Combating Label Distribution Shift for Active Domain Adaptation

Sehyun Hwang[1]   Sohyun Lee[2]   Sungyeon Kim[1]
Jungseul Ok[1,2*]   Suha Kwak[1,2*]

[1]Department of Computer Science and Engineering, POSTECH, Korea
[2]Graduate School of Artificial Intelligence, POSTECH, Korea
http://cvlab.postech.ac.kr/research/LAMDA/

Abstract. We consider the problem of active domain adaptation (ADA) to unlabeled target data, of which subset is actively selected and labeled given a budget constraint. Inspired by recent analysis on a critical issue from label distribution mismatch between source and target in domain adaptation, we devise a method that addresses the issue for the first time in ADA. At its heart lies a novel sampling strategy, which seeks target data that best approximate the entire target distribution as well as being representative, diverse, and uncertain. The sampled target data are then used not only for supervised learning but also for matching label distributions of source and target domains, leading to remarkable performance improvement. On four public benchmarks, our method substantially outperforms existing methods in every adaptation scenario.

Keywords: active domain adaptation, active learning, domain adaptation, label distribution shift

1   Introduction

Domain adaptation is the task of adapting a model trained on a label-sufficient source domain to a label-scarce target domain when their input distributions are different. It has played crucial roles in applications that involve significant input distribution shifts such as recognition under adverse conditions (*e.g.*, climate changes [10,25,45,46] and nighttime [44]) and synthetic-to-real adaptation [35]. The most popular direction in this field is unsupervised domain adaptation [4,15] which assumes a totally unlabeled target domain. However, in practice, labeling a small part of target data is usually feasible. Hence, label-efficient domain adaptation tasks such as semi-supervised domain adaptation [26,27,43,61,62] and active domain adaptation [14,36,40,52] have attracted increasing attention.

In this paper, we consider active domain adaptation (ADA) [14,36,40,52], where we can interact with an oracle to obtain annotations on a subset of target data given budget constraint, while utilizing the annotations for domain adaptation. The key to the success of ADA is to co-design sampling mechanism selecting

---

* Co-corresponding authors

a subset of target data to be annotated and utilization of the annotations. Existing ADA methods utilize the obtained annotations only for supervised learning, similar to existing Active Learning (AL) methods [3, 47, 51, 56]. Accordingly, they count diversity, representativeness, and uncertainty of the data to boost the effect of supervised learning.

We argue that for domain adaptation, there is another use of the sampled data, which deserves attention but is missing in the previous work: matching label distributions of source and target domains. In practice, domain adaptation often encounters label distribution shift, i.e., the frequencies of classes significantly differ between source and target domains. It has been proven in [6, 63] that matching label distributions of source and target domains is a necessary condition for successful domain adaptation [6, 63]. Also, it has been empirically verified in [6] that mismatched label distributions restrict or even deteriorate performance of existing domain adaptation methods [15, 29, 30].

Motivated by this, we present a new method that addresses the label distribution shift for the first time in ADA. At the heart of our method lies LAbel distribution Matching through Density-aware Active sampling, and thus it is dubbed LAMDA. Its key idea is to use sampled data for label distribution matching as well as supervised learning. During training, it estimates the label distribution of the target domain through the annotated labels of sampled target data, and builds each source data mini-batch in a way that the label frequencies of the batch follow the estimated target label distribution. To this end, we design a new sampling strategy useful for label distribution estimation as well as supervised learning. For supervised learning, sampled data are encouraged to be representative, diverse, and uncertain. For label distribution estimation, on the other hand, sampled data should well approximate the entire data distribution of the target domain. As will be demonstrated empirically, existing ADA methods often fail to satisfy the second condition since they blindly select uncertain instances or do not take the overall target distribution into account.

Our sampling method satisfies both of the above conditions. Specifically, it selects a subset of target data whose statistical distance from the entire target data is minimized. Since the distribution of the sampled data well approximates that of the entire target data, their labels are expected to follow the latent target label distribution. They also spontaneously become diverse and representative in order to cover the entire target data distribution. In addition, LAMDA asks the oracle for labeling only uncertain instances in the sampled subset; it in turn utilizes the manually labeled samples for both supervised learning and label distribution estimation, while the rest are assigned pseudo labels by the model's prediction and used only for label distribution estimation. This strategy lets LAMDA annotate and exploit only uncertain data in the subset for supervised learning, and estimate the target label distribution accurately by using the entire subset. The advantage of our sampling method is illustrated in Fig. 1.

In addition, we propose to use the cosine classifier [16, 38], instead of the conventional linear classifier, in order to further alleviate the adverse effect of label distribution shift. The cosine classifier is known to be less biased to domi-
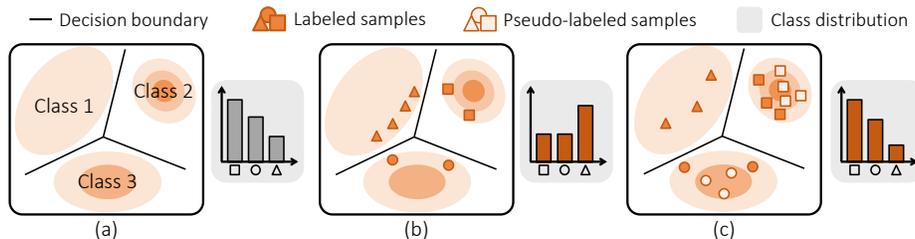
Fig. 1: Comparison between sampling methods. (a) Data distribution and label distribution of target data. (b) Uncertainty preferred sampling of conventional ADA and the label distribution of corresponding sampled data. (c) Density-aware sampling of LAMDA and the label distribution of corresponding sampled data.

nant classes since its classification weights are ℓ_2-normalized, and thus has been used for long-tailed recognition [21] and few-shot learning [5,16,38]. We find that such a property is also useful to combat label distribution shift; it is empirically verified that the cosine classifier significantly improves ADA performance when combined with a domain alignment method.

To evaluate and compare LAMDA with existing ADA methods thoroughly, we present a unified evaluation protocol for ADA. Extensive experiments based on the evaluation protocol demonstrate impressive performance of LAMDA, which largely surpasses records of existing ADA methods [14,36,40], on four public benchmarks for domain adaptation [34,35,53,55]. The main contribution of this paper is four-fold:

- LAMDA is the first attempt to tackle the label distribution shift for ADA. The importance of this research direction is demonstrated by the outstanding performance of LAMDA.
- We propose a new sampling strategy for choosing target data best preserving the entire target data distribution as well as being representative, diverse, and uncertain. Data selected by our strategy are useful for both label distribution matching and supervised learning.
- For the first time, we benchmark existing ADA methods [14,36,40] on four public datasets for domain adaptation [34,35,53,55] through a unified evaluation protocol.
- In our experiment with each of the four domain adaptation datasets, LAMDA substantially outperforms all the existing ADA models.

2   Related work

Unsupervised domain adaptation (UDA). Major approaches in UDA aim at learning domain invariant features so that a classifier trained on the labeled source domain data can be transferred to the unlabeled target domain data [4]. To do so, previous methods align feature distribution between the two domains using various domain discrepancy measures such as MMD [28,30], Wasserstein discrepancy [7,8,11,24], and $\mathcal{H}$-divergence [1,9,15,29,37,54]. On the other hand,

recent studies [6,63] found that such domain alignment is only effective when the label distributions of the two domains are matched. This condition is difficult to be satisfied due to the limited access to the target class distribution. In this work, we propose to utilize the actively sampled data in ADA to estimate the target label distribution and match the label distribution of the two domains for the effective domain alignment.

Active learning (AL). AL is a task of selecting the most performance-profitable samples to be annotated from an oracle [48]. Previous methods design various selection strategies, where they often refer to uncertainty [2, 19, 33], diversity [47, 51], or the both [3, 56, 57] for the selection. Uncertainty-based methods prefer difficult samples for the model, *e.g.*, samples with high entropy. Diversity-based methods prefer samples that are different from the selected ones. Our method shares a similar idea with Wang [56, 57] in that we use MMD [17], but we additionally select the easy-but-representative samples as a pseudo-labeled set to precisely estimate the target label distribution, which can be used to help domain adaptation process.

Active domain adaptation (ADA). ADA is a variant of active learning that selects samples to maximize the domain adaptation performance. ADA is first introduced by Rai *et al.* [39] and first adapted to image classification by AADA [52]. Existing methods mainly refer to the difficulty of samples (*i.e.*, uncertainty) for selection. TQS [14] selects uncertain samples by combining three sampling criteria: disagreement among ensemble models [49], top-2 margin of predictive probabilities [42], and confidence of domain discriminator [52]. CLUE [36] additionally considers the diversity among the selected samples along with the uncertainty by using entropy-weighted k-means clustering [20]. S^{3}VAADA [40] designs a set-based scoring function that favors three properties: vulnerability to adversarial perturbation, diversity within the sampled set, and representativeness to avoid outliers. More recent methods utilize a free energy biases [23] of the two domains [58], K-medoids algorithm [12, 41], and the distance to different class centers [59] for the selection. To newly tackle the critical issue of label distribution shift in ADA, we propose a sampling strategy that considers the data distribution of the target domain. The main technical difference between the sampling of conventional ADA and ours is illustrated in Fig. 1.

3   Problem formulation

Given a labeled source dataset $\mathcal{D}_\mathrm{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_\mathrm{S}}$ of size n_S and an unlabeled target dataset $\mathcal{D}_\mathrm{T} = \{\mathbf{x}_i\}_{i=1}^{n_\mathrm{T}}$ of size n_T, we study a standard ADA scenario of R rounds, in each of which B samples of target data are newly labeled and utilized for model update, *i.e.*, the per-round budget is B and the total budget is $RB \leq n_\mathrm{T}$. Let $\mathcal{D}_\mathrm{L}$ be the labeled target dataset actively collected, which grows up to size RB. We consider image classification such that $\mathbf{x}_i$ is an image and $y_i \in \mathcal{Y} = \{1, 2, \ldots, C\}$ is a categorical variable, where a model, parameterized by $\boldsymbol{\theta}$, predicts $\mathrm{argmax}_{y \in \mathcal{Y}}\, p_{\boldsymbol{\theta}}(y|\mathbf{x})$ for input image $\mathbf{x}$. The goal of ADA is to
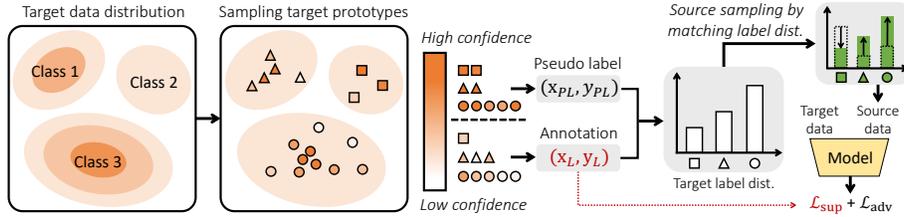
Fig. 2: LAMDA first samples a set of target prototypes that well represent the entire target data distribution. The prototypes are annotated by an oracle if their predictions are uncertain or assigned pseudo labels otherwise. It then estimates the target label distribution using the prototypes, and builds source data mini-batches whose label distributions follow the estimated target label distribution. Finally, our model is trained by the cross-entropy loss $\mathcal{L}_{\text{sup}}$ and the domain adversarial loss $\mathcal{L}_{\text{adv}}$.

maximize the test accuracy of $\boldsymbol{\theta}$ in the target domain, where $\boldsymbol{\theta}$ is trained on $\mathcal{D}_{\text{S}}$, $\mathcal{D}_{\text{T}}$ and $\mathcal{D}_{\text{L}}$ in the iterative manner.

4   Proposed method

4.1   Overview of LAMDA

We present a novel ADA method, named LAMDA, that addresses *label distribution shift* between source and target domains. Our core idea is to select and utilize target samples useful for both label distribution matching and supervised learning. This idea is implemented in LAMDA by three components: prototype sampling, label distribution matching, and model training. First, LAMDA selects a set of prototypes, *i.e.*, target data that best approximate the entire target data distribution. Uncertain prototypes in the set are then identified by the model and annotated by oracle, while the rest are assigned pseudo-labels (Sec. 4.2). Next, LAMDA estimates the target label distribution using the assigned labels of the prototypes and adjusts the label distribution of source data being drawn within each mini-batch according to the estimated target label distribution (Sec. 4.3). Under the matched label distribution, the model is trained by both cross-entropy loss and domain adversarial loss (Sec. 4.4). The overall framework of LAMDA is illustrated in Fig. 2. In what follows, we describe each component at a round.

4.2   Prototype set sampling in target data

We begin with a model $\boldsymbol{\theta}$ which is from the previous round, or pretrained on source dataset $\mathcal{D}_{\text{S}}$ for the first round. Let $X_{(\cdot)}$ denote the set of images in dataset $\mathcal{D}_{(\cdot)}$ for notational simplicity. To select the prototype set that represents the target data distribution, we first seek subset $X \subset X_{\text{T}}$ which minimizes a statistical distance between X and the entire target data X_{T}. Inspired by the sampling technique for example-based model explanation [22], we employ the squared

Maximum Mean Discrepancy (MMD) [17] between X and X_T on the feature space, which is formally given by

$$
\begin{aligned}
\mathrm{MMD}^2(X, X_T) := {} & \frac{1}{|X|^2} \sum_{\mathbf{x}_i, \mathbf{x}_j \in X} k(f(\mathbf{x}_i), f(\mathbf{x}_j)) \\
& - \frac{2}{n_T |X|} \sum_{\mathbf{x}_i \in X, \mathbf{x}_j \in X_T} k(f(\mathbf{x}_i), f(\mathbf{x}_j)) \\
& + \frac{1}{n_T^2} \sum_{\mathbf{x}_i, \mathbf{x}_j \in X_T} k(f(\mathbf{x}_i), f(\mathbf{x}_j)) \,,
\end{aligned}
\tag{1}
$$

where we let $f(\mathbf{x})$ be the feature of input $\mathbf{x}$ extracted by $\boldsymbol{\theta}$, and $k(\mathbf{z}, \mathbf{z}') = \exp(-\gamma \|\mathbf{z} - \mathbf{z}'\|^2)$ be the Radial Basis Function (RBF) kernel. Noting that the last term in Eq. (1) is constant with respect to X, we define $J(X)$ as follows:

$$
\begin{aligned}
J(X) := {} & \mathrm{MMD}^2(\emptyset, X_T) - \mathrm{MMD}^2(X, X_T) \\
= {} & \frac{2}{n_T |X|} \sum_{\mathbf{x}_i \in X, \mathbf{x}_j \in X_T} k(f(\mathbf{x}_i), f(\mathbf{x}_j)) - \frac{1}{|X|^2} \sum_{\mathbf{x}_i, \mathbf{x}_j \in X} k(f(\mathbf{x}_i), f(\mathbf{x}_j)) \,.
\end{aligned}
\tag{2}
$$

where a constant $\mathrm{MMD}^2(\emptyset, X_T)$ is added to make $J(\emptyset) = 0$, and the first and second terms measure representativeness and diversity of X, respectively.

The prototypes can be then identified by a constrained combinatorial optimization to maximize $J(X)$ given a certain size limit n_P, *i.e.*,

$$
\max_{X \in 2^{X_T} : |X| \le n_P} J(X) \,.
\tag{3}
$$

This is generally intractable due to the exponentially many candidates. However, a greedy process selecting samples one after one to locally maximize J can efficiently find a near-optimal solution in polynomial time since $J(X)$ is monotone submodular when k is RBF kernel [22]. To be specific, the greedy process is proven to achieve at least $1 - [(n_P - 1)/n_P]^{n_P}$ of the optimum [32]. We hence adopt the greedy process to select subset X_P from the unlabeled target data.

We note that setting $n_P = B$ and spending all the budget for X_P would be a waste of budget when X_P includes easy prototypes, whose labels are accurately predicted by $\boldsymbol{\theta}$. We hence set n_P in an adaptive way so that we spend budget B only for hard prototypes. To be specific, starting from $X_{PL} = \emptyset$ and X_L from the previous round (or $X_{PL} = X_L = \emptyset$ for the first round), each greedy selection is added to either X_{PL} or X_L. X_{PL} includes only easy prototypes of X_P whose margin between top-1 and top-2 predictions is larger than threshold Δ, and only hard prototypes in $X_L = X_P \setminus X_{PL}$ are labeled by oracle. For X_{PL}, we use top-1 prediction as the pseudo label which is given by

$$
\hat{y}_{i,1} := \operatorname*{argmax}_{y \in \mathcal{Y}} p_{\boldsymbol{\theta}}(y | \mathbf{x}_i) \,.
\tag{4}
$$

In each round, we continue the sampling process until B hard samples are newly annotated by oracle. Thus, $n_P = |X_P| \ge B$ is determined by the adaptation

---

Algorithm 1 Prototype sampling at a round

Require: Model parameter $\boldsymbol{\theta}$ from the previous round, labeled source dataset $\mathcal{D}_\mathrm{S}$, unlabeled target image set X_T, per-round budget B, threshold Δ.
 1: Retrieve X_L from the previous round or set it as empty set for the first round.
 2: Set $X_\mathrm{PL} \leftarrow \emptyset$ and $X_\mathrm{P} \leftarrow X_\mathrm{L} \cup X_\mathrm{PL}$.
 3: **repeat**
 4:      $\mathbf{x}^* \leftarrow \underset{\mathbf{x}_i \in X_\mathrm{T} \setminus X_\mathrm{P}}{\mathrm{argmax}} \; (J(X_\mathrm{P} \cup \{\mathbf{x}_i\}) - J(X_\mathrm{P}))$ ▷ Prototype selection w.r.t. $J(X)$ in Eq. (2)
 5:      $\hat{y}_1 \leftarrow \underset{y \in \mathcal{Y}}{\mathrm{argmax}}\, p_{\boldsymbol{\theta}}(y|\mathbf{x}^*), \; \hat{y}_2 \leftarrow \underset{y \in \mathcal{Y} \setminus \{\hat{y}_1\}}{\mathrm{argmax}}\, p_{\boldsymbol{\theta}}(y|\mathbf{x}^*)$ ▷ Get top-1 and top-2 prediction
 6:      **if** $p_{\boldsymbol{\theta}}(\hat{y}_1|\mathbf{x}^*) - p_{\boldsymbol{\theta}}(\hat{y}_2|\mathbf{x}^*) > \Delta$ **then**   ▷ Identify easy/hard prototype by margin
 7:          $X_\mathrm{PL} \leftarrow X_\mathrm{PL} \cup \{\mathbf{x}^*\}$ ▷ Pseudo-labeling for easy prototype
 8:      **else**
 9:          $X_\mathrm{L} \leftarrow X_\mathrm{L} \cup \{\mathbf{x}^*\}$ ▷ Oracle-labeling for hard prototype
10:      **end if**
11:      $X_\mathrm{P} \leftarrow X_\mathrm{P} \cup \{\mathbf{x}^*\}$ ▷ $X_\mathrm{P} = X_\mathrm{L} \cup X_\mathrm{PL}$
12: **until** B samples are newly added to X_L (and labeled by oracle)
13: **return** $X_\mathrm{P}, X_\mathrm{PL}, X_\mathrm{L}$

---

scenario and the model in hand. This is possible because the greedy selection can return a near-optimal solution at any iteration. The sampling process is illustrated in Fig. 2, and described formally in Algorithm 1.

We denote the set of labeled prototypes by $\mathcal{D}_\mathrm{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^B$ and that of pseudo-labeled prototypes by $\mathcal{D}_\mathrm{PL} = \{(\mathbf{x}_i, \hat{y}_{i,1})\}_{i=1}^{n_\mathrm{PL}}$. $\mathcal{D}_\mathrm{L}$ is used for both supervised learning and label distribution estimation, while $\mathcal{D}_\mathrm{PL}$ is used only for label distribution estimation; details will be described in the following section.

4.3 Label distribution matching

We use the prototype set to estimate the target data distribution $p_\mathrm{T}(y)$, which is in turn used for label distribution matching. To estimate $p_\mathrm{T}(y)$, we investigate the frequency of each class within $\mathcal{D}_\mathrm{L}$ and $\mathcal{D}_\mathrm{PL}$. The frequency of class c in $\mathcal{D}_\mathrm{L}$ is computed by

$$n_{\mathrm{L},c} := \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_\mathrm{L}} \mathbb{1}[y_i = c] \,, \tag{5}$$

where $\mathbb{1}$ is an indicator function. On the other hand, the class frequency in $\mathcal{D}_\mathrm{PL}$ is weighted by the corresponding predictive probability, which is given by

$$\hat{n}_{\mathrm{PL},c} := \sum_{(\mathbf{x}_i, \hat{y}_{i,1}) \in \mathcal{D}_\mathrm{PL}} \mathbb{1}[\hat{y}_{i,1} = c] p_{\boldsymbol{\theta}}(\hat{y}_{i,1}|\mathbf{x}_i) \,, \tag{6}$$

Then, the target label distribution $p_\mathrm{T}(y)$ is estimated by

$$\hat{p}_\mathrm{T}(y) := \frac{n_{\mathrm{L},y} + \hat{n}_{\mathrm{PL},y} + 1}{n_\mathrm{L} + \hat{n}_\mathrm{PL} + C} \,, \tag{7}$$

where $\hat{n}_\mathrm{PL} = \sum_c \hat{n}_{\mathrm{PL},c}$ and C is the number of classes. Note that we add an offset 1 to each category frequency of Eq. (7) to ensure at least a single instance is considered to be present in the target domain. This is consistent with

the assumption of UDA, where both domains have the same label space $\mathcal{Y}$. To make the observed source label distribution follow $\hat{p}_\mathrm{T}(y)$, we apply class-weighted sampling when building source mini-batches. The ratio between the source label distribution $p_\mathrm{S}(y)$ and the estimated target label distribution $\hat{p}_\mathrm{T}(y)$ is denoted by $w(y) := \frac{\hat{p}_\mathrm{T}(y)}{p_\mathrm{S}(y)}$. Then, the probability of sampling $(\mathbf{x}_i, y_i)$ from $\mathcal{D}_\mathrm{S}$ for source mini-batch construction is defined by

$$\rho_i := \frac{w(y_i)}{\sum_{(\mathbf{x}_j,y_j)\in\mathcal{D}_\mathrm{S}} w(y_j)} \ , \tag{8}$$

where i indicates the sample index.

4.4   Model training

Loss functions. As the label frequencies of a source mini-batch match those of the target domain by Eq. (8), we can now apply a domain alignment loss while alleviating the label distribution shift. We choose the domain adversarial loss [15], but any other losses [15,29,30] for domain alignment can be employed. For domain adversarial training, a domain discriminator, parameterized by $\boldsymbol{\phi}$, is trained to classify the domain of input feature by probability $p_{\boldsymbol{\theta}_f,\boldsymbol{\phi}}(d|\mathbf{x})$, where $d \in \{0,1\}$ is domain label. In the meantime, the feature extractor parameterized by $\boldsymbol{\theta}_f$ is adversarially trained to confuse the discriminator. The domain adversarial loss with the matched label distributions is given by

$$\mathcal{L}_\mathrm{adv} := \mathbb{E}_{\mathbf{x}_i \overset{\rho_i}{\sim} X_\mathrm{S}}[-\log p_{\boldsymbol{\theta}_f,\boldsymbol{\phi}}(d|\mathbf{x}_i)] + \mathbb{E}_{\mathbf{x}_j \overset{\mathrm{iid}}{\sim} X_\mathrm{T}}[-\log(1 - p_{\boldsymbol{\theta}_f,\boldsymbol{\phi}}(d|\mathbf{x}_j))] \ , \tag{9}$$

where the first expectation is taken over ρ_i of X_S and the second one is taken over uniform distribution of X_T. The $\boldsymbol{\theta}_f$ is updated to maximize $\mathcal{L}_\mathrm{adv}$, while $\boldsymbol{\phi}$ is updated to minimize $\mathcal{L}_\mathrm{adv}$. Meanwhile, the cross-entropy loss for labeled data $\mathcal{D}_\mathrm{S}$ and $\mathcal{D}_\mathrm{L}$ is given by

$$\mathcal{L}_\mathrm{sup} := \mathbb{E}_{(\mathbf{x}_i,y_i) \overset{\rho_i}{\sim} \mathcal{D}_\mathrm{S}}[-\log p_\theta(y_i|\mathbf{x}_i)] + \mathbb{E}_{(\mathbf{x}_j,y_j) \overset{\mathrm{iid}}{\sim} \mathcal{D}_\mathrm{L}}[-\log p_\theta(y_j|\mathbf{x}_j)] \ . \tag{10}$$

In summary, the total training loss for the proposed framework is given by

$$\mathcal{L} := \mathcal{L}_\mathrm{sup} + \mathcal{L}_\mathrm{adv} \ . \tag{11}$$

Cosine classifier. To further alleviate the negative effect of label distribution shift, LAMDA employs a cosine classifier [16,38], which measures cosine similarities between the classifier weights and an embedding vector as classification scores. The norm of classifier weight is known to be greatly affected by the label distribution [16,21,60]. Since the norm does not interfere with the classification score in the cosine classifier, it can alleviate the label distribution shift. Specifically, let $\mathbf{W} := \{\mathbf{w}_c\} \in \mathbb{R}^{d \times C}$, where $\mathbf{w}_c \in \mathbb{R}^d$ indicates a weight of classifier for class c with embedding dimension d. Then, the class probability predicted by the cosine classifier is given by

$$p_{\boldsymbol{\theta}}(y = c|\mathbf{x}_i) := \mathrm{softmax}\left(\frac{h \circ f(\mathbf{x}_i)^\top \mathbf{w}_c}{\tau \, ||h \circ f(\mathbf{x}_i)|| \, ||\mathbf{w}_c||}\right) \ , \tag{12}$$

where h is a single hidden layer that projects feature vector $f(\mathbf{x})$ into d-dimensional embedding space, and τ is a temperature term that adjusts sharpness of the predicted probability.

5 Experiments

We first describe datasets, experiment setup, and implementation details in Sec. 5.1. Then LAMDA is evaluated and compared with previous work in Sec. 5.2, and contributions of its components are scrupulously analyzed in Sec. 5.3.

5.1 Setup

Datasets. We use four domain adaptation datasets with different characteristics: OfficeHome [55], OfficeHome-RSUT [53], VisDA-2017 [35], and Domain-Net [34]. OfficeHome contains 16k images from four domains {Art, Clipart, Product, Real}, where we conduct a diverse set of domain adaptation for each of 12 source-target permutations. OfficeHome-RSUT is a dataset sub-sampled from three domains {Clipart, Product, Real} of OfficeHome, where the subsampling protocol, called reversely-unbalanced source and unbalanced target (RSUT), is employed to make a large label shift between source and target domains. VisDA-2017 is a large-scale dataset consisting of 207k images from two domains {Synthetic, Real} in a realistic scenario of synthetic-to-real domain adaptation. DomainNet is also a large-scale dataset but has a prevalent labeling noise. In DomainNet, we use five domains {Real, Clipart, Painting, Sketch, Quickdraw}[1] consisting of 362k images. We use 10% of the datasets for validation and the rest are kept for training. While DomainNet includes an independent test set, the other datasets do not provide an explicit test set. Hence, for OfficeHome, OfficeHome-RSUT, and VisDA-2017, we use the whole dataset (*i.e.*, trainval set) as the test set following the conventional protocol of UDA and previous work on ADA [14, 36].

Experimental setup. We compare LAMDA to the state-of-the-art ADA methods: TQS [14], CLUE [36], and S^{3}VAADA [40]. We note that the existing ADA works have evaluated their methods with different evaluation protocols (e.g., budget size, sampling interval, and dataset). For fair comparison, we first benchmark them on four public datasets for domain adaptation through a unified evaluation protocol. We conduct 5 rounds of data sampling, each of which updates the model from the previous round after newly acquiring labels of 2%-budget, *i.e.*, 10%-budget in total, where we let *n%-budget* denote n% of the target train set size. For both of our method and the previous methods, the model is selected based on the validation accuracy. For each of the methods, we use the original authors' official implementation. The detailed descriptions are provided in the supplementary material (Sec. C).

---

[1] The domains are chosen considering their consistency with existing benchmarks [36].
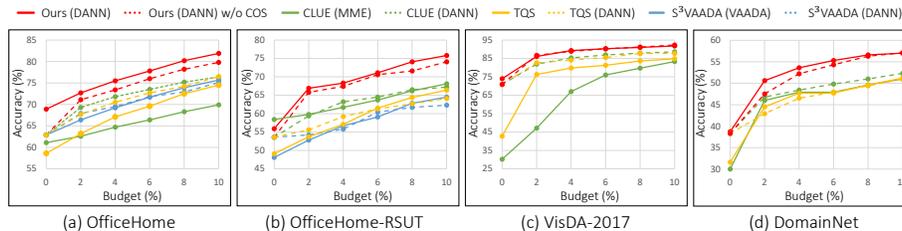
Fig. 3: Accuracy versus the percent of labeled target instances as budget. The accuracies are averaged on *all* scenarios of the OfficeHome, OfficeHome-RSUT, VisDA-2017, and DomainNet. The solid lines represent the results of using the specialized adaptation technique of each method, and the dotted lines represent the results of using the same adaptation technique (*i.e.*, DANN [15]). w/o COS: Ours without cosine classifier

Implementation details. We use ResNet-50 [18] backbone initialized with pretrained weights from ImageNet [13] classification for both our and the previous methods. Our classifier consists of 2 fully connected layers where the embedding dimension d is 512. For all experiments, we use an identical set of hyperparameters. Our model is trained using SGD optimizer with a learning rate of 0.1, and a weight decay of 5^{-4} for 100 epochs. We set the margin threshold Δ to 0.8, the temperature τ in Eq. (12) to 0.1 and the γ of RBF kernel in Eq. (1) to an inverse of the feature dimension, which in our case is $\frac{1}{2048}$.

5.2   Results

Overall superiority of LAMDA with varying budget. In Fig. 3, we compare the performance of LAMDA and the existing approaches[2] varying budget for each of OfficeHome, OfficeHome-RSUT, VisDa-2017, and DomainNet datasets. Note that each method is equipped with its own domain adaptation technique (*e.g.*, VAADA [50] for S^{3}VAADA, and MME [43] for CLUE) and classifier (*i.e.*, cosine classifier for LAMDA). We evaluate these methods while varying their adaptation techniques or classifier to examine the contribution of their components thoroughly. The results show that LAMDA clearly outperforms the previous arts in every setting on all the datasets. In particular, LAMDA with only 2%-budget is often as competitive as or even outperforms the methods with 10%-budget. The performance gap between LAMDA and other methods increases as the budget increases. This suggests that LAMDA utilizes the budget effectively by both ways: label distribution matching and supervised learning.

Advantages of LAMDA across diverse source-target domain pairs. In Table 1-2, we compare LAMDA and the existing ADA methods in every domain

---

[2] Unfortunately, S^{3}VAADA [40] for DomainNet and VisDA-2017 requires infeasible memory consumption, in the supplementary material, we report its performance on a part of scenarios of DomainNet which our resource allows.

Table 1: Accuracy (%) on OfficeHome using 10%-budget for each source-target pair of four domains: **A**rt, **C**lipart, **P**roduct, and **R**eal. w/o COS: Ours without cosine classifier

DA method	AL method	OfficeHome												
		A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
-	TQS [14]	64.3	84.8	83.5	66.1	81.0	76.7	66.5	61.4	82.0	73.7	65.9	88.5	74.5
MME	CLUE [36]	62.1	80.6	73.9	55.2	76.4	75.4	53.9	62.1	80.7	67.5	63.0	88.1	69.9
VAADA	S³VAADA [40]	67.8	83.9	82.9	67.0	81.5	79.5	65.8	65.9	82.4	74.8	68.6	87.8	75.7
	TQS [14]	68.7	80.1	83.1	64.0	83.1	76.9	67.7	71.0	84.4	76.4	72.7	90.0	76.5
	CLUE [36]	70.3	81.9	80.4	65.6	83.8	75.8	64.7	73.9	82.7	76.1	74.3	87.0	76.4
DANN	S³VAADA [40]	65.5	79.6	80.0	65.4	82.2	75.5	68.4	68.1	84.0	73.5	70.7	88.6	75.1
	Ours w/o COS	73.0	87.6	84.2	69.5	85.9	81.0	71.9	74.6	85.3	77.3	75.9	91.6	79.8
	Ours	**74.8**	**88.5**	**86.9**	**73.8**	**88.2**	**83.3**	**74.6**	**75.5**	**86.9**	**80.8**	**77.8**	**91.7**	**81.9**

Table 2: (a) Accuracy (%) on OfficeHome-RSUT using 10%-budget for each source-target pair of three domains: **C**lipart, **P**roduct, and **R**eal. (b) Accuracy (%) on VisDA-2017 and DomainNet using 10%-budget where VisDA-2017 consists of two domains: **R**eal and **S**ynthetic, and DomainNet consists of five domains: **R**eal, **C**lipart, **S**ketch, **P**ainting, and **Q**uickdraw. w/o COS: Ours without cosine classifier

DA method	AL method	OfficeHome-RSUT						
		C→P	C→R	P→C	P→R	R→C	R→P	Avg
-	TQS [14]	69.4	65.7	53.0	76.3	53.1	81.1	66.4
MME	CLUE [36]	69.7	65.9	57.1	73.4	59.5	82.7	68.1
VAADA	S³VAADA [40]	73.0	63.0	50.7	69.6	52.6	78.3	64.5
	TQS [14]	67.6	61.4	54.8	74.7	53.6	77.6	64.9
	CLUE [36]	71.5	64.3	56.3	76.5	54.6	79.9	67.2
DANN	S³VAADA [40]	66.9	61.4	53.0	75.4	52.4	76.4	64.2
	Ours w/o COS	78.1	72.1	61.5	82.3	64.2	86.5	74.1
	Ours	**81.2**	**75.7**	**64.1**	81.6	**65.1**	**87.2**	**75.8**

(a)

DA method	AL method	VisDa-2017	DomainNet				
		S→R	R→C	C→S	S→P	C→Q	Avg
-	TQS [14]	84.8	54.2	51.7	51.4	47.4	51.2
MME	CLUE [36]	83.3	60.7	50.4	53.5	39.4	51.0
	TQS [14]	87.7	59.3	50.9	52.4	41.5	51.0
DANN	CLUE [36]	88.6	60.9	52.2	52.4	43.7	52.3
	Ours w/o COS	**92.3**	64.6	**56.4**	**58.7**	**48.5**	**57.1**
	Ours	91.8	**65.3**	56.1	58.1	48.3	57.0

(b)

adaptation scenario of the four datasets given 10%-budget, where LAMDA always outperforms the others. Regarding that OfficeHome-RSUT has a significant class distribution shift compared to OfficeHome, the advantage of LAMDA equipped with the label distribution matching becomes clearer in OfficeHome-RSUT (Table 2a) than OfficeHome (Table 1). Table 2b demonstrates the scalability of LAMDA, where it clearly outperforms the previous work by about 4% or more in all scenarios of the large-scale datasets, VisDA 2017 and DomainNet. In the supplementary material (Sec. B.1), we also show that LAMDA surpasses state-of-the-art SSDA methods [26, 27].

5.3 Analysis

Contribution of each component of LAMDA. Table 3 quantifies the contribution of each components of LAMDA: (i) prototype set sampling in Sec. 4.2; (ii) label distribution matching in Sec. 4.3; and (iii) cosine classifier in Sec. 4.4. Every component in LAMDA improves the performance in both OfficeHome and OfficeHome-RSUT. The performance gap between the last (random sampling with DANN [15]) and the second last rows verifies that our prototype

Table 3: Accuracy (%) averaged over *all* scenarios when using 10%-budget, where we conduct an ablation study from ablation baseline at the last row to LAMDA at the first row by sequentially adding three components: (i) Prototype: sampling described in Sec. 4.2 (o/w, sampling uniformly at random); (ii) Matching: label distribution matching in Sec. 4.3 (o/w, replacing p_i in Eq. (8) with uniform distribution); and (iii) Cosine: cosine classifier described in Sec. 4.4 (o/w, linear classifier). (·): accuracy gain by adding each component.

Prototype	Matching	Cosine	OfficeHome	OfficeHome-RSUT
✓	✓	✓	81.9 (+2.1)	75.8 (+1.7)
✓	✓	✗	79.8 (+2.7)	74.1 (+6.8)
✓	✗	✗	77.1 (+3.8)	67.3 (+3.7)
✗	✗	✗	73.3	63.6

(a) Estimated label distribution   (b) Domain alignment training
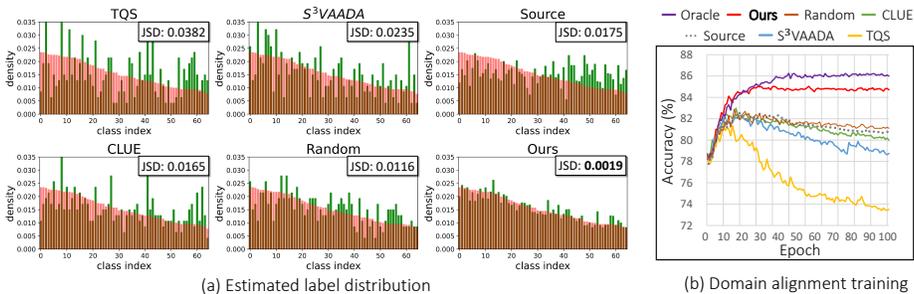
Fig. 4: Effect of active sampling strategy on label distribution matching (Sec. 4.3) in OfficeHome Real to Product scenario. (a) The true (red) and the estimated (green) label distribution of target domain, where each sampling methods estimates the distribution using 10%-budget. The methods are sorted by the estimation quality. JSD: Jensen-Shannon Divergence between the estimated and the true label distribution (lower is better). Source: Label distribution of source data. (b) Training curve of domain alignment learning (Eq. (9)) combined with label distribution matching using the estimations in (a). Source: naive domain alignment. Oracle: using true target label distribution.

sampling method boost the effect of supervised learning. Comparing the second and the third rows, one can see the remarkable performance gain by our label distribution matching strategy, in particular on OfficeHome-RSUT with significant label distribution shift. Finally, the use of cosine classifier further improves performance by 1.9% in average. The results for every individual adaptation scenarios are reported in the supplementary material (Sec. A.1).

Quality of estimated label distribution. As described in Sec. 4.3, estimating target label distribution plays a prominent role in LAMDA. In Fig. 4a, we visualize label distributions of sampled data of LAMDA and those of the previous work, and compute Jensen-Shannon divergence (JSD) between the estimated distributions and the true one. The results demonstrate that LAMDA enables to estimate target label distribution most accurately compared to the previous

t-SNE colored with our sampling   *Unclustered region*   *t*-SNE colored with TQS sampling
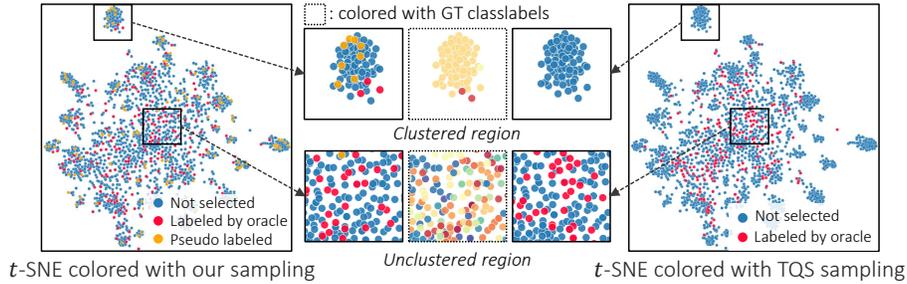
Fig. 5: *t*-SNE [31] visualization of target feature vectors from source pre-trained model on OfficeHome Real to Art scenario.

work and the random sampling, which is a naive but intuitive sampling strategy for the estimation. Note that the previous work is even worse than the random sampling in terms of the estimation accuracy, which empirically reconfirm that the sampling strategies of the previous work are not aware of the target data distribution. When solely utilizes all of the pseudo-labels from source pretrained model for the estimation, it gives JSD of 0.025 which is worse than 'Source' baseline. This is mainly due to the bias of the pseudo-labeled data; they are highly confident samples. Our sampling method avoids this bias by combining labeled and pseudo-labeled data.

Benefit of label distribution matching. In Fig. 4b, we plot training curves of domain alignment combined with label distribution matching, where each methods utilizes identical source classification loss and domain alignment loss as in Eq. (9), but with different label distribution estimated from each sampling methods in Fig. 4a. Training without label distribution matching (*e.g.*, *Source*) or matching with inaccurate target label distribution degrade accuracy, while ours does not, thanks to the accurate estimation of target label distribution. It is worth noting that our model using 10%-budget shows comparable accuracy with *Oracle*, which has access to the true target label distribution.

Visualization of sampled data by *t*-SNE. Fig. 5 visualizes distributions of target features and those selected by LAMDA and TQS, to show the difference of their sampling strategies. Since TQS prefers to select uncertain data, mostly located in unclustered regions, its samples do not reflect the target data distribution, *e.g.*, the certain instances in the clustered region are undersampled. In contrast, LAMDA considers certain samples ignored in TQS and assigns them pseudo-labels for label distribution prediction, while it requests an oracle to annotate uncertain data within the budget. Such a sampling strategy allows us to mainly invest a budget on uncertain data while utilizing density-aware samples to estimate the target label distribution. These observations align with our design rationale, depicted in Fig. 1. We also visualize the selected target feature vectors of CLUE and S³VAADA in the supplementary material (Sec. A.4).

Hyper-parameter analysis. In Fig. 6, we evaluate the sensitivity of LAMDA to the choice of the threshold Δ in Algorithm 1. LAMDA is surprisingly robust to the change of Δ, where the change of accuracy is less than 1% for both OfficeHome and OfficeHome-RSUT when the Δ is between 0.7 and 0.9. We note that while the optimal value of Δ varies among the datasets, we use the same

Fig. 6: Accuracy of LAMDA versus hyper-parameter Δ. The blue dot indicates the value used in the main paper.

value for all of our experiments. When we do not utilize the pseudo label in LAMDA (*i.e.*, $\Delta = 1$), the accuracy drops 4% and 1.1% in OfficeHome and OfficeHome-RSUT, respectively. This shows the effectiveness of our prototype sampling strategy.

Analysis of cosine classifier. To inspect the cosine classifier, we compare in Fig. 7 the frequencies and the weight norm of the linear classifier for each class. The norm of the linear classifier is positively correlated to the frequencies of each class within the source domain (blue and green lines). Since a large norm of classifier weights has been known to result in predictions biased to major classes [21], the mismatch between the classifier norm and the target domain class frequencies (red and blue lines) is undesirable. The cosine classifier alleviates

Fig. 7: The frequencies (# ins) of each class in source and target domain (OfficeHome-RSUT Clipart to Product scenario) and the l2-norm of the corresponding classifier weight trained with the source data.

this issue by normalizing its weight scale. Existing ADA methods combined with cosine classifier are evaluated in the supplementary material (Sec. B.2).

6   Conclusion

We proposed LAMDA, a new method to address the issue of label distribution shift in ADA. It selects target data best preserving the target data distribution as well as being representative, diverse, and uncertain. During training, LAMDA estimates the label distribution of the target domain, and builds each source data mini-batch in a way that the label frequencies of the batch follow the estimated target label distribution. On the four different domain adaptation datasets, the proposed method substantially outperforms all the existing ADA models.

References

1. Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M.: Domain-adversarial neural networks. arXiv preprint arXiv:1412.4446 (2014)
2. Asghar, N., Poupart, P., Jiang, X., Li, H.: Deep active learning for dialogue generation. In: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017) (2017)
3. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: Proc. International Conference on Learning Representations (ICLR) (2020)
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1), 151–175 (2010)
5. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.: A closer look at few-shot classification. In: Proc. International Conference on Learning Representations (ICLR) (2019)
6. Tachet des Combes, R., Zhao, H., Wang, Y.X., Gordon, G.J.: Domain adaptation with conditional distribution matching and generalized label shift. In: Proc. Neural Information Processing Systems (NeurIPS) (2020)
7. Courty, N., Flamary, R., Habrard, A., Rakotomamonjy, A.: Joint distribution optimal transportation for domain adaptation. In: Proc. Neural Information Processing Systems (NeurIPS) (2017)
8. Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **39**(9), 1853–1865 (2016)
9. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: Proc. Neural Information Processing Systems (NeurIPS) (2020)
10. Dai, D., Sakaridis, C., Hecker, S., Van Gool, L.: Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. International Journal of Computer Vision (IJCV) (2020)
11. Damodaran, B.B., Kellenberger, B., Flamary, R., Tuia, D., Courty, N.: Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
12. Deheeger, F., MOUGEOT, M., Vayatis, N., et al.: Discrepancy-based active learning for domain adaptation. In: Proc. International Conference on Learning Representations (ICLR) (2021)
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
14. Fu, B., Cao, Z., Wang, J., Long, M.: Transferable query selection for active domain adaptation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
15. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research (JMLR) **17**(1), 2096–2030 (2016)
16. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
17. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. Journal of Machine Learning Research (JMLR) **13**(1), 723–773 (2012)

18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
19. He, T., Jin, X., Ding, G., Yi, L., Yan, C.: Towards better uncertainty sampling: Active learning with multiple views for deep convolutional neural network. In: 2019 IEEE International Conference on Multimedia and Expo (ICME) (2019)
20. Huang, J.Z., Ng, M.K., Rong, H., Li, Z.: Automated variable weighting in k-means type clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **27**(5), 657–668 (2005)
21. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: Proc. International Conference on Learning Representations (ICLR) (2020)
22. Kim, B., Khanna, R., Koyejo, O.O.: Examples are not enough, learn to criticize! criticism for interpretability. In: Proc. Neural Information Processing Systems (NeurIPS) (2016)
23. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predicting structured data **1**(0) (2006)
24. Lee, C.Y., Batra, T., Baig, M.H., Ulbricht, D.: Sliced wasserstein discrepancy for unsupervised domain adaptation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
25. Lee, S., Son, T., Kwak, S.: Fifo: Learning fog-invariant features for foggy scene segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
26. Li, J., Li, G., Shi, Y., Yu, Y.: Cross-domain adaptive clustering for semi-supervised domain adaptation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
27. Li, K., Liu, C., Zhao, H., Zhang, Y., Fu, Y.: Ecacl: A holistic framework for semi-supervised domain adaptation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
28. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: Proc. International Conference on Machine Learning (ICML). PMLR (2015)
29. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. Proc. Neural Information Processing Systems (NeurIPS) (2018)
30. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: Proc. International Conference on Machine Learning (ICML). PMLR (2017)
31. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research (JMLR) **9**(11) (2008)
32. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. Mathematical programming **14**(1), 265–294 (1978)
33. Ostapuk, N., Yang, J., Cudré-Mauroux, P.: Activelink: deep active learning for link prediction in knowledge graphs. In: The World Wide Web Conference (WWW) (2019)
34. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
35. Peng, X., Usman, B., Kaushik, N., Wang, D., Hoffman, J., Saenko, K.: Visda: A synthetic-to-real benchmark for visual domain adaptation. In: Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 2021–2026 (2018)

36. Prabhu, V., Chandrasekaran, A., Saenko, K., Hoffman, J.: Active domain adaptation via clustering uncertainty-weighted embeddings. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2021)

37. Purushotham, S., Carvalho, W., Nilanon, T., Liu, Y.: Variational recurrent adversarial deep domain adaptation. In: Proc. International Conference on Learning Representations (ICLR) (2016)

38. Qi, H., Brown, M., Lowe, D.G.: Low-shot learning with imprinted weights. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

39. Rai, P., Saha, A., Daumé III, H., Venkatasubramanian, S.: Domain adaptation meets active learning. In: Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing (2010)

40. Rangwani, H., Jain, A., Aithal, S.K., Babu, R.V.: S3vaada: Submodular subset selection for virtual adversarial active domain adaptation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2021)

41. Rdusseeun, L., Kaufman, P.: Clustering by means of medoids. In: Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland. vol. 31 (1987)

42. Roth, D., Small, K.: Margin-based active learning for structured output spaces. In: European Conference on Machine Learning. pp. 413–424. Springer (2006)

43. Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

44. Sakaridis, C., Dai, D., Gool, L.V.: Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

45. Sakaridis, C., Dai, D., Hecker, S., Van Gool, L.: Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In: Proc. European Conference on Computer Vision (ECCV) (2018)

46. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. International Journal of Computer Vision (IJCV) (2018)

47. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A coreset approach. In: Proc. International Conference on Learning Representations (ICLR) (2018)

48. Settles, B.: Active learning literature survey (2009)

49. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: Proceedings of the fifth annual workshop on Computational learning theory. pp. 287–294 (1992)

50. Shu, R., Bui, H.H., Narui, H., Ermon, S.: A dirt-t approach to unsupervised domain adaptation. In: Proc. International Conference on Learning Representations (ICLR) (2018)

51. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

52. Su, J.C., Tsai, Y.H., Sohn, K., Liu, B., Maji, S., Chandraker, M.: Active adversarial domain adaptation. In: Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2020)

53. Tan, S., Peng, X., Saenko, K.: Class-imbalanced domain adaptation: an empirical odyssey. In: Proc. European Conference on Computer Vision (ECCV) (2020)

54. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2015)
55. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
56. Wang, Z., Du, B., Tu, W., Zhang, L., Tao, D.: Incorporating distribution matching into uncertainty for multiple kernel active learning. IEEE Transactions on Knowledge and Data Engineering **33**(1), 128–142 (2019)
57. Wang, Z., Ye, J.: Querying discriminative and representative samples for batch mode active learning. ACM Transactions on Knowledge Discovery from Data (TKDD) **9**(3), 1–23 (2015)
58. Xie, B., Yuan, L., Li, S., Liu, C.H., Cheng, X., Wang, G.: Active learning for domain adaptation: An energy-based approach. In: Proc. AAAI Conference on Artificial Intelligence (AAAI) (2022)
59. Xie, M., Li, Y., Wang, Y., Luo, Z., Gan, Z., Sun, Z., Chi, M., Wang, C., Wang, P.: Learning distinctive margin toward active domain adaptation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
60. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
61. Yang, L., Wang, Y., Gao, M., Shrivastava, A., Weinberger, K.Q., Chao, W.L., Lim, S.N.: Deep co-training with task decomposition for semi-supervised domain adaptation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
62. Yoon, J., Kang, D., Cho, M.: Semi-supervised domain adaptation via sample-to-sample self-distillation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1978–1987 (2022)
63. Zhao, H., Des Combes, R.T., Zhang, K., Gordon, G.: On learning invariant representations for domain adaptation. In: Proc. International Conference on Machine Learning (ICML). PMLR (2019)