Active Universal Domain Adaptation

Xinhong Ma[1,2], Junyu Gao[1,2] and Changsheng Xu[1,2,3]
[1] National Lab of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences (CASIA)
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)
[3] Peng Cheng Laboratory, Shenzhen, China
{xinhong.ma, junyu.gao, csxu}@nlpr.ia.ac.cn

Abstract

Most unsupervised domain adaptation methods rely on rich prior knowledge about the source-target label set relationship, and they cannot recognize categories beyond the source classes, which limits their applicability in practical scenarios. This paper proposes a new paradigm for unsupervised domain adaptation, termed as Active Universal Domain Adaptation (AUDA), which removes all label set assumptions and aims for not only recognizing target samples from source classes but also inferring those from target-private classes by using active learning to annotate a small budget of target data. For AUDA, it is challenging to jointly adapt the model to the target domain and select informative target samples for annotations under a large domain gap and significant semantic shift. To address the problems, we propose an Active Universal Adaptation Network (AUAN). Specifically, we first introduce Adversarial and Diverse Curriculum Learning (ADCL), which progressively aligns source and target domains to classify whether target samples are from source classes. Then, we propose a Clustering Non-transferable Gradient Embedding (CNTGE) strategy, which utilizes the clues of transferability, diversity, and uncertainty to annotate target informative sample, making it possible to infer labels for target samples of target-private classes. Finally, we propose to jointly train ADCL and CNTGE with target supervision to promote domain adaptation and target-private class recognition. Extensive experiments demonstrate that the proposed AUDA model equipped with ADCL and CNTGE achieves significant results on four popular benchmarks.

1. Introduction

Recent advances in deep neural networks have convincingly demonstrated the high capability of learning effective models on large datasets. The impressive achievements heavily rely on quantities of labeled training instances,

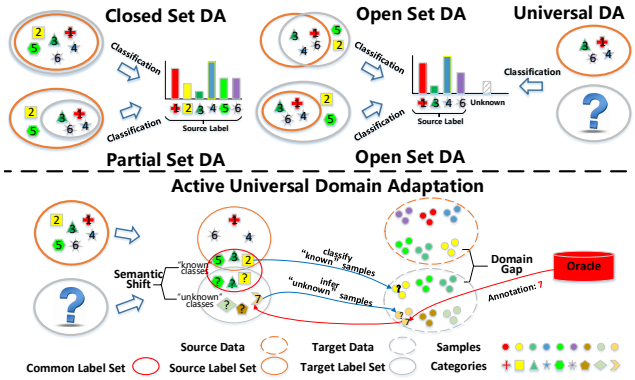


Figure 1. Comparison between Active Universal Domain Adaptation and representative domain adaptation settings with respect to classification tasks and assumptions on target label set. AUDA removes all label set assumptions and aims for not only recognizing target samples belonging to the shared common label set but also inferring labels for those belong to target-private label set by using active learning to annotate a small budget of target data.

which requires expensive and time-consuming labor work of collection and annotation. A reasonable question is why not directly recycling the off-the-shelf knowledge or models from a source domain to new domains. As data from different domains are sampled from different data distributions, there is probably a large domain gap [60] which may degrade the model performance in the target domain [35, 40]. An appealing way to address this issue is *Unsupervised Domain Adaptation (UDA)* [44], which aims to learn a classification model with source labeled data and target unlabeled data to ensure that the learned model could perform well in the target domain.

Most unsupervised domain adaptation methods can be divided into four categories, namely, closed set domain adaptation [25, 42, 47, 52, 57, 23], partial domain adaptation [4, 5, 6], open set domain adaptation [36, 46, 65, 28], and universal domain adaptation [61, 11, 45], as shown in the top of Figure 1. Specifically, closed set domain adaptation [16, 30, 33] supposes that the source and target do-

mains share the same label set. The partial domain adaptation [4, 64, 6] assumes that the source label set contains the target label set. The open set domain adaptation assumes that common classes between two domains are known [36] or the source label set is a subset of the target label set [46]. Recently, Universal Domain Adaptation [61, 11, 45] removes all assumptions about source-target label set relationship, and classifies target samples as labels contained in the source label set or marks them as "unknown" similar to the open set domain adaptation. However, the "unknown" category is still unknown, which is inapplicable for practical applications, *e.g.*, new products recommendation or rare animal/plant recognition. Therefore, it is necessary for practical domain adaptation algorithms to infer actual labels for samples belonging to the "unknown" category.

To achieve this goal, we propose to define a new paradigm for unsupervised domain adaptation, referred as *Active Universal Domain Adaptation (AUDA)*. As shown in the bottom of Figure 1, a labeled source domain and a target domain without any explicit restrictions on the classes are provided for model training. Classes are defined as "known" if they belong to the source label set. Otherwise, they are defined as "unknown". Since target samples of "unknown" classes are much more difficult to recognize than the ones of "known" classes, AUDA algorithms need to draw knowledge from the source domain to firstly recognize the "known"/"unknown" label for the test samples from the target domain. Then, actual class labels should be inferred for both "known" and "unknown" samples. However, it is nearly impossible to infer labels for the "unknown" samples without any labeled training data. Since practical applications offer the possibility of annotating a small budget of target instances, termed as *Active Learning (AL)*, we are motivated to acquire labels for a subset of target data from an oracle, especially, labels of target "unknown" samples, to assist the unknown category inference.

To design algorithms for active universal domain adaptation, we are exposed to two aspects of technical challenges: **(1)** Without any prior knowledge of the source-target label set relationship, there exist a large domain gap and significant semantic shift problems in AUDA. Specifically, source and target data are sampled from different distributions and the domain gap makes it hard to recognize "known"/"unknown" instances in the target domain. Moreover, the unexpectable semantic shift means that many unknown classes are contained in the target domain, making it extremely difficult to reduce the domain gap between the shared classes. If the domain gap and semantic shift cannot be well reduced, it is challenging for active learning to annotate informative instances to infer target "unknown" instances. **(2)** During active learning, the most informative target instances should be annotated and used for learning to infer target "unknown" instances. Most existing AL approaches prefer to annotate instances that are highly uncertain [10, 12, 24, 54] or diverse [49, 15]. As these approaches perform active learning without considering domain gap and semantic shift, uncertainty and diversity may be wrongly estimated [34]. Therefore, directly applying the traditional AL approaches easily lead to select outliers, redundant instances, or uninformative instances for annotation, which is detrimental for further reducing the domain gap and semantic shift, and damages the performance of inferring target "unknown" samples. Although the prior work in active domain adaptation [53] tries to deal with the problem of domain gap, it dose not consider the semantic shift problem, making it inapplicable for AUDA. As a result, it is advisable to design active learning strategies that can annotate the most informative target instances with the joint consideration of domain gap and semantic shift.

Motivated by the above observations, we propose an Active Universal Adaptation Network, which simultaneously adapts the model from the source domain to the target domain, and performs active learning towards target informative instances for unknown category inference. Specifically, we first propose Adversarial and Diverse Curriculum Learning (ADCL), which designs an adversarial curriculum loss and a diverse curriculum loss to align source and target domains, and learn the ability of target "known"/"unknown" instances recognition[1]. Thus, the negative effects of domain gap and semantic shift in active learning can be alleviated, which helps to select more informative instances for annotation. Then, we propose an active learning strategy named Clustering Non-transferable Gradient Embedding (CNTGE), which utilizes the clues of transferability, diversity, and uncertainty to annotate target samples of target-private classes and assign pseudo labels to highly confident target "known" instances. The labeled and pseudo labeled target instances could provide better supervision for ADCL, which helps to learn better curriculums. Finally, jointly training with ADCL and CNTGE could further reinforce the adaptation process, and learn to infer actual labels for target "unknown" instances.

The main contributions of this paper are: (1) We introduce a more practical unsupervised domain adaptation paradigm, Active Universal Domain Adaptation, which requires no assumptions about the target label set and aims for not only recognizing target samples belonging to the shared label set but also inferring those of target-private classes via active learning. (2) To address the AUDA task, we propose Active Universal Adaptation Network, an end-to-end model, which performs adversarial and diverse curriculum learning and clustering non-transferable gradient embedding to cooperatively promote domain adaptation and active

[1]Here, we define target samples from the common label set as the target "known" instances while others from target private label set are target "unknown" instances.

learning. (3) Extensive experiments demonstrate that the proposed AUDA model equipped with ADCL and CNTGE achieves significant classification results.

2. Related Work

Domain Adaptation. According to the assumptions of the source-target label set, most domain adaptation approaches can be categorized into closed set adaptation, partial domain adaptation, openset domain adaptation, and universal domain adaptation. *Closed Set Domain Adaptation* assumes that the source and target domains share the same label set, which focuses on mitigating the impact of the domain gap between source and target domains. Solutions to closed set domain adaptation mainly fall into feature adaptation [17, 31, 63, 26, 9] and generative model [13, 21, 22, 29, 55, 59, 32]. *Partial Domain Adaptation* assumes that the label set of the source domain is supposed to be large enough to contain the target label set [4, 5, 6, 8]. *Open Set Domain Adaptation* assumes that classes shared by two domains are known [36] or the source label set is a subset of the target label set [46], which could classify target samples as source classes or a "unknown" class. Although knowledge graph is leveraged to further infer actual labels for "unknown" samples [65], it still follows the open set domain adaptation assumptions. *Universal Domain Adaptation* [61, 11, 45] adopts a more generated setting, which can classify target samples as any class in the source labels set or mark them as "unknown" without any prior knowledge on the target label set. Unfortunately, the existing paradigms of unsupervised domain adaptation can only classify samples as source classes. As for others, they can only be marked as an "unknown" class. Different from the existing DA settings, we are motivated to infer actual classes for all target instances without any assumptions about the source-target label set relationship via the cooperation between domain adaptation and active learning.

Active Learning. Active Learning aims to develop label-efficient algorithms by sampling the most representative queries to be labeled by an oracle [50]. Current approaches can be mainly divided into two categories: *uncertainty* and *diversity*. The first one aims to annotate samples for which the model has uncertain prediction [12, 10, 54, 48, 58, 43, 18, 3]. The second focuses on picking a set of instances that are representative and diverse for the entire dataset rata [49, 15, 51, 14]. Several approaches also propose a trade-off between uncertainty and diversity [20, 2]. Recently, active learning with domain adaptation, termed as *Active Domain Adaptation*, is of great practical interest. However, only a little previous work addresses the problem. The pioneering work [41] studies the task of active adaptation applied to sentiment classification for text data. Rita *et al*. [7] select target samples to learn importance weights for source instances by solving a convex optimization problem of minimizing maximum mean discrepancy (MMD). However, those strategies do not fit model adaptation with deep nets. More Recently, Su *et al*. [53] study this task in the context of deep convolutional nets and instances are selected based on their uncertainty and "targetness". However, these label acquisition strategies are designed based on the assumption that source and target domains share the same label set. In our work, we design a novel active learning strategy under the challenges of domain gap and semantic shift, which does not rely on any assumptions about the source-target label set relationship.

3. Our Approach

3.1. Problem Setting

In active universal domain adaptation, the learning algorithm has access to a labeled source domain $\mathcal{D}_S = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{D}_{UT} = \{(\mathbf{x}_i^t)\}_{i=1}^{n_t}$, which are respectively sampled from different distributions p_s and p_t. At each active learning round, the learning algorithm may query an oracle to obtain labels of n_r instances from $\mathcal{D}_{UT}$. After R rounds of active learning, n_b labeled target instances are added to the budget $\mathcal{D}_{LT} = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^{n_b}$ where $n_b = R \cdot n_r$. Besides, we use $\mathcal{C}_s$ to represent the label set of source domain while the label set of target domain is denoted as $\mathcal{C}_t$. $\mathcal{C}_c$=$\mathcal{C}_s \cap \mathcal{C}_t$ is the common label set shared by both domains. $\tilde{\mathcal{C}}_s$=$\mathcal{C}_s \backslash \mathcal{C}_c$ and $\tilde{\mathcal{C}}_t$=$\mathcal{C}_t \backslash \mathcal{C}_c$ respectively represent source private label set and target private label set. Note that the target label set $\mathcal{C}_t$ is inaccessible during training. The task of AUDA is to infer actual labels for all target instances no matter they are from $\mathcal{C}_c$ or $\tilde{\mathcal{C}}_t$.

3.2. Active Universal Adaptation Network

We propose an Active Universal Adaptation Network (AUAN) to address the AUDA task. The AUAN consists of a feature extractor G_f, a classifier G_c, a domain discriminator G_d, and prototype classifiers G_p, which are respectively parameterized by θ_f, θ_c, θ_d and θ_p. G_f is learned to generate discriminative representations for source and target samples. G_c aims to classify target "known" instances as source classes. G_d is trained adversarially to align source and target domain. G_p is designed to classify target "unknown" instances as target private classes, which maintains class representations (prototypes) in the target domain. During training, new prototypes will be dynamically added into G_p, once an instance with target private classes is annotated by active learning and its prototype is not stored in G_p.

The learning process mainly consists of three main parts at each training loop, as shown in Figure 2. The AUAN needs to be trained several loops. For simplicity, we take one training loop as an example to introduce our algorithm. During adversarial and diverse curriculum learning, we propose to train G_c and G_d as a curriculum learning style to progressively adapt G_c to the target domain. Besides, the model gradually learns the ability to identify

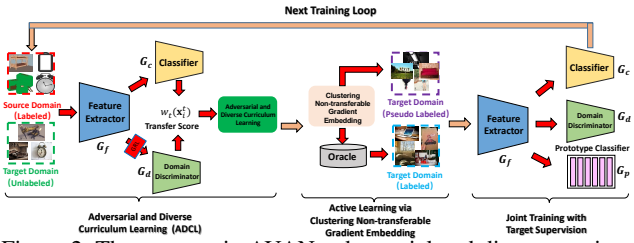


Figure 2. Three stages in AUAN: adversarial and diverse curriculum learning, active learning via clustering non-transferable gradient embedding and joint training with target supervision.

target "known"/"unknown" instances, which helps to annotate target informative instances during active learning. In the active learning stage, we propose a Clustering Non-transferable Gradient Embedding strategy which utilizes the clues of transferability, uncertainty and diversity. Target informative instances are selected for annotation. Meanwhile, high confident target "known" instances are assigned with pseudo labels predicted by G_c. Finally, all the labeled and pseudo labeled target data are used for further improving G_c in cross-domain alignment and learning G_p for target private classes. After several training loops of the three stages, the model can infer actual labels for all target instances.

3.2.1 Adversarial and Diverse Curriculum Learning

Due to the domain gap and semantic shift in AUDA, it is challenging to directly train a reliable model to recognize the "known"/"unknown" label for target instances and predict actual labels for target "known" instances. In addition, as G_c overfits the source domain, G_c may classify target "unknown" instances as source classes with high confidence. To alleviate the above problems subtly, motivated by curriculum learning [27], we select samples from easy to hard for cross-domain alignment and meanwhile, reduce the over-reliance of G_c on target "unknown" instances. Specifically, we design two curriculum losses, namely, an adversarial curriculum loss L_{adv} and a diverse curriculum loss L_{div}. The overall objective in ADCL is:

$$\begin{aligned} &\min_{\theta_f,\theta_c} L_c + L_{div} - L_{adv}, \\ &\max_{\theta_d} L_{adv}, \end{aligned} \tag{1}$$

where L_c is the standard cross-entropy classification loss calculated in the source domain. The min-max optimization is achieved by a gradient reverse layer [13]. Both L_{adv} and L_{div} are designed based on transfer score metric, which measures the transferability of a target sample $\mathbf{x}_i^t$. Given a target sample $\mathbf{x}_i^t$, its transfer score is a combination of two signals:

$$w_t(\mathbf{x}_i^t) = \max \bar{y}(\mathbf{x}_i^t) + d(\mathbf{x}_i^t), \tag{2}$$

where $\max \bar{y}(\mathbf{x}_i^t) \in [0,1]$, $d(\mathbf{x}_i^t) \in [0,1]$, and $w_t(\mathbf{x}_i^t) \in [0,2]$. The first term refers to the classification confidence that can manifest itself by the max value of classification probabilities, *i.e.*, $\bar{y}(\mathbf{x}_i^t) = G_c(G_f(\mathbf{x}_i^t))$. The second term is the similarity to the source domain, which can be estimated by the output of the domain discriminator, *i.e.*, $d(\mathbf{x}_i^t) = G_d(G_f(\mathbf{x}_i^t))$. A higher value $w_t(\mathbf{x}_i^t)$ indicates that $\mathbf{x}_i^t$ appears to be from the shared label set $\mathcal{C}_c$; otherwise $\mathbf{x}_i^t$ may be an "unknown" instance.

The adversarial curriculum loss L_{adv} aims for progressively aligning source and target samples from the common label set $\mathcal{C}_c$, as shown in Eq (3):

$$\begin{aligned} L_{adv} = &\mathbb{E}_{\mathbf{x}_i^s \in \mathcal{D}_S} \left[w_s(\mathbf{x}_i^s) \cdot \log(1 - G_d(G_f(\mathbf{x}_i^s))) \right] \\ &+ \mathbb{E}_{\mathbf{x}_i^t \in \mathcal{D}_{UT}} \left[\mathbb{1}_{w_t(\mathbf{x}_i^t) \geqslant w_\alpha(t)} \cdot \log(G_d(G_f(\mathbf{x}_i^t))) \right], \end{aligned} \tag{3}$$

where the indicator $\mathbb{1}_{w_t(\mathbf{x}_i^t) \geqslant w_\alpha(t)}$ in L_{adv} can select target samples $\mathbf{x}_i^t$ belonging to $\mathcal{C}_c$ from easy to hard by gradually reducing the value of $w_\alpha(t)$. The source weight $w_s(\mathbf{x}_i^s)$ aims to assign higher values for source samples from $\mathcal{C}_c$ and lower values for source samples from $\bar{\mathcal{C}}_s$, which can be reliably estimated by G_c's predictions on target samples from $\mathcal{C}_c$. First, we utilize the curriculum $w_t(\mathbf{x}_i^t) \geqslant w_\alpha(t)$ to select target samples from $\mathcal{C}_c$, which should have higher classification probability (predicted by G_c) on the shared categories than source private categories. Then, we can get the G_c's predictions on the selected target samples, and calculate the average classification probabilities $\mathbf{V}$, *i.e.*, $\mathbf{V} = \text{avg}_{w_t(\mathbf{x}_i^t) \geqslant w_\alpha(t)} G_c(G_f(\mathbf{x}_i^t))$. Note that categories with higher values in $\mathbf{V}$ are probably the shared categories while those with lower values are likely to be source-private categories. Therefore, $\mathbf{V}$ can be used to calculate the weight of a source sample $(\mathbf{x}_i^s, \mathbf{y}_i^s)$, *i.e.*, $w_s(\mathbf{x}_i^s) = \mathbf{V}_{\mathbf{y}_i^s}$, where $\mathbf{y}_i^s$ is used as the index of $\mathbf{V}$. Note that source samples with the same category label are assigned with the same weight.

To gradually reduce the over-reliance of classifier G_c, the diverse curriculum loss L_{div} defined in Eq (4) utilizes the indicator $\mathbb{1}_{w_t(\mathbf{x}_i^t) < w_\alpha(t)}$ to select target "unknown" samples and enforces these selected samples to be uniformly distributed across different classes in $\mathcal{C}_s$ by minimizing the negative entropy of G_c's predictions.

$$L_{div} = \mathbb{E}_{\mathbf{x}_i^t \sim p_t} [\mathbb{1}_{w_t(\mathbf{x}_i^t) < w_\alpha(t)} \cdot -H(G_c(G_f(\mathbf{x}_i^t)))], \tag{4}$$

where $H(\cdot)$ is the entropy function. With the cooperation between adversarial curriculum loss and diverse curriculum loss, the model can progressively and reliably predict the "known"/"unknown" label for target instances and classify target "known" instances.

3.2.2 Active Learning via Clustering Non-transferable Gradient Embedding

We hope the model can find out the most informative instances in the unlabeled target dataset $\mathcal{D}_{UT}$ and query their labeling information from an oracle to construct the labeled target dataset $\mathcal{D}_{LT}$. The informative target instances, intuitively, are the *ones most different from what the model has already known.* Previous AL strategies focus on finding instances that are highly uncertain or diverse, which is suboptimal for AUDA. The informative instances in AUDA

should satisfy the following conditions: (1) Similar to traditional AL, the selected instances should also be highly uncertain and diverse. (2) The selected instances should be target "unknown" samples and their actual labels are from the target private label set, making it possible to learn prototype classifiers for target private classes. To satisfy the above requirements, we propose to perform active learning by clustering Non-transferable Gradient Embedding (CNTGE), which utilizes the clues of transferability, uncertainty and diversity.

Transferability. To accurately select target samples of target private classes for annotation, we should firstly remove target "known" samples from $\mathcal{D}_{UT}$ and perform active learning on the remaining unlabeled data. To achieve this goal, we first run the K-means algorithm [1] on all the unlabeled target features $\{\mathbf{f}_i^{ut}|\mathbf{f}_i^{ut} = G_f(\mathbf{x}_i^t), \mathbf{x}_i^t \in \mathcal{D}_{UT}\}$ to obtain n_r centroids $\{\mathbf{u}_i\}_{i=1}^{n_r}$. Then, we calculate the transfer scores $w_t(\mathbf{u}_i)$ (Eq (2)) of these centroids. We assume that a cluster's category is from the common label set $\mathcal{C}_c$ and samples belonging to the cluster are transferable if $w_t(\mathbf{u}_i) > \beta$, otherwise, the cluster's category is from the target private label set $\tilde{\mathcal{C}}_t$ and its samples are non-transferable. As for clusters with $w_t(\mathbf{u}_i) > \beta$, we can construct pseudo labeled target dataset $\mathcal{D}_{PLT}$ without any annotation cost, *i.e.*, $\mathcal{D}_{PLT} = \{(\mathbf{x}_{ij}^t, \tilde{\mathbf{y}}_i^t)|w_t(\mathbf{u}_i) > \beta, \tilde{\mathbf{y}}_i^t = \arg\max G_c(\mathbf{u}_i), i = 1 \cdots, n_r\}$ where $\mathbf{u}_i$ is the clustering centroid of $\mathbf{x}_{ij}^t$. As for the rest of the target unlabeled samples, their labels are most likely from target private label set $\tilde{\mathcal{C}}_t$, These non-transferable instances will be used as query candidates $\mathcal{D}_{NT} = \mathcal{D}_{UT} \backslash \mathcal{D}_{PLT}$ for active learning.

Uncertainty and Diversity: Clustering Gradient Embeddings of Non-transferable Instances. To jointly capture both uncertainty and diversity of non-transferable instances in $\mathcal{D}_{NT}$, we aim to select n_r target instances to query their labels from an oracle and add the selected instances into the target labeled dataset $\mathcal{D}_{LT}$ at each active learning round. Specifically, we firstly compute the gradient embeddings [2] for all non-transferable instances $\mathcal{D}_{NT}$. Note that the magnitude of a gradient vector captures the uncertainty of the model on the instance: if the model is highly certain about the instance's label, the norm of the instance's gradient embedding is small, and vice versa for samples where the model is uncertain. Then, n_r diverse high-magnitude samples are selected. It is impossible to make sure all instances in $\mathcal{D}_{LT}$ are with labels in $\tilde{\mathcal{C}}_t$, and some of them probably are with labels in $\mathcal{C}_c$. Even so, they are helpful for promoting the adaptation process.

3.2.3 Joint Training with Target Supervision

After the active learning process, two types of target supervision are provided: $\mathcal{D}_{PLT}$ and $\mathcal{D}_{LT}$. These annotated target instances will be leveraged to further promote the functions of ADCL and learn G_p for inferring target "unknown" instances. To promote ADCL, instances in $\mathcal{D}_{LT}$ from source classes should join in the learning of classifier G_c and cross-domain adversarial training while instances in $\mathcal{D}_{LT}$ from target private classes should help to reduce G_c's over-reliance. Therefore, the classification loss L_c, adversarial curriculum loss L_{adv} (Eq (3)) and diverse curriculum loss L_{div} (Eq (4)) are re-formulated, as shown in Eq (5).

$$\begin{aligned}
\tilde{L}_c &= L_c + \mathbb{E}_{(\mathbf{x}_i^t, \mathbf{y}_i^t) \in \mathcal{D}_{LT}, \mathbf{y}_i^t \in \mathcal{C}_s} \left[L_{ce}(\mathbf{y}_i^t, G_c(G_f(\mathbf{x}_i^t))) \right], \\
\tilde{L}_{adv} &= L_{adv} + \mathbb{E}_{(\mathbf{x}_i^t, \mathbf{y}_i^t) \in \mathcal{D}_{LT}, \mathbf{y}_i^t \in \mathcal{C}_s} \left[\log(G_d(G_f(\mathbf{x}_i^t))) \right], \\
\tilde{L}_{div} &= L_{div} + \mathbb{E}_{(\mathbf{x}_i^t, \mathbf{y}_i^t) \in \mathcal{D}_{LT}, \mathbf{y}_i^t \notin \mathcal{C}_s} [-H(G_c(G_f(\mathbf{x}_i^t)))].
\end{aligned} \tag{5}$$

Since $\mathcal{D}_{PLT}$ contains noisy labels, for accurate cross-domain alignment, we only leverage $\mathcal{D}_{LT}$ to enforce the adaptation process.

As $\mathcal{D}_{LT}$ is too small to discriminatively learn the prototype classifiers G_p, we leverage $\mathcal{D}_{PLT}$ to serve as complementary cues in the learning process. To further improve the instance-level discriminative power for all the target samples, we are motivated to cluster target features in $\mathcal{D}_{UT}$ with its neighbors (labeled target features or prototypes) by a self-supervised cluster objective L_{nc}. Thus, similar features could cluster together and G_p could make more reliable predictions. The overall objective to learn G_p is:

$$\min_{\theta_f, \theta_p} L_p + L_{nc}, \tag{6}$$

where the classification loss L_p is defined as Eq (7):

$$\begin{aligned}
L_p = &\mathbb{E}_{(\mathbf{x}_i^t, \tilde{\mathbf{y}}_i^t) \in \mathcal{D}_{PLT}} \left[L_{ce}(\tilde{\mathbf{y}}_i^t, G_p(G_f(\mathbf{x}_i^t))) \right] \\
&+ \mathbb{E}_{(\mathbf{x}_i^t, \mathbf{y}_i^t) \in \mathcal{D}_{LT}} \left[L_{ce}(\mathbf{y}_i^t, G_p(G_f(\mathbf{x}_i^t))) \right],
\end{aligned} \tag{7}$$

where L_{ce} is the cross-entropy loss.

To cluster target features in $\mathcal{D}_{UT}$ to meaningful neighbors, we propose to calculate a self-supervised cluster loss L_{nc}. Here, the meaningful neighbors are labeled target samples or prototypes in G_p. Firstly, in a mini-batch, we calculate the similarity of unlabeled target samples $\mathbf{f}_i^{ut}$ in $\mathcal{D}_{UT}$ to all labeled target samples $\{\mathbf{f}_i^{lt}|\mathbf{f}_i^{lt} = G_f(\mathbf{x}_i^t), \mathbf{x}_i^t \in \mathcal{D}_{LT}\}$ and K prototypes $\{\mathbf{w}_1, \cdots, \mathbf{w}_k, \cdots, \mathbf{w}_K\}$ in G_p. Since a mini-batch data cannot contain all the labeled target samples, we construct a memory bank $\mathbf{M} \in R^{(n_r+K)\times d}$ to store all the labeled target samples and prototypes, *i.e.*, $\mathbf{M} = [\mathbf{f}_1^{lt}, \cdots, \mathbf{f}_i^{lt}, \cdots \mathbf{f}_{n_r}^{lt}, \mathbf{w}_1, \cdots, \mathbf{w}_k, \cdots, \mathbf{w}_K]$ where $\mathbf{f}_i^{lt}$ and $\mathbf{w}_k$ are L2-normalized. Because G_f and G_p are updated at each training step, $\mathbf{M}$ is updated with mini-batch data by replacing the older ones with the updated ones. Let $\mathbf{M}_j$ denotes the j-th item in $\mathbf{M}$. Then, the probability that a target feature $\mathbf{f}_i^{ut}$ in $\mathcal{D}_{UT}$ is a neighbor of $\mathbf{M}_j$ is:

$$p_{i,j} = \frac{\exp(\mathbf{M}_j^T \mathbf{f}_i^{ut}/\tau)}{Z_i} \text{ and } Z_i = \sum_{\mathbf{M}_j \in \mathbf{M}} \exp(\mathbf{M}_j^T \mathbf{f}_i^{ut}/\tau), \tag{8}$$

where τ is the temperature parameter. Then, the entropy-based clustering loss is calculated as:

$$L_{nc} = \mathbb{E}_{\mathbf{x}_i^t \in \mathcal{D}_{UT}} \left[\sum\nolimits_j -p_{i,j} \log(p_{i,j}) \right]. \tag{9}$$

Algorithm 1: Active Universal Adaptation Network

Require: Feature extractor G_f, classifier G_c, domain discriminator G_d and prototype classifier G_p, parameterized by θ_f, θ_c, θ_d and θ_p respectively, labeled source domain $\mathcal{D}_S$, unlabeled target domain $\mathcal{D}_{UT}$, total rounds R, per-round budget n_r.
Define: target labeled dataset $\mathcal{D}_{LT} = \emptyset$, target pseudo labeled dataset $\mathcal{D}_{PLT} = \emptyset$. G_p contains no prototypes in the beginning.
Warm Up: Solve Eq (1) with $\mathcal{D}_S$ and $\mathcal{D}_{UT}$.
for $r = 0$ *to* R **do**
    # AL: Clustering Non-transferable Gradient Embedding:
    $\mathcal{D}_{UT} = \mathcal{D}_{UT} \backslash \mathcal{D}_{LT}$
    For all instances in $\mathcal{D}_{UT}$:
    1. Run K-Means and calculate the transfer scores of n_r centroids.
    2. Construct target pseudo dataset $\mathcal{D}_{PLT}$.
    3. Compute gradient embedding [2] on $\mathcal{D}_{NT} = \mathcal{D}_{UT} \backslash \mathcal{D}_{PLT}$, query n_r instances' labels, finally add them into $\mathcal{D}_{LT}$, and finally add new prototypes to G_p.
    # Joint Training AUAN via Adversarial and Diverse Curriculum Learning:
    Solve Eq (10) with $\mathcal{D}_S$, $\mathcal{D}_{UT}$, $\mathcal{D}_{LT}$ and $\mathcal{D}_{PLT}$.
end
Output: Model parameters: θ_f, θ_c, θ_d and θ_p.

Finally, The overall objective to learn the active universal adaptation network is shown as Eq (10)

$$\begin{aligned} &\min_{\theta_f,\theta_c} \tilde{L}_c + \tilde{L}_{div} - \tilde{L}_{adv}, \\ &\max_{\theta_d} \tilde{L}_{adv}, \\ &\min_{\theta_f,\theta_p} L_p + L_{nc}, \end{aligned} \tag{10}$$

where the parameters are optimized as an alternate style. Algorithm 1 shows the processes of training and active learning. Once the model is trained, we can leverage AUAN model to classify all target instances according to Eq (11):

$$y(\mathbf{x}_i^t) = \begin{cases} \arg\max G_c(G_f(\mathbf{x}_i^t)) & w_t(\mathbf{x}_i^t) > w_0 \\ \arg\max G_p(G_f(\mathbf{x}_i^t)) & \text{otherwise} \end{cases} \tag{11}$$

4. Experiments

In this section, we first illustrate datasets, compared methods, evaluation protocols, and implementation details. Then, we show extensive experimental results and analysis. Due to the limited space, more results and analysis can be found in the **supplementary material**.

4.1. Setup

Datasets. The first dataset **Office-Home** [56] contains four domains: Art (Ar), Clipart (Cl), Product (Pr) and Real-World (Rw) across 65 classes. In the alphabet order, we use the first 10 classes as $\mathcal{C}_c$, the next 5 classes as $\tilde{\mathcal{C}}_s$ and the rest as $\tilde{\mathcal{C}}_t$. The second dataset **VisDA** [39] contains 12 classes from two domains: synthetic (S) and real (R) images. The class numbers of $\mathcal{C}_c$, $\tilde{\mathcal{C}}_s$ and $\tilde{\mathcal{C}}_t$ are respectively 6, 5 and 3. The third dataset is **Office-31** [44], which contains three domains (Amazon (A), DSLR (D), Webcam (W)) and 31 classes. The class numbers of $\mathcal{C}_c$, $\tilde{\mathcal{C}}_s$ and $\tilde{\mathcal{C}}_t$ are respectively 10, 11 and 10. The forth dataset is **DomainNet** [38], which contains six domains: Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real (R) and Sketch (S) across 345 classes. The class numbers of $\mathcal{C}_c$, $\tilde{\mathcal{C}}_s$ and $\tilde{\mathcal{C}}_t$ are respectively 150, 50 and 145. Following [11], We choose 3 domains in the DomainNet dataset to transfer between each other. For a fair comparison, all dataset partitions follow the universal domain adaptation [61]. We set the per-round budget as 21 for office-home, 10 for office31, 100 for VisDA and 115 for DomainNet, and perform 15 rounds of active learning.

Table 1. The average class accuracy (%) on Office-Home, Office-31, VisDA and DomainNet datasets for different DA methods equipped with different AL strategies. The best results are bolded.

DA \ AL	Random	Margin	Coreset	BADGE	AVG
			Office-Home		
ResNet	26.32	28.06	30.42	28.35	28.29
UAN	32.58	32.58	32.77	33.86	32.95
ADCL	**47.19**	**46.86**	**45.73**	**47.94**	**47.33**
			Office-31		
ResNet	75.67	76.37	77.97	77.29	76.83
UAN	64.56	60.56	65.43	61.96	63.13
ADCL	**79.15**	**78.55**	**80.79**	**80.51**	**79.75**
			VisDA		
ResNet	59.27	61.98	61.43	61.91	61.15
UAN	59.11	72.45	57.28	62.83	62.92
ADCL	**63.15**	**63.49**	**62.58**	**64.00**	**63.31**
			DomainNet		
ResNet	27.43	29.07	28.91	30.68	29.02
UAN	34.12	34.91	35.47	35.90	35.10
ADCL	**37.54**	**37.37**	**34.34**	**37.09**	**36.59**

Compared Methods. As the existing UDA methods cannot handle the new AUDA task, we extend two domain adaptation baselines to the AUDA setting, *i.e.*, ResNet [19] and UAN [61] equipped with state-of-the-art active learning approaches. To compare four types of active learning strategies, we select the following seven approaches: (1) Random: The naive baseline that randomly selects several instances to annotate labels at each round. (2) Uncertainty: a) Entropy [58]: Sampling instances over which the model has high predictive entropy. b) Margin [43]: Sampling instances for which the score between the model's top-2 predictions is the smallest. c) Confidence [58]: Sampling instances for which the predictive confidence is the lowest. (3) Diversity: a) K-means: K-means is performed at each round and one sample closest to its centroid is selected for each cluster. b) Coreset [49]: Sampling instances that geometrically cover data distributions. (4) Mixture of Uncertainty and Diversity: BADGE [2]: Sampling instances that are disparate and high magnitude when presented in a hallucinated gradient space.

Evaluation Protocols. We report the average class accuracy for comparison. Specifically, we firstly calculate the classification accuracy for each category in the target domain and finally average them. Besides, the curves of average class accuracy with the annotation round increasing are drawn for comparing different active learning strategies.

Implementation Details. We use Pytorch [37] for our implementation. Following the UAN [61], ResNet-50 [19] is used as the feature extractor. A bottleneck layer with 256 units followed by a classifier and a domain discriminator, is added after the feature extractor. Another bottleneck layer

Table 2. Average class accuracy (%) at 5th, 10th and 15th annotation round on Office-Home, Office-31, VisDA and DomainNet datasets for comparing different active learning strategies. The best results are bolded.

Office-Home																					
AL Strategy	**Ar→Cl**			**Ar→Pr**			**Ar→Rw**			**Cl→Ar**			**Cl→Pr**			**Cl→Rw**			**Pr→Ar**		
	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th
Random	21.36	26.34	29.33	50.58	54.09	60.05	42.67	46.03	48.32	34.48	41.51	47.27	48.53	53.06	56.53	42.10	45.99	47.20	39.10	45.63	52.78
Entropy	20.01	24.51	27.19	45.32	51.71	55.41	41.98	46.17	48.48	31.56	37.94	44.62	39.84	45.94	51.52	41.73	45.12	45.44	32.55	41.19	47.86
Confidence	18.59	26.72	29.36	46.99	52.66	55.81	42.06	46.35	47.41	35.52	42.31	43.89	45.98	49.25	52.20	42.17	47.15	50.10	36.08	41.44	46.10
K-means	21.68	25.27	27.45	46.53	50.37	53.17	40.40	41.24	42.71	34.17	37.79	41.36	43.51	47.21	48.09	39.08	41.17	41.54	38.69	41.41	46.54
Margin	24.60	26.37	29.49	50.63	56.31	58.49	49.24	49.95	51.36	36.93	42.35	44.79	47.41	52.62	56.35	42.15	46.63	47.73	37.43	45.34	50.11
Coreset	25.04	26.58	26.94	49.94	53.44	56.77	45.97	46.99	48.89	39.11	42.01	44.29	47.78	48.45	52.94	44.09	45.30	48.50	40.33	46.33	51.47
BADGE	22.28	28.45	30.53	51.03	55.81	58.44	43.41	48.05	49.79	32.60	39.66	44.59	49.07	54.25	58.63	42.20	45.58	48.00	**41.14**	47.86	**54.68**
CNTGE (Ours)	**27.25**	**32.44**	**36.51**	**56.02**	**63.58**	**67.96**	**48.39**	**57.56**	**61.02**	**40.89**	**50.20**	**53.75**	**51.57**	**61.58**	**65.82**	**46.49**	**56.03**	**61.88**	38.50	**49.49**	54.26

Office-Home																			VisDA		
AL Strategy	**Pr→Cl**			**Pr→Rw**			**Rw→Ar**			**Rw→Cl**			**Rw→Pr**			**AVG**			**S→R**		
	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th
Random	21.47	24.80	29.25	50.80	54.00	58.68	38.50	46.78	52.11	18.16	21.67	25.01	51.59	55.83	61.77	38.28	42.98	46.86	62.72	63.15	63.15
Entropy	17.01	21.58	22.95	45.15	50.11	54.75	33.31	38.93	45.95	17.12	20.47	22.38	45.18	49.82	53.01	34.23	39.46	43.30	57.46	57.97	60.06
Confidence	19.72	23.47	24.54	47.95	53.59	57.96	35.34	43.32	45.51	17.41	19.43	23.74	48.21	52.51	57.44	36.34	41.52	44.50	58.77	60.64	61.17
K-means	22.97	27.58	28.95	44.68	47.44	49.68	37.88	40.96	46.13	20.01	22.76	22.94	45.27	51.35	53.49	36.24	39.55	41.84	64.40	64.59	64.59
Margin	23.84	28.26	30.02	50.74	56.72	60.89	39.19	44.74	49.26	19.98	22.64	22.73	56.16	59.34	61.06	39.86	44.27	46.86	62.05	63.49	63.49
Coreset	24.21	27.34	28.00	51.51	58.47	59.10	41.10	45.25	50.17	22.63	22.68	22.68	50.89	55.54	58.99	40.22	43.20	45.73	62.24	62.24	62.58
BADGE	23.43	28.59	29.91	49.21	58.96	61.01	38.67	47.69	51.79	18.35	23.39	25.64	49.27	58.45	62.28	38.39	44.73	47.94	63.76	64.00	64.00
CNTGE (Ours)	**28.49**	**34.38**	**39.11**	**55.02**	**64.44**	**69.76**	**40.96**	**50.93**	**55.03**	**25.65**	**31.13**	**36.55**	**56.18**	**64.56**	**69.24**	**42.95**	**51.36**	**55.91**	**71.32**	**73.23**	**73.91**

Office-31																					
AL Strategy	**A→D**			**A→W**			**D→A**			**D→W**			**W→A**			**W→D**			**AVG**		
	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th
Random	80.75	82.78	85.23	71.58	79.30	84.22	59.80	63.01	64.90	81.65	88.06	91.35	58.69	61.38	62.84	83.29	84.09	86.33	72.63	76.43	79.15
Entropy	75.54	80.08	82.98	69.11	75.75	80.75	49.34	54.63	54.63	82.55	87.75	90.47	48.60	50.87	55.36	82.61	83.99	85.86	67.96	72.18	75.01
Confidence	74.65	80.38	81.97	72.93	76.19	81.06	52.06	54.04	54.04	87.17	90.01	90.82	48.63	54.86	56.62	84.02	85.43	87.73	69.91	73.48	75.37
K-means	76.96	79.85	82.34	70.20	73.39	78.83	54.65	58.84	61.46	75.51	78.86	86.72	54.01	56.64	58.88	83.41	84.16	86.57	69.12	71.96	75.80
Margin	80.50	83.34	83.75	75.37	81.46	82.11	58.79	62.46	66.31	86.35	90.15	91.27	53.55	57.34	61.55	85.38	86.32	86.32	73.32	76.84	78.55
Coreset	**81.08**	84.38	86.18	76.77	**83.64**	86.10	57.19	61.84	66.33	85.84	**92.01**	**92.76**	58.43	60.73	65.21	83.51	86.28	88.17	73.80	78.15	80.79
BADGE	79.90	84.12	84.45	78.21	81.58	86.16	58.07	64.28	**68.77**	87.77	90.43	91.34	58.12	62.16	65.74	84.49	85.27	86.59	74.43	77.97	80.51
CNTGE (Ours)	79.81	**85.48**	**87.21**	**80.53**	83.05	**86.19**	**61.91**	**67.50**	68.13	**90.91**	91.13	92.26	**62.20**	**65.50**	**65.67**	**86.38**	**86.95**	**87.60**	**76.96**	**79.94**	**81.18**

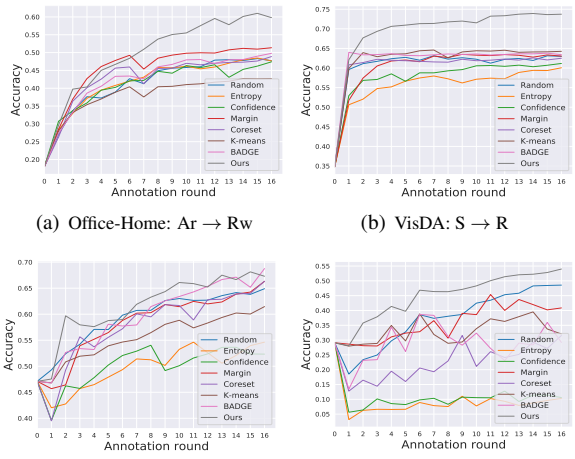
DomainNet																					
AL Strategy	**P→R**			**R→P**			**P→S**			**S→P**			**R→S**			**S→R**			**AVG**		
	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th	5th	10th	15th
Random	38.57	43.46	48.56	29.92	34.03	36.24	24.72	28.11	29.53	28.86	32.16	33.98	24.51	28.09	29.49	42.01	**46.56**	47.46	31.43	35.40	37.54
Entropy	8.92	11.05	11.05	25.61	27.99	29.29	22.23	24.90	26.85	23.99	25.63	27.64	21.82	24.98	26.96	28.69	29.66	30.43	21.88	24.04	25.37
Confidence	32.83	45.38	46.69	29.99	33.29	36.57	25.43	**29.06**	31.11	28.23	30.83	34.05	24.83	28.50	29.93	39.39	45.24	45.90	30.12	35.38	37.37
K-means	10.14	10.73	12.77	28.16	29.47	30.74	22.35	25.92	28.71	26.27	26.86	28.88	22.67	25.82	28.14	30.94	31.83	32.67	23.42	25.10	26.99
Margin	38.85	38.85	39.52	30.73	32.07	34.92	**26.47**	27.98	29.43	28.78	30.07	32.31	24.70	26.36	28.26	40.11	44.25	46.42	31.60	33.26	35.14
Coreset	20.57	31.55	31.74	28.75	31.67	34.57	24.30	28.03	29.45	28.99	32.39	33.77	24.17	27.39	29.09	38.46	43.35	47.42	27.54	32.40	34.34
BADGE	23.16	38.66	40.31	30.22	33.71	37.31	25.09	28.33	30.69	29.83	32.01	35.70	25.14	27.94	31.13	40.99	45.09	47.38	29.07	34.29	37.09
CNTGE (Ours)	**46.85**	**50.06**	**54.00**	**30.86**	**34.43**	**38.12**	24.84	28.94	**31.55**	**30.54**	**32.22**	**36.71**	**25.34**	**28.72**	**32.35**	**45.02**	45.87	**48.77**	**33.91**	**36.71**	**40.25**

with 256 units embeds features extracted by G_f to learn prototype classifiers. The optimization setting follows [13]. The margin function is set as $w_\alpha(t) = w_0 + (1 - \frac{t}{T}) \cdot \alpha$ where $w_0 = 1.0$, t is the t-th training step and T is the total training iterations. The hyper-parameters are tuned with cross-validation [62], and fixed for each dataset, *i.e.*, $\alpha = 0.2, \beta = 1.5, \tau = 0.05$. More details are illustrated in the supplementary material.

4.2. Comparative Results

Comparison agaist different domain adaptation methods. To justify the effectiveness of our proposed ADCL for AUDA, we extend two domain adaptation baselines, *i.e.*, ResNet and UAN, to the AUDA setting by learning prototype classifiers. Similar to our method, the prototype classifiers in baselines are learned with $\mathcal{D}_{LT}$ by optimizing Eq 6. We construct different combinations between domain adaptation models (ResNet, UAN, and our proposed ADCL) and active learning strategies (Random, Margin, Coreset, and BADGE) for comparison. The average class accuracy results are shown in Table 1. We can observe that ADCL performs the best when equipped with different AL strategies, especially, outperforms UAN which also deals with the domain gap and semantic shift problems. The results support that the proposed ADCL can effectively alleviate the negative impact of domain gap and semantic shift, and helps AL strategies annotate informative instances to infer actual labels for all target instances.

Comparison with different active learning strategies. To evaluate the effectiveness of our proposed CNTGE strategy, we consider fixing the domain adaptation method as ADCL and varying the active learning methods (seven prior work) for comparison. As shown in Table 2, we report the average class accuracy on Office-Home, Office-31, VisDA and DomainNet datasets at the 5th, 10th and 15th annotation round for conciseness. Besides, the full performance curves of some hard transfer tasks are shown in Figure 3. Our proposed AL strategy CNTGE performs the best on most tasks or the second on a few tasks, which proves that target instances annotated by CNTGE are more informative than those annotated by other methods under domain gap and semantic shift. More importantly, CNTGE performs well in DomainNet, indicating that CNTGE is robust to large dataset with plenty of categories. In particular, we have some key observations. (1) In the practical AUDA setting, especially in difficult transfer tasks in the Office-Home, VisDA and DomainNet datasets, some traditional AL methods perform similarly to or even worse than Random. A possible reason is that the uncertainty or diversity are wrongly estimated by traditional AL methods due to



(a) Office-Home: Ar → Rw (b) VisDA: S → R

(c) Office-31: D → A (d) DomainNet: P → R

Figure 3. Average class accuracy across four hard transfer tasks from Office-Home, Office-31, VisDA and DomainNet datasets.

the violation of the assumption about the source-target label set relationship in AUDA. These traditional AL approaches may easily lead to sampling outliers, redundant instances, or uninformative instances from source classes, which is not beneficial for inferring labels for all target instances or even damages the classification performance. (2) Coreset tries to select instances geometrically matching the data distributions for annotations, and it performs the best in a few transfer tasks where CNTGE underperforms Coreset. However, Coreset only works well in small datasets, which is not robust enough since practical datasets are usually large. Fortunately, CNTGE is more practical and performs very well on large datasets such as Office-Home, VisDA and DomainNet.

4.3. Ablation Studies

Ablation studies on ADCL. To analyze the efficiency of the proposed adversarial curriculum loss L_{adv} and diverse curriculum loss L_{div}, we derive two variants: (1) w/o L_{adv} is the variant by replacing L_{adv} with naive adversarial loss [13] which is widely used in adversarial domain adaptation. The native adversarial loss can be obtained by removing $w_s(\mathbf{x}_i^s)$ and indicator function $\mathbb{1}_{w_t(\mathbf{x}_i^t) \geqslant w_\alpha(t)}$ in Eq (3). (2) w/o L_{div} is the variant learned without loss L_{div}. All other loss functions remain the same as AUAN. Compared with AUAN in Table 3, the average performance drop of w/o L_{adv} and w/o L_{div} are respectively 2.34% and 9.3%. It indicates that L_{adv} could effectively constrain the cross-domain alignment into the shared common label set. Besides, L_{div} could reduce the over-reliance of classifier G_c on target "unknown" samples, which promotes CNTGE to select more informative target instances for active learning.

Ablation studies on learning G_p. Two variants are proposed to study the effectiveness of L_p and L_{nc} on learning the prototype classifiers: (1) w/o L_p is the variant where the prototype classifiers are learned without loss L_p. In this case, the prototype classifiers cannot be well learned, and we apply a KNN classifier to infer labels for target unknown instances. (2) w/o L_{nc} is the variant where the prototype classifiers are learned without loss L_{nc}. As shown in Table 3, the w/o L_p and w/o L_{nc} both underperform AUAN. The performance drop of w/o L_p and w/o L_{nc} are respectively 3.21% and 1.96%. It implies that L_p and L_{nc} are beneficial for learning to infer labels in $\tilde{\mathcal{C}}_t$ with limited data.

Table 3. Ablation studies on Office-Home (6 challenging tasks).

Variant	Ar → Rw	Cl→Rw	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	AVG
AUAN	**61.02**	**61.88**	**69.76**	**55.03**	**36.55**	**69.24**	**58.91**
w/o L_{adv}	57.40	56.79	68.46	54.35	34.36	68.09	$56.57_{\downarrow 2.34}$
w/o L_{div}	50.40	43.78	50.44	54.35	34.37	64.31	$49.61_{\downarrow 9.30}$
w/o L_p	58.28	58.06	64.74	52.97	31.63	68.56	$55.71_{\downarrow 3.21}$
w/o L_{nc}	60.55	59.21	66.58	54.19	33.77	67.42	$56.95_{\downarrow 1.96}$
AUAN-1	55.18	56.12	65.11	52.58	32.77	66.46	$54.70_{\downarrow 4.21}$
AUAN-2	60.28	60.44	68.08	53.88	34.62	68.73	$57.67_{\downarrow 1.24}$

Effect of $\mathcal{D}_{LT}$ during adaptation. To testy whether $\mathcal{D}_{LT}$ helps to suppress the negative impact of domain gap and semantic shift, we design the AUAN-1 model which is optimized by Eq (1) and Eq (6) while the original AUAN is optimized by Eq (5) and Eq (6). The average performance of AUAN-1 drops 4.21%, as shown in Table 3, indicating that $\mathcal{D}_{LT}$ could enforce the adaptation process and narrow the domain gap and semantic shift.

Effect of $\mathcal{D}_{PLT}$ when learning G_p. To study the effectiveness of learning G_p with $\mathcal{D}_{PLT}$, we derive the AUAN-2 model which is learned without the first term in Eq (7). Results are shown in Table 3. The 1.24 % performance drop of AUAN-2 illustrates that although $\mathcal{D}_{PLT}$ contains noisy labels, $\mathcal{D}_{PLT}$ could cluster similar instances and assist to learn discriminative prototypes.

5. Conclusion

In this paper, we propose a novel paradigm for unsupervised domain adaptation, termed as Active Universal Domain Adaptation (AUDA), which extends the applicability of domain adaptation in practical scenarios. An active universal adaptation network equipped with ADCL and CNTGE is proposed to address this issue. Extensive experiments show the effectiveness of our model. In the future, we will design AL strategies that consider the distribution information of known and unknown samples and utilize the knowledge graph for unknown category inference.

Acknowledgements

This work was supported by the National Key Research & Development Plan of China under Grant 2020AAA0106200, in part by the National Natural Science Foundation of China under Grants 62036012, 61721004, 62072286, 61720106006, 61832002, 62072455, 62002355, U1836220, and U1705262, in part by the Key Research Program of Frontier Sciences of CAS under Grant QYZDJSS-WJSC039, and in part by Beijing Natural Science Foundation (L201001).

References

[1] David Arthur and Sergei Vassilvitskii. k-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*, 2019.

[3] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[4] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *CVPR*, pages 2724–2732, 2018.

[5] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *ECCV*, pages 135–150, 2018.

[6] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *CVPR*, pages 2985–2994, 2019.

[7] Rita Chattopadhyay, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Joint transfer and batch-mode active learning. In *ICML*, pages 253–261, 2013.

[8] Zhihong Chen, Chao Chen, Zhaowei Cheng, Boyuan Jiang, Ke Fang, and Xinyu Jin. Selective transfer with reinforced transfer network for partial domain adaptation. In *CVPR*, pages 12706–12714, 2020.

[9] Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu. Graph adaptive knowledge transfer for unsupervised domain adaptation. In *ECCV*, pages 37–52, 2018.

[10] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

[11] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *ECCV*, pages 567–583, 2020.

[12] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *ICML*, pages 1183–1192, 2017.

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

[14] Yonatan Geifman and Ran El-Yaniv. Deep active learning over the long tail. *arXiv preprint arXiv:1711.00941*, 2017.

[15] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

[16] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[17] Philip Haeusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. Associative domain adaptation. In *ICCV*, pages 2765–2773, 2017.

[18] Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2-3):131–309, 2014.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[20] Wei-Ning Hsu and Hsuan-Tien Lin. Active learning by learning. In *AAAI*, pages 2659–2665, 2015.

[21] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *CVPR*, pages 1498–1507, 2018.

[22] Sheng-Wei Huang, Che-Tsung Lin, Shu-Ping Chen, Yen-Yi Wu, Po-Hao Hsu, and Shang-Hong Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *ECCV*, pages 718–731, 2018.

[23] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, pages 4893–4902, 2019.

[24] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *NeurIPS*, pages 7024–7035, 2019.

[25] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Workshop*, 2015.

[26] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. In *NeurIPS*, pages 9345–9356, 2018.

[27] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NeurIPS*, page 1189–1197, 2010.

[28] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*, pages 2927–2936, 2019.

[29] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *CVPR*, pages 8867–8876, 2018.

[30] Mingsheng Long, Guiguang Ding, Jianmin Wang, Jiaguang Sun, Yuchen Guo, and Philip S Yu. Transfer sparse coding for robust image representation. In *CVPR*, pages 407–414, 2013.

[31] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, pages 136–144, 2016.

[32] Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *CVPR*, pages 8266–8276, 2019.

[33] Tzu Ming Harry Hsu, Wei Yu Chen, Cheng-An Hou, Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Unsupervised domain adaptation with imbalanced cross-domain data. In *ICCV*, pages 4121–4129, 2015.

[34] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's

uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, pages 13969–13980, 2019.

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

[36] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *ICCV*, pages 754–763, 2017.

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

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

[39] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *CVPR Workshops*, pages 2102–2105, 2018.

[40] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. 2009.

[41] Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. Domain adaptation meets active learning. In *NAACL Workshop*, pages 27–32, 2010.

[42] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[43] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *ECML*, pages 413–424, 2006.

[44] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.

[45] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self-supervision. In *NeurIPS*, pages 16282–16292, 2020.

[46] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *ECCV*, pages 153–168, 2018.

[47] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.

[48] Greg Schon. Less is more: Active learning with support vector machines. In *ICML*, pages 839–846, 2000.

[49] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

[50] Burr Settles. Active learning literature survey. 2009.

[51] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *ICCV*, pages 5972–5981, 2019.

[52] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087, 2017.

[53] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *WACV*, pages 739–748, 2020.

[54] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(11):45–66, 2001.

[55] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 7167–7176, 2017.

[56] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5385–5394, 2017.

[57] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, pages 3630–3638, 2016.

[58] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *IJCNN*, pages 112–119, 2014.

[59] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, pages 5419–5428, 2018.

[60] Ting Yao, Chong-Wah Ngo, and Shiai Zhu. Predicting domain adaptivity: Redo or recycle? In *ACM MM*, pages 821–824, 2012.

[61] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *CVPR*, pages 2720–2729, 2019.

[62] Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *ICML*, pages 7124–7133, 2019.

[63] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

[64] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*, pages 8156–8164, 2018.

[65] Junbao Zhuo, Shuhui Wang, Shuhao Cui, and Qingming Huang. Unsupervised open domain recognition by semantic discrepancy minimization. In *CVPR*, pages 750–759, 2019.