Y.; Wang, Y.; Luo, Z.; Gan, Z.; Sun, Z.; Chi, M.; Wang, C.; and Wang, P. 2022b. Learning Distinctive Margin toward Active Domain Adaptation. In *CVPR*, 7993–8002.

Yan, Z.; Wu, Y.; Li, G.; Qin, Y.; Han, X.; and Cui, S. 2022. Multi-level Consistency Learning for Semi-supervised Domain Adaptation. In Raedt, L. D., ed., *IJCAI*, 1530–1536. ijcai.org.

Yang, L.; Wang, Y.; Gao, M.; Shrivastava, A.; Weinberger, K. Q.; Chao, W.-L.; and Lim, S.-N. 2021a. Deep co-training with task decomposition for semi-supervised domain adaptation. In *ICCV*, 8906–8916.

Yang, S.; van de Weijer, J.; Herranz, L.; Jui, S.; et al. 2021b. Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation. *NeurIPS*, 34.

Yu, Y.-C.; and Lin, H.-T. 2023. Semi-Supervised Domain Adaptation with Source Label Adaptation. In *CVPR*, 24100–24109.

Zhang, Y.; Liu, T.; Long, M.; and Jordan, M. 2019. Bridging theory and algorithm for domain adaptation. In *ICML*, 7404–7413. PMLR.