Towards Adversarial Robustness in Unlabeled Target Domains

Jiajin Zhang, Hanqing Chao, Pingkun Yan, *Senior Member, IEEE*

***Abstract*—In the past several years, various adversarial training (AT) approaches have been invented to robustify deep learning model against adversarial attacks. However, mainstream AT methods assume the training and testing data are drawn from the same distribution and the training data are annotated. When the two assumptions are violated, existing AT methods fail because either they cannot pass knowledge learnt from a source domain to an unlabeled target domain or they are confused by the adversarial samples in that unlabeled space. In this paper, we first point out this new and challenging problem—*adversarial training in unlabeled target domain*. We then propose a novel framework named Unsupervised Cross-domain Adversarial Training (UCAT) to address this problem. UCAT effectively leverages the knowledge of the labeled source domain to prevent the adversarial samples from misleading the training process, under the guidance of automatically selected high quality pseudo labels of the unannotated target domain data together with the discriminative and robust anchor representations of the source domain data. The experiments on four public benchmarks show that models trained with UCAT can achieve both high accuracy and strong robustness. The effectiveness of the proposed components is demonstrated through a large set of ablation studies. The source code is publicly available at https://github.com/DIAL-RPI/UCAT.**

***Index Terms*—Adversarial robustness, domain adaptation, contrastive learning, pseudo labeling.**

I. Introduction

ADVERSARIAL robustness of deep learning models has been intensively studied in the past few years. Many works show the vulnerability of deep learning models to adversarial samples that contain imperceptible perturbations [1], [2], [3]. Various approaches have been proposed to train deep learning models resilient to adversarial attacks [3], [4]. However, the existing approaches assume that **a)** training and testing data are drawn from the same distribution, and **b)** the training set is fully labeled. When the assumptions are violated, *i.e.*, testing data distribution deviating from the training data and lacking training labels, such approaches would experience a significant performance drop or even become inapplicable. In this paper, we aim to tackle this new and yet very challenging problem, where both the two assumptions are violated. The problem is coined as *adversarial training in unlabeled target domain.*

Manuscript received on Jan. 6th, 2022; revised on Sept. 26th, 2022; accepted on Jan. 23rd, 2023.

J. Zhang, H. Chao and P. Yan are with the Department of Biomedical Engineering and the Center for Biotechnology and Interdisciplinary Studies, Rensselaer Polytechnic Institute, Troy, NY, USA 12180;

J. Zhang and H. Chao are co-first authors. P. Yan is the corresponding author. Emails: {zhangj41, chaoh, yanp2}@rpi.edu

This work was supported in part by the NSF CAREER award 2046708 and the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu).

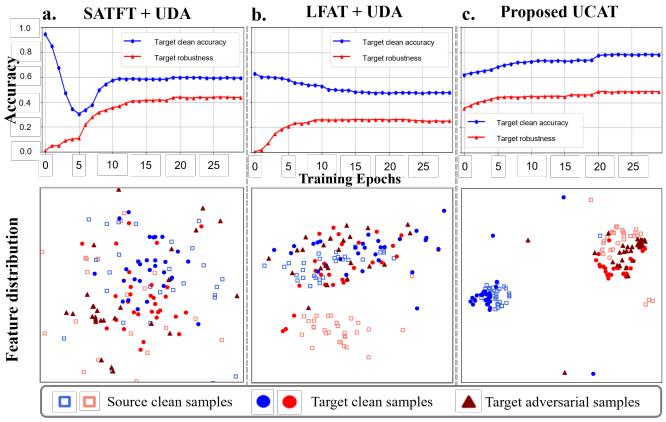

Fig. 1. Performance of *adversarial training in unsupervised target domain* on dataset Office-31 **D**→**A**. First row shows the variation of the performance through training. Second row shows the final learnt representations visualized by t-SNE. Compared to the two baseline strategies, i.e., SAT finetuned UDA (SATFT + UDA) and label-free adversarial training loss with UDA (LFAT + UDA), the proposed UCAT effectively utilizes the knowledge from the source domain and achieved both high accuracy and strong robustness in the unlabeled target domain. The UDA method used here is SRDC [5].

Solving this new problem is highly nontrivial. Recent work of semi-supervised adversarial training (SAT) [6] deals with partially labeled training data, but the training and testing sets still belong to the same distribution/domain. Shafahi et al. [7] adapted a robust model trained in a *source domain* to a *target domain* but requires full supervision. A simple strategy is to combine such existing methods, *i.e.*, to first train a non-robust model in the unlabeled target domain with unsupervised domain adaptation (UDA), then generate pseudo labels using the trained model in the unlabeled target domain, and finally apply SAT with those pseudo labels to further finetune the model. We call this strategy SAT finetuned UDA (SATFT + UDA). Nevertheless, since the model cannot effectively leverage source domain data during finetuning, the model would "forget" the knowledge learned with UDA, which results in poor performance (see Fig. 1a). The other baseline strategy is to introduce label-free adversarial training loss used in SAT into UDA, denoted as LFAT + UDA. Such a loss constrains KL-divergence between logits from a clean sample and its adversarial counterpart, respectively. While the label-free adversarial training loss works effectively in a single domain problem, such as SAT, significant distribution shift and missing labels in the target domain confuse the models as illustrated in Fig. 1b. Since the adversarial samples are designed to fool the model, merely enforcing the consistency between the target clean-adversarial sample pairs in the unlabeled target domain could mislead the UDA training and cause the model to map

clean samples into the wrong classes.

In this paper, we propose a novel adversarial training framework named *Unsupervised Cross-domain Adversarial Training* (UCAT), which can effectively leverage the knowledge of the labeled source domain to prevent the adversarial samples from misleading the training process and achieve both high accuracy and strong robustness in the unlabeled target domain. Under the guidance of high quality pseudo labels, the proposed UCAT aligns the representations of target domain clean and adversarial samples with the source anchors, which are discriminative and robust representations of the source domain samples. This alignment helps UCAT in two ways: explicitly minimizing the distribution deviation between the source and target domains and implicitly regularizing the distance between the target domain clean samples and their adversarial counterparts.

The proposed UCAT consists of three key components, **1)** a Source Anchored Learning (SAL) loss for robustly aligning the target and source domains, **2)** QUality-QUantity Auto-balanced Pseudo Labeling (QUAPL) for providing pseudo labels to SAL loss, and **3)** a discriminative and robust source model for generating anchors.

Our work makes three major contributions:

a) To the best of our knowledge, this is the first work focusing on adversarial training in an unlabeled target domain.[1]. Our analysis demonstrate that robustifying models in an unlabeled target domain is a daunting task, because the inevitable domain distribution deviation worsens the adversarial attacks.

b) We propose a new framework of *Unsupervised Cross-domain Adversarial Training* (UCAT) to tackle the above problem by effectively utilizing knowledge from the labeled source domain to enhance the representation learning in the unlabeled target domain.

c) The experiments on four public benchmarks (DIGITS [9], [10], [11], Office-31 [12] and VisDA-2017 [13]) demonstrate that the proposed UCAT can efficiently train a model to have high accuracy and be resilient to adversarial attacks in an unlabeled target domain.

II. Related Works

A. Adversarial Attack and Defense

Szegedy [14] first reported that deep neural networks can be fooled by adversarial samples, which heralded the era of adversarial attacks and robustness of deep learning models. Soon after that, Goodfellow et al. [2] proposed the fast gradient sign method (FGSM) to efficiently find such adversaries and presented a robust training approach by including adversarial samples into the training data. More effective attack and defence approaches, such as CW [1], PGD [3], BIM [15], MIM [16], DeepFool [17], JSMA [18], FAB [19], and AWP [20], were soon proposed to identify the instabilities of deep neural networks. However, all these methods require fully supervised learning, and the training and testing data to be in the same domain. Kannan et al. [21] proposed the first label-free robust training strategy, adversarial logit pairing (ALP). It applies an additional loss term constraining the pairwise logit feature distance between a clean-adversarial sample pair. Later, TRADES [4] proposed a trade-off-inspired adversarial defense via surrogate-loss minimization, which measures model accuracy on clean samples and model robustness on adversarial samples with two separate loss terms derived theoretically.

Inspired by the seminal work of TRADES [4], which generates adversarial samples without labels, several Semi-supervised Adversarial Training (SAT) methods [6], [22] were proposed for adversarial training with partially labeled samples. However, SAT assumes there is no domain shift between the labeled and unlabeled data. Another group of recent works explored robust transfer learning to deal with the Out-of-Distribution(OoD) data in a target domain [7], [23]. Nevertheless, those works require both the source and target domain data to be fully labeled, in order to adopt the knowledge distillation and fine tuning strategies. Unlike the previous works, we consider a more challenging situation, where both the requirements are unmet, *i.e.*, improving adversarial robustness in an unlabeled OoD target domain.

B. Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) aims to train a model in an unlabeled target domain with significant data distribution shift from the source domain. One representative group of UDA approaches, such as [11], [24], [25], [26], align source and target domains by learning domain invariant representations. Among these works, [27], [28], [29] use individual task classifiers for the two domains to detect non-discriminative features and reversely learn a discriminative feature extractor. Other works [30], [31], [32] focus attention on transferable regions to derive a domain-invariant classification model. To help achieve target-discriminative features, [33], [34] generate synthetic images from raw input data of the two domains via generative adversarial networks (GANs) [35]. A recent work [36] improves adversarial feature adaptation by dealing with the deterioration of the discriminative structures of target data.

The cluster assumption states that the classification boundary should not pass through high-density regions, but instead lie in low density regions [37]. To enforce the cluster assumption, conditional entropy minimization is widely used in the UDA community [38], [39], [40], [41], [42], [43], [44]. Another group of methods including [45], [46], [47], directly minimize the domain discrepancies, which can be quantified by the maximum mean discrepancy (MMD) in practice [48]. Some recent works, such as DIRT-T [42], CAN [49], SRDC [5] and CAT [50], show that alleviating the intrinsic inter-class mismatch via cluster assumption benefits the model adaptation performance. Recently, there is also another line of researches [51], [52] working on to leverage adversarial training process to find hard-to-learn or OoD samples for further enhancing the model generalization performance. However, the existing methods listed above generally neglect the adversarial robustness of models in the target domain.

Our work aims to bridge the research gap by enhancing the adversarial robustness of models in an unlabeled target

[1]This work was firstly posted on Arxiv on Nov. 2020 [8]

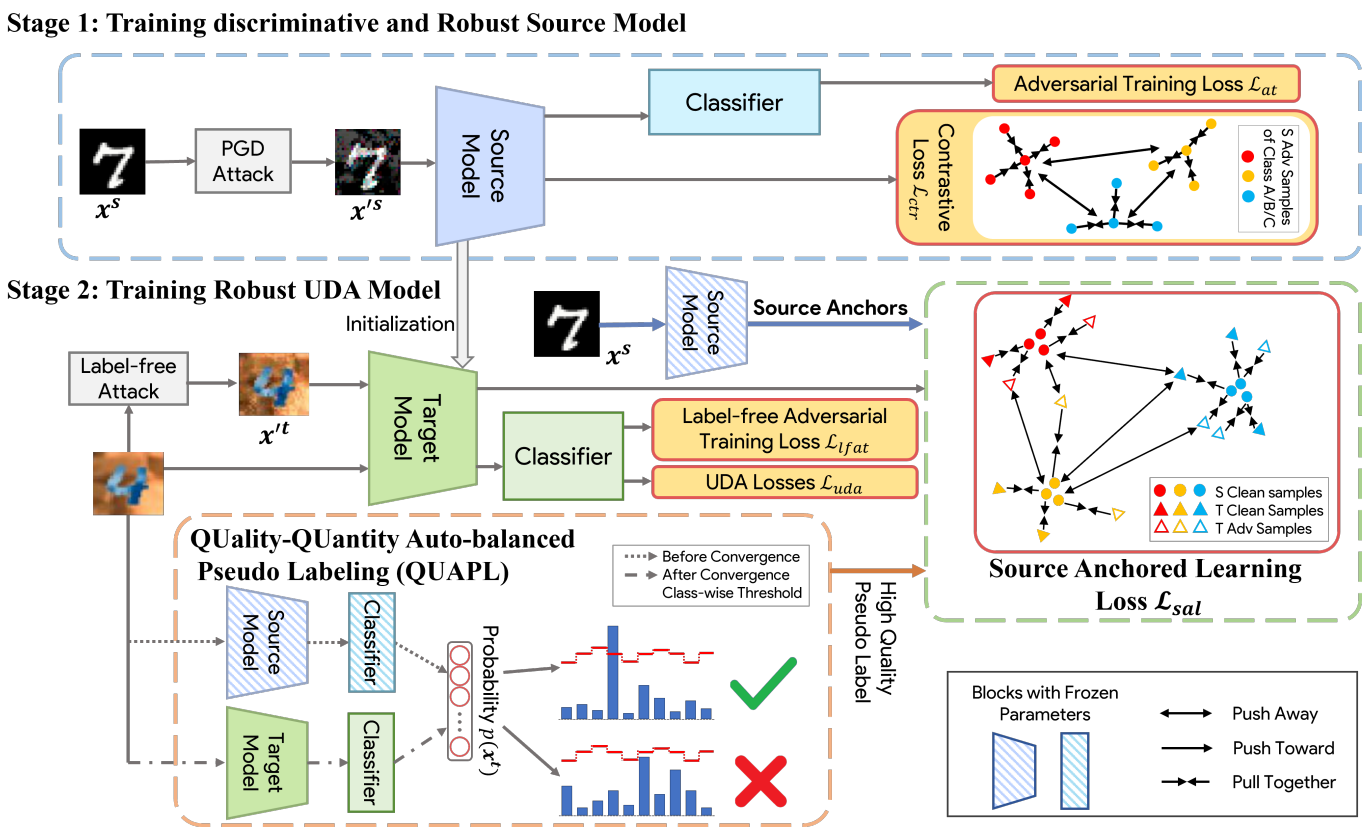

Fig. 2. The overview of the Unsupervised Cross-domain Adversarial Training (UCAT). The proposed UCAT has three key components, *i.e.*, Source Anchored Learning (SAL) loss, QUality-QUantity Auto-balanced Pseudo Labeling (QUAPL), and the discriminative and robust source model.

domain. We were the first in the field to formulate the problem and a preprint of our work was publicly released earlier [8]. Recently, Yang et al. [53] explored the robustness of UDA in segmentation tasks, where a contrastive loss constrains the model prediction on clean-adversarial sample pairs. It is worth noting that, modern segmentation deep neural networks are intrinsically robust to adversarial attacks compared with classification networks [54]. Thus, the problem tackled in our work is more challenging and general. This paper presents a novel robust training framework to effectively deal with the problem.

C. Pseudo Labeling

Pseudo-labeling methods [55], [56] generate pseudo labels to facilitate the use of unlabeled data for training models. The output from pretrained models are directly use for this task in some early works. Inspired by the noise correction works [57], [58] attempted to update the pseudo-labels through an optimization framework. Xie [59] showed self-training can improve the performance of supervised classification tasks. Although the previous pseudo-labeling methods are general and domain-agnostic, they tend to generate noisy labels. Nayeem [60] tried to reduce label noise by improving network calibration, which is defined as the consistency between the model's confidence and accuracy [61]. Naive strategies typically use a manually set threshold and accept all predicted probabilities higher than that threshold as valid pseudo labels. However, such strategies cannot balance the quantity and quality of valid pseudo labels. In our work, the target domains are completely unlabeled and thus parameter-tuning is inapplicable. We present a novel approach to balance the quantity and quality of the pseudo labels to automatically determine optimal thresholds.

III. Method

Fig. 2 shows the overall architecture of the proposed framework of *Unsupervised Cross-domain Adversarial Training* (UCAT). Sitting at the center of our innovation, the Source Anchored Learning (SAL) loss utilizes the discriminative and robust representations from a source domain as fixed anchors to pull the clean and adversarial representations in the target domain towards the anchors or push them away, depending on the pseudo labels of the target domain data. SAL requires solid source anchors and high quality pseudo lables to be effective. In our work, the source anchors are generated by a discriminative and robust source model, which is trained by integrating the fully supervised adversarial training with the conventional contrastive learning. High quality pseudo labels are obtained using QUAPL, which automatically selects suitable labeling threshold for each class by gauging the quantity and quality of pseudo labels.

In addition to the SAL loss, UCAT also incorporates the label-free adversarial training [6] and the typical UDA losses [24], [5] to further improve the performance of the target model.

A. Problem Formalization

Let $\mathcal{S} = \{(\boldsymbol{x}_1^s, y_1^s), ..., (\boldsymbol{x}_{N_s}^s, y_{N_s}^s)\}$ denote a labeled source domain dataset and $\mathcal{T} = \{\boldsymbol{x}_1^t, ..., \boldsymbol{x}_{N_t}^t\}$ be an unlabeled target domain dataset, where $\boldsymbol{x}$ denotes the input images and $y \in \{1, ..., K\}$ indicates the labels. The new problem requires training a model, which not only performs well on the clean samples $\boldsymbol{x}^t$ in the target domain $\mathcal{T}$, but also be robust against the adversarial attacks $\boldsymbol{x'}^t$. Here, $\boldsymbol{x'}^t = \boldsymbol{x}^t + \boldsymbol{\eta}$ denotes an adversarial counterpart of $\boldsymbol{x}^t$, where $||\boldsymbol{\eta}||_p \in (0, \epsilon]$ is an imperceptible adversarial perturbation. We use $p^s(\boldsymbol{x}), p^t(\boldsymbol{x}) \in [0,1]^K$, and $f^s(\boldsymbol{x}), f^t(\boldsymbol{x})$ to denote the predicted probabilities and the learnt representations (the output of the penultimate layer).

B. Discriminative and Robust Source Model

To provide discriminative and robust representations of the source domain data as source anchors for the SAL loss, we first integrate the adversarial training (AT) [3] loss $\mathcal{L}_{at}$ and the contrastive loss $\mathcal{L}_{ctr}$ to train a discriminative and robust source model as shown in Fig. 2).

Specifically, AT feeds the model with adversarial samples $\boldsymbol{x'}_i^s$ generated by using the method of projected gradient descent (PGD) [3] and optimizes the model with cross entropy loss $\ell_{ce}(\boldsymbol{x}, y) = -\log p_y^s(\boldsymbol{x})$:

$$\mathcal{L}_{at} = \mathop{\mathbb{E}}_{(\boldsymbol{x}_i^s, y_i^s)\in\mathcal{S}} \ell_{ce}(\boldsymbol{x'}_i^s, y_i^s), \tag{1}$$

where $\boldsymbol{x'}_i^s = \arg\max_{||\boldsymbol{x'}-\boldsymbol{x}_i^s||_p \le \epsilon} \ell_{ce}(\boldsymbol{x'}, y_i^s)$, and $p_y^s(\boldsymbol{x})$ denotes the y-th item of the K dimensional prediction.

The contrastive loss constrains the learned representations $\boldsymbol{f^s}(\boldsymbol{x'}^s)$ of the adversarial samples via minimizing the intra-class distance and maximizing the inter-class distance. It is formulated as

$$\mathcal{L}_{ctr} = \mathop{\mathbb{E}}_{1\le i,j\le N_s} \mathbb{1}_{[y_i^s = y_j^s]} {\mathcal{D}_{ij}^s}^2 + \mathbb{1}_{[y_i^s \neq y_j^s]} [m - \mathcal{D}_{ij}^s]_+^2, \tag{2}$$

where $\mathcal{D}_{ij}^s = ||f^s({\boldsymbol{x'}_i}^s) - f^s({\boldsymbol{x'}_j}^s)||_2$ is the Euclidean distance between two feature vectors, $m > 0$ is a margin to prevent over-fitting, and $[\cdot]_+$ represents $\max(0, \cdot)$. The complete loss function for the discriminative and robust source model training is given by $\mathcal{L}_{src} = \mathcal{L}_{at} + \lambda_{ctr}\mathcal{L}_{ctr}$, where λ_{ctr} is a positive weighting parameter.

C. Source Anchored Learning

In order to robustify a model in the unlabeled target domain, the Source Anchored Learning (SAL) loss is proposed to simultaneously align the target and source domains and regularize the deviations caused by adversarial perturbations. As introduced in Sec. I, although a regular label-free adversarial training loss can help reduce the distance between clean samples and their adversarial counterparts, such a direct constraint may confuse the the model when the unlabeled data has significant distribution deviation from the labeled data. While it intends to enforce a classifier to assign similar labels to the adversarial samples and the corresponding clean samples, such an adversarial training loss will inevitably pull clean samples towards the adversarial samples due to the absence of labels.

In contrast, as shown in Fig. 2, our proposed SAL loss uses source data representations as fixed anchors. Based on the pseudo labels of target domain data samples, SAL pulls the representations of both clean and adversarial target samples towards the source anchors of the same class and pushes them away from the fixed source anchors of different classes. The pulling minimizes the distance between the target clean-adversarial pairs, while in the same time prevents adversarial samples from dragging clean samples towards wrong classes. The pushing enlarges the margin of the decision boundary, which makes the model more robust against attacks.

Let $\hat{\mathcal{T}} = \{(\boldsymbol{x}_i^t, \hat{y}_i^t) \mid \boldsymbol{x}_i^t \in \mathcal{T}, \boldsymbol{x}_i^t \text{ has valid } \hat{y}_i^t\}$ denote the pseudo-labeled target domain subset which includes all target domain data with valid pseudo label $\hat{y}_i^t$ generated by QUAPL (details of QUAPL are introduced in Sec. III-D).

The proposed SAL loss is formulated as

$$\begin{aligned}\mathcal{L}_{sal} = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{x}_i^t, \hat{y}_i^t)\in\hat{\mathcal{T}},\\ (\boldsymbol{x}_j^s, y_j^s)\in\mathcal{S}}} \Big\{ & \mathbb{1}_{[\hat{y}_i^t = y_j^s]} ({\mathcal{D}_{\boldsymbol{x}_i^t, j}^t}^2 + {\mathcal{D}_{\boldsymbol{x'}_i^t, j}^t}^2) \\ & + \mathbb{1}_{[\hat{y}_i^t \neq y_j^s]} \left([m - \mathcal{D}_{\boldsymbol{x}_i^t, j}^t]_+^2 + [m - \mathcal{D}_{\boldsymbol{x'}_i^t, j}^t]_+^2 \right) \Big\},\end{aligned} \tag{3}$$

where $\mathcal{D}_{\boldsymbol{x},j}^t = ||f^t(\boldsymbol{x}) - f^s(\boldsymbol{x}_j^s)||_2$ and m is the same margin used in $\mathcal{L}_{ctr}$. $\boldsymbol{x}_j^s$ is a clean source sample. $\boldsymbol{x'}_i^t$ is the adversarial counterpart of the clean sample $\boldsymbol{x}_i^t$ generated using the label-free adversarial training [4]: $\boldsymbol{x'}_i^t = \arg\max_{||\boldsymbol{x'}-\boldsymbol{x}_i^t||_p \le \epsilon} \mathrm{KL}(p^t(\boldsymbol{x'})||p^t(\boldsymbol{x}_i^t))$, where $\mathrm{KL}(\cdot)$ represents KL-divergence. In practice, to include sufficient number of source anchors of each class in a mini-batch with size n_t to compute the SAL loss, we randomly sample $\lceil n_t/K \rceil$ source samples of each class as source anchors.

D. Pseudo Labeling (PL)

To align the unlabeled target data with the labeled source data, SAL loss requires using pseudo labels. Naïve strategies typically use a manually set threshold and accept all predicted probabilities higher than that threshold as valid pseudo labels. However, such strategies cannot balance the quantity and quality of valid pseudo labels. A greater threshold would help select higher quality labels but could result in small number of valid samples. In contrast, a lower threshold would yield abundant samples, but with noisy labels.

To deal with the problem, we propose QUality-QUantity Auto-balanced Pseudo Labeling (QUAPL) strategy, which evaluates both the quality and quantity of pseudo labels under various thresholds to automatically determine the best threshold.

Quantity Computation: Let $\mathcal{S}^c = \{(\boldsymbol{x}_i^s, y_i^s) \mid \arg\max_{k\in\{1,...,K\}} p_k(\boldsymbol{x}_i^s) = c\}$ denote a sample set composed of all the source test samples predicted as class $c \in \{1, ..., K\}$, where $p_k(\cdot)$ represents the k-th entry of the label generator's K-dimensional prediction. Let $\gamma^c \in \{\frac{i-1}{L} | i \in \{1, ..., L\}\}$ denotes a discretized PL threshold on class c and $\mathcal{V}_{\gamma^c} = \{(\boldsymbol{x}_i^s, y_i^s) \mid (\boldsymbol{x}_i^s, y_i^s) \in \mathcal{S}^c,\ p_c(\boldsymbol{x}_i^s) > \gamma^c\} \subseteq \mathcal{S}^c$ represents the correspondingly generated valid sample set. The quantity is evaluated as $\frac{|\mathcal{V}_{\gamma^c}|}{|\mathcal{S}^c|}$.

Quality Evaluation: To evaluate the quality of the predicted labels, we split the sample set $\mathcal{S}^c$ into L disjoint subsets based on the range of the predicted probability $p_c(\boldsymbol{x}_i^s)$. The l-th sample subset ($l \in \{1, ..., L\}$) is denoted as $\mathcal{S}_l^c = \{(\boldsymbol{x}_i^s, y_i^s) \mid (\boldsymbol{x}_i^s, y_i^s) \in \mathcal{S}^c,\ p_c(\boldsymbol{x}_i^s) \in (\frac{l-1}{L}, \frac{l}{L}]\}$. For a nonempty subset $\mathcal{S}_l^c$, the label generator's prediction accuracy is formulated as $Acc_l^c = \mathbb{E}_{(\boldsymbol{x}_i^s, y_i^s) \in \mathcal{S}_l^c} \mathbb{1}_{[y_i^s = c]}$. The confidence of the label generator on a nonempty subset $\mathcal{S}_l^c$ is $Con_l^c = \mathbb{E}_{(\boldsymbol{x}_i^s, y_i^s) \in \mathcal{S}_l^c} p_c(\boldsymbol{x}_i^s)$, *i.e.*, the mean prediction value. Following [61], the calibration of the label generator is defined as

$$Cal_l^c = 1 - |Acc_l^c - Con_l^c|, \tag{4}$$

which indicates the consistency between the confidence and accuracy of the label generator [61].

Finally, we define a quality-quantity score $Q(\cdot)$ of PL threshold γ^c as

$$Q(\gamma^c) = \underbrace{\frac{|\mathcal{V}_{\gamma^c}|}{|\mathcal{S}^c|}}_{Quantity} \cdot \overbrace{\mathop{\mathbb{E}}_{\substack{\gamma^c \times L + 1 \le i \le L \\ \mathcal{S}_i^c \neq \varnothing}} (Acc_i^c \cdot Cal_i^c)}^{Quality}. \tag{5}$$

It is worth noting that calibration indicates how likely a generator predicted label may be correct. Good calibration has been assumed by the existing pseudo labeling works. However, as pointed out by the previous work [61], deep learning models are usually mis-calibrated. By considering both accuracy and calibration, we strike a good balance between them when generating the pseudo labels.

To automatically select the best threshold, we calculate $Q(\gamma^c)$ for all γ^c and use the one with the highest score as the optimal threshold τ^c for class c, *i.e.*, $\tau^c = \arg\max_{\gamma^c} Q(\gamma^c)$. Fig. 3 illustrates how QUAPL balances the quantity and quality of pseudo labels through the quality-quantity score $Q(\gamma^c)$. Without loss of generality, we use the class '0' (c=0) in the MNIST dataset [10] as an example. When increasing the value of γ^0, the quality-quantity score $Q(\gamma^0)$ initially increases gradually. In this stage, since the PL threshold is relatively small, most the samples are selected, so the quantity term in $Q(\gamma^0)$ does not change much. As the PL threshold further becomes greater, the quality of the selected pseudo labels quickly increases. Therefore, $Q(\gamma^0)$ is dominated by quality in this stage. Until $\gamma^0 = 0.91$, $Q(\gamma^0)$ reaches its peak value. After that, it goes down rapidly. In this stage, because the quality of the selected pseudo labels is already good, further increasing the threshold cannot effectively increase the quality. However, many samples are filtered out by the large threshold. Thus, the quantity term reduces quickly, which causes a significant drop of $Q(\gamma^0)$. The peak of the curve marks the tipping point that the dominance shifts from quality to quantity. Thus, $\gamma^0 = 0.91$ for this peak is chosen as the optimal threshold τ^0 for class '0'.

In our work, the data in the target and source domains are on the same task, despite the distribution difference between them. Therefore, a model may have a similar quality-quantity relationship in both domains. We then apply the label generator with the set of thresholds $\{\tau^1, ..., \tau^K\}$ selected using the test samples in the source domain to the unlabeled target domain samples. Let $\hat{y}_i^t = \arg\max_{k \in \{1,...,K\}} p_k(\boldsymbol{x}_i^t)$ denote the predicted label of $\boldsymbol{x}_i^t$. A pseudo label considered to be valid, if $p_{\hat{y}_i^t}(\boldsymbol{x}_i^t) > \tau^{\hat{y}_i^t}$. The SAL loss is then computed using the target domain sample set $\hat{\mathcal{T}} = \{(\boldsymbol{x}_i^t, \hat{y}_i^t) \mid \boldsymbol{x}_i^t \in \mathcal{T},\ \hat{y}_i^t = \arg\max_{k \in \{1,...,K\}} p_k(\boldsymbol{x}_i^t),\ p_{\hat{y}_i^t}(\boldsymbol{x}_i^t) > \tau^{\hat{y}_i^t}\}$. In practice, we compute the thresholds twice during the training procedure. We first use the trained source model $F^s(\cdot)$ as the label generator and determine the initial set of thresholds. After v epochs, when the target model F^t becomes stable, it kicks in to serve as the label generator. The second set of thresholds is computed using the new label generator.

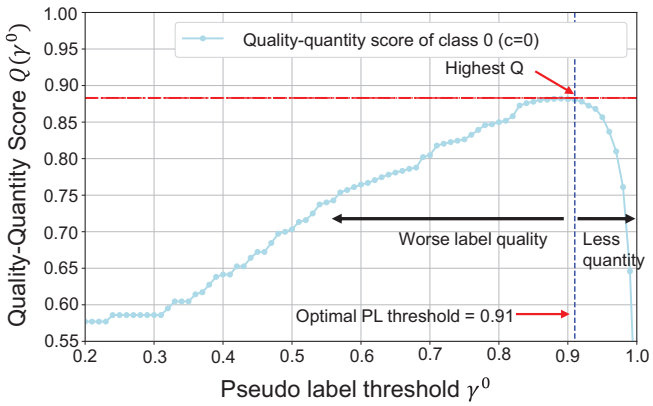

Fig. 3. The illustration of how QUAPL works to balance the quality and quantity of the valid pseudo labels.

E. Overall Loss Function and Training Procedure

Although the proposed SAL loss with QUAPL can utilize most of the unlabeled target samples, it is inevitable that some target samples cannot be used due to the lack of reliable pseudo labels. To make use of the most of all data, we include the label-free adversarial training loss and UDA losses in the the target model training. Following [6], the label-free adversarial training loss can be formulated as $\mathcal{L}_{lfat} = \mathbb{E}_{\boldsymbol{x}_i^t \in \mathcal{T}} \mathrm{KL}(p^t(\boldsymbol{x'}) || p^t(\boldsymbol{x}_i^t))$. For UDA loss $\mathcal{L}_{uda}$, we investigated two representative UDA methods using different mechanisms, *i.e.*, ADDA [24] and SRDC [5]. ADDA applies an adversarial-discriminative loss to force the target model $F^t(\cdot)$ to map the target samples into the same representation space as the source model $F^s(\cdot)$. SRDC is a deep clustering based method, which achieves state-of-the-art performance by regularizing source and target domain feature distributions with additionally introduced auxiliary distributions. The overall loss function for target model training is:

$$\mathcal{L}_{tgt} = \lambda_{sal} \mathcal{L}_{sal} + \mathcal{L}_{lfat} + \mathcal{L}_{uda}, \tag{6}$$

where λ_{sal} is a positive weighting parameter.

Alg. 1 shows the overall training procedure of the proposed UCAT. First, a discriminative and robust source model F^s is trained with $\mathcal{L}_{src}$. Then, the robust target model F^t is trained by optimizing $\mathcal{L}_{tgt}$. In this process, the previously trained F^s has three uses: **a)** its parameters are used to initialize F^t; **b)** it is served as the label generator to provide pseudo labels to the SAL loss until F^t is converged; **c)** it provides representations of source data as source anchors to $\mathcal{L}_{sal}$. When

Algorithm 1 The overall training procedure of UCAT.

Input: source domain data $\{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{N_s}$; target domain data $\{\boldsymbol{x}_i^t\}_{i=1}^{N_t}$; batch size n_t; weighting parameter λ_{sal}.
Output: target model F^t

1: Train a discriminative and robust source model F^s with $\mathcal{L}_{src}$ on the source domain data.
2: Initialize the target model $F^t = F^s$
3: Initialize the label generator $g = F^s$.
4: Select class-wise pseudo-label thresholds via **QUAPL**.
5: **while** $\mathcal{L}_{tgt}$ has not converged **do**
6: Form a source batch: randomly sample $\lceil n_t/K \rceil$ source data of each class as source anchors $\{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_t}$.
7: Form a target batch: randomly sample n_t target data $\{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$.
8: Form the target pseudo-labeled batch $\hat{\mathcal{T}}$ from the target batch.
9: Compute $\mathcal{L}_{uda}$ with $\{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_t}$ & $\{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$.
10: Compute $\mathcal{L}_{lfat}$ with $\{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$.
11: Compute $\mathcal{L}_{sal}$ with $\{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_t}$ & $\hat{\mathcal{T}}$.
12: $\mathcal{L}_{tgt} = \lambda_{sal} \cdot \mathcal{L}_{sal} + \mathcal{L}_{lfat} + \mathcal{L}_{uda}$
13: Update the parameters of F^t by minimizing $\mathcal{L}_{tgt}$.
14: **end while**
15: Update the label generator $g = F^t$.
16: Re-select the pseudo-label thresholds with **QUAPL**.
17: **while** $\mathcal{L}_{tgt}$ has not converged **do**
18: Repeat Step 6 - Step 13.
19: **end while**
20: **Return** F^t

the target model converges with the pseudo label provided by the source model, F^t will replace F^s as the label generator to provide better pseudo labels. With this new label generator, the target model will be iteratively trained until it reaches a new convergence.

IV. Experiments

A. Datasets

*1) **Handwritten digits (DIGITS):*** DIGITS includes three data domains, i.e., **M**NIST (**M**) [10], **U**SPS (**U**) [9] and **M**NIST-**m** (**M-m**) [11]. The number of samples are imbalanced across the three domains, with 100,000 binary images in MNIST, 9,298 binary images in USPS and 68,002 RGB images in MNIST-M. The binary images in the MINIST and USPS dataset are converted into RGB images. We follow the original dataset split for the train and test set in the source and target domain.

*2) **Office-31** [12]:* It is composed by 31 classes of images from three imbalanced domains, *i.e.*, **A**mazon (**A**), **W**ebcam (**W**) and **D**SLR (**D**). The datasets are imbalanced across domains, with $2,817$ images in domain **A**, 795 images in domain **W**, and 498 images in domain **D**. There are a total of 31 classes shared across domains. Both the target and source domain data are split to training and test sets following the ratio of 4:1.

*3) **VisDA-2017** [13]:* This is a large-scale dataset focusing on the **S**ynthetic-**R**eality(**S**→**R**) adaptation. In the source domain, $120,000$ synthetic images are used for training, and $32,397$ images are reserved for testing. In the target domain, $40,000$ images are randomly selected for training, and the remaining $10,000$ images are kept for testing.

*4) **Office-Home** [62]:* This challenging benchmark dataset consists of 65 classes shared across four extremely distinct domains: $2,427$ **Ar**tistic images (**Ar**), $4,365$ **Cl**ip Art images (**Cl**), $4,439$ **Pr**oduct images (**Pr**), and $4,357$ **R**eal-**w**orld images (**Rw**). All 4 domains are split into training and test sets in a ratio of 4:1.

B. Baseline Methods

To the best of our knowledge, this is the first work focusing on adversarial robustness in an unlabeled target domain. We compare the proposed method with two types of baselines introduced in Sec. I, *i.e.*, SAT finetuned UDA (SATFT+UDA) and direct integration of label-free adversarial training loss with UDA (LFAT+UDA). For each of the two types, multiple baseline models are created by using different UDA methods (ADDA and SRDC) and PL strategies. In addition, we also include the results of using fully supervised adversarial training (w/ PGD) in the target domain as a reference.

C. Implementation Details

*1) **Training Details**:* A modified LeNet [10], ResNet-50 [63], ResNet-101, and ResNet-50 are used as the backbone models in DIGITS, Office-31, VisDA-2017, and Office-Home, respectively. For each benchmark, only the last FC layer is modified to fit the task. Mini-batch Stochastic Gradient Descent (SGD) is used as the optimizer in all the training processes with a momentum of 0.9 and a weight decay of $2e-4$. The total training epochs for DIGITS, Office-31, VisDA-2017, and Office-Home are 80, 250, 300, and 200, respectively. We follow the same learning rate schedule as described in [5].

Switching the pseudo label generator at a particular epoch v is determined by the convergence of the training loss $\mathcal{L}_{tgt}$. In our experiments, v is set to 20, 100, 200, and 100 for DIGITS, Office-31, VisDA-2017, and Office-Home, respectively. The adaptive PL thresholds are only calculated twice during training: first at the beginning when setting the fixed robust source model $F^s(\cdot)$ as the label generator, then at the time of switching the label generator to using the target model $F^t(\cdot)$. Compared with the overall training time, the computation time for PL threshold selection is negligible. For each experiment, we trained the model from scratch three times with different random seeds and reported averaged performance in the form of $mean \pm std$. All our networks were implemented using PyTorch v1.3.0.

*2) **Hyperparameter Selection**:* To balance the contributions of the losses, both λ_{ctr} and λ_{sal} are fixed to be 0.01 in all the experiments. In QUAPL, to balance the fineness of thresholds τ^c and the computation complexity, L is fixed to be 100 in all the experiments. For margin size m, we performed a grid search in the source domain and then adopted the selected value in the target domain. It is set to be 10 for DIGITS and 25 for the other benchmarks. A further sensitivity analysis of m is presented in Sec. IV-G.

TABLE I
CLEAN DATA ACCURACY (ACC. (%)) AND ADVERSARIAL ROBUSTNESS (ROB. (%)) ON DIGITS. THE BEST RESULT IN EACH COLUMN IS MARKED IN **BOLD** (EXCEPT FOR THE LAST ROW).

AT methods	M→U		U→M		M→M-m	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
SATFT(0.8) +ADDA	70.3±0.2	48.8±0.2	68.7±0.2	46.7±0.3	58.4±0.3	42.8±0.1
SATFT(0.8) +SRDC	77.5±0.3	56.4±0.2	74.8±0.2	54.5±0.2	66.2±0.2	48.5±0.3
SATFT(QUAPL) +ADDA	72.4±0.2	51.0±0.1	71.2±0.2	49.5±0.1	60.7±0.2	44.2±0.2
SATFT(QUAPL) +SRDC	79.3±0.1	58.3±0.1	78.3±0.2	56.2±0.1	67.7±0.2	53.8±0.1
LFAT +ADDA	82.4±0.3	53.4±0.2	75.4±0.3	54.2±0.1	56.7±0.3	36.4±0.2
LFAT +SRDC	88.1±0.3	57.2±0.1	82.2±0.2	60.8±0.1	68.5±0.2	48.2±0.2
$\text{UCAT}_{\text{ADDA}}$	90.4±0.1	78.5±0.2	89.5±0.1	79.3±0.2	72.7±0.2	64.0±0.2
$\text{UCAT}_{\text{SRDC}}$	**96.6**±0.1	**86.9**±0.1	**97.2**±0.2	**90.3**±0.2	**86.6**±0.1	**74.5**±0.2
Fully Supervised AT	96.2±0.2	81.1±0.2	99.5±0.1	90.0±0.2	96.1±0.2	79.6±0.2

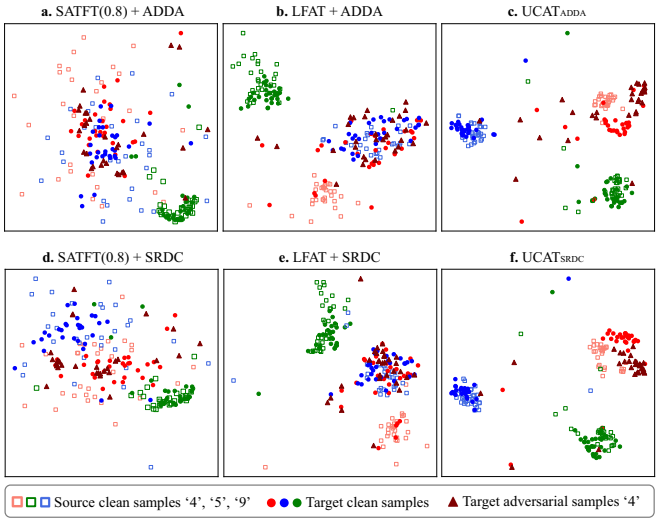

Fig. 4. Visualization by t-SNE on representations learnt by different adversarial training strategies on DIGITS **M→M-m**. Hollow squares (□) and Solid dots (●) represent source and target clean samples respectively. Colors red, blue, and green depicts clean samples of class '4', '5', and '9' respectively. Auburn triangles (▲) represents adversarial samples of class '4'.

*3) **Adversarial Attacks**:* In our experiments, all the images are normalized into the range of $[0, 1]$ and l_∞-norm is the metric to bound the adversarial perturbation. In the training phase, PGD and label-free attack are used for generating adversarial samples in source and target domains, respectively. For MNIST and USPS, the perturbation radii is 0.3, and the number of perturbation steps is 10 with step size $= 0.07$. For other colorful image datasets, the attack perturbation is 0.031, and the number of attack steps$= 10$ with step size$= 0.007$. In Secs. IV-D and IV-E, the target model *robustness* is evaluated by PGD-20 with perturbation radii and step size the same as the training phase. We also evaluate target model *robustness* under a wide variety of attacks in Sec.IV-H.

D. Main Results

*1) **DIGITS**:* Tab. I presents the comparison of UCAT with baselines on the DIGITS benchmark. For SATFT+UDA, "QUAPL" indicates the use of the proposed QUAPL and the fraction number is the manually selected threshold for the naïve PL.

UCAT outperforms all baselines on both clean data accuracy and adversarial robustness. Comparing the results of SATFT+UDA and LFAT+UDA, it can be seen that incorporating UDA in the adversarial training process (LFAT+UDA) benefits the clean data accuracy in most of the case (**M→U**, **U→M**), especially when using advanced UDA method like SRDC. Such an observation verifies that SATFT+UDA would "forget" the source domain knowledge in the finetuning process.

However, on a more challenging task like **M→M-m**, LFAT+UDA hardly outperforms SATFT+UDA and sometimes even performs worse, especially on adversarial robustness. It is because when the target domain has a larger distribution deviation from the source domain, label-free adversarial training loss is more likely to mislead the UDA training. To further verify this, we visualize the learnt representations on **M→M-m** with t-SNE [64] (see Fig. 4). As shown in Fig. 4b&e, although the target adversarial samples of class '4' (▲) are close to their corresponding clean samples (●), all of them are mapped to '9'(□&●).

We also notice that on the task of **M→M-m**, comparing with LFAT+SRDC, $\text{UCAT}_{\text{ADDA}}$ achieves much higher adversarial robustness, in spite of the accuracy on clean data is more close. It demonstrates the significance of learning discriminative representations. Comparing the visualizations in Fig. 4c&e, we can see that representations learnt by $\text{UCAT}_{\text{ADDA}}$ are more discriminative between classes and more compact within each class, which makes the model more resilient to adversarial perturbations. Similar results are observed in Fig. 4f ($\text{UCAT}_{\text{SRDC}}$).

Furthermore, Fig. 4a&d illustrates the weakness of SATFT+UDA on losing the knowledge from a source domain learnt by UDA pretraining. After finetuning, not only the source clean sample '4' and '9' are mixed together, but also the target domain data of different classes are close to each other. This explained the poor performance of SATFT+UDA.

*2) **Office-31**:* The results on Office-31 are presented in Tab. II. Compared with the DIGITS benchmark, Office-31 includes more challenging tasks like **D→A** and **W→A**. Here the weakness of LFAT+UDA becomes more evident. On **D→A**, the performance of LFAT+UDA baselines drops up to 13.8% on clean data accuracy and around 20% on adversarial robustness, compared with their SATFT(QUAPL)+UDA counterparts with the same UDA strategies. In comparison, UCAT consistently obtained high performance on both accuracy and robustness on all the three tasks. On all the tasks except **D→A** and **W→A**, $\text{UCAT}_{\text{SRDC}}$ even significantly outperforms 'Fully

TABLE II
CLEAN DATA ACCURACY (ACC. (%)) AND ADVERSARIAL ROBUSTNESS (ROB. (%)) ON THE OFFICE-31. THE BEST RESULT IS MARKED IN **BOLD** IN EACH COLUMN (LAST ROW EXCLUDED).

AT methods	A→W		W→D		D→A		A→D		D→W		W→A	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
SATFT(0.8) +ADDA	65.2±0.2	51.7±0.3	64.2±0.2	43.8±0.3	47.2±0.3	40.3±0.3	64.2±0.2	43.8±0.1	60.4±0.2	52.3±0.2	51.6±0.2	44.7±0.1
SATFT(0.8) +SRDC	66.1±0.2	53.7±0.2	87.5±0.2	58.6±0.2	59.2±0.2	44.8±0.2	66.8±0.2	45.2±0.1	79.5±0.2	55.7±0.2	60.3±0.1	46.2±0.2
SATFT(QUAPL) +ADDA	66.8±0.2	53.0±0.2	66.0±0.2	44.6±0.1	48.5±0.2	41.1±0.3	66.0±0.2	44.6±0.1	64.8±0.2	55.6±0.1	53.4±0.2	46.2±0.1
SATFT(QUAPL) +SRDC	68.8±0.2	54.6±0.1	90.2±0.2	61.7±0.1	63.4±0.2	46.5±0.2	68.9±0.1	47.2±0.1	83.2±0.2	60.2±0.1	62.2±0.2	47.1±0.2
SATFT(QUAPL) +ADDA +src	64.2±0.1	50.7±0.1	63.5±0.1	41.2±0.2	44.2±0.2	38.0±0.2	63.4±0.1	40.3±0.1	61.3±0.1	52.8±0.1	50.2±0.2	43.1±0.1
SATFT(QUAPL) +SRDC +src	65.1±0.1	51.5±0.1	89.1±0.1	58.2±0.1	60.7±0.2	42.7±0.2	63.2±0.2	43.0±0.1	80.4±0.2	57.6±0.1	57.5±0.2	43.5±0.2
LFAT +ADDA	66.5±0.2	37.7±0.1	84.3±0.2	53.5±0.1	49.5±0.2	22.1±0.1	64.3±0.2	33.5±0.1	88.6±0.1	42.3±0.1	31.6±0.2	20.7±0.1
LFAT +SRDC	66.8±0.2	47.7±0.2	96.2±0.1	63.8±0.1	49.6±0.1	26.3±0.2	67.2±0.1	44.1±0.2	93.6±0.1	62.3±0.1	48.6±0.2	28.5±0.2
$\text{UCAT}_{\text{ADDA}}$	72.1±0.1	59.6±0.1	82.8±0.2	66.2±0.1	60.8±0.2	47.2±0.1	81.7±0.1	56.5±0.1	91.0±0.1	69.9±0.2	73.7±0.2	44.6 ±0.1
$\text{UCAT}_{\text{SRDC}}$	**89.2**±0.1	**68.4**±0.1	**98.7**±0.1	**70.6**±0.2	**78.0**±0.2	**48.6**±0.2	**95.2**±0.2	**67.9**±0.1	**97.1**±0.1	**70.5**±0.1	**78.0**±0.1	**49.7**±0.2
Fully supervised AT	80.5±0.1	59.7±0.1	92.1±0.1	64.0±0.2	77.3±0.2	56.9±0.1	92.1±0.1	64.0±0.1	80.5±0.1	59.7±0.2	77.3±0.2	56.9±0.1

TABLE III
CLEAN DATA ACCURACY (ACC. (%)) AND ADVERSARIAL ROBUSTNESS (ROB. (%)) ON VISDA-2017. THE BEST RESULTS IN EACH COLUMN IS MARKED IN **BOLD** (LAST ROW EXCLUDED).

AT methods	S→R	
	Acc.	Rob.
SATFT(0.8) +ADDA	55.1±0.2	31.9±0.1
SATFT(0.8) +SRDC	64.0±0.2	44.5±0.2
SATFT(QUAPL) +ADDA	57.7±0.2	33.8±0.1
SATFT(QUAPL) +SRDC	66.3±0.1	46.2±0.1
LFAT+ADDA	52.7±0.2	27.6±0.1
LFAT+SRDC	60.4±0.2	38.6±0.2
$\text{UCAT}_{\text{ADDA}}$	65.5±0.2	52.7±0.1
$\text{UCAT}_{\text{SRDC}}$	**80.7**±0.1	**61.3**±0.2
Fully Supervised AT	84.3±0.1	63.7±0.1

Supervised AT'. This observation may attribute to the fact that UCAT can fully utilize both the source and target domain images, while fully supervised adversarial training only has access to the target domain data. It further demonstrates the importance of fully utilizing the labeled source data.

*3) **VisDA-2017**:* The comparison of UCAT and the other baseline models on VisDA-2017 benchmark is presented in Tab. III. UCAT significantly outperforms all the baselines on adversarial robustness with steady high accuracy on clean data. Consistent with the observation on DIGITS and Office-31, since the target domain severely deviates from the source domain, LFAT+UDA suffered significant performance degradation. Similarly, SATFT+UDA is limited by the lack of guidance from the source domain.

*4) **Office-Home**:* The results on Office-Home are presented in Tab. IV. Consistent with the observations on other datasets, UCAT significantly outperforms all the baselines on adversarial robustness with steady high accuracy on clean data. In addition, we also observed that the proposed UCAT outperform fully supervised adversarial training (the last row), especially on the tasks with a small target domain, such as **Cl→Ar**, **Pr→Ar**, and **Rw→Ar**.

E. Ablation Studies

In this section, effectiveness of all the components of UCAT is evaluated on DIGITS **M→U**, and Office-31 **A→W**, **D→A**. Tab. V presents the results of leave-one-component-out ealuation of both $\text{UCAT}_{\text{ADDA}}$ and $\text{UCAT}_{\text{SRDC}}$.

*1) **Source Domain Contrastive Loss** $\mathcal{L}_{crt}$:* For both $\text{UCAT}_{\text{ADDA}}$ and $\text{UCAT}_{\text{SRDC}}$, removing $\mathcal{L}_{ctr}$ in source model training caused a drop of performance on both target clean data accuracy and target adversarial robustness. This result confirms our analysis of the experiments that mapping data into a more discriminative and robust space can benefit *both* clean data accuracy and adversarial robustness.

*2) **SAL Loss** $\mathcal{L}_{sal}$:* Removing $\mathcal{L}_{sal}$ significantly harms both the clean data accuracy and the adversarial robustness on all the three tasks. The observation indicates that the SAL loss plays a critical role to simultaneously align the target and source domains and regularize the deviations caused by adversarial perturbations, which leads to an increase of robustness and high clean data accuracy.

*3) **Label-free Adversarial Training Loss** $\mathcal{L}_{lfat}$:* Comparing the rows 'w/o. $\mathcal{L}_{lfat}$' and 'Full model' in Tab. V, one can see that incorporating label-free adversarial training loss brings benefits to UCAT, instead of causing performance drop like in LFAT+UDA. This phenomenon shows that, by using source anchors, SAL loss can prevent $\mathcal{L}_{lfat}$ from misleading the model training. By correctly aligning the representations of clean-adversarial sample pairs, $\mathcal{L}_{lfat}$ helps leverage the target domain data without valid pseudo label from QUAPL to improve the performance.

*4) **UDA Losses** $\mathcal{L}_{uda}$:* From the rows 'w/o. $\mathcal{L}_{uda}$' and 'Full model' in Tab. V, one can see that, for $\text{UCAT}_{\text{ADDA}}$, the $\mathcal{L}_{uda}$ only brought limited benefits. However, the $\mathcal{L}_{uda}$ significantly increase the performance in $\text{UCAT}_{\text{SRDC}}$. On one hand, the results indicate that the original UDA loss $\mathcal{L}_{uda}$ can help our UCAT better utilize all data from both source and target domains. In addition, better UDA loss, *i.e. SRDC loss in our case*, could have more positive contributions to the overall UCAT clean sample accuracy. On the other hand, the model robustness is less sensitive to the $\mathcal{L}_{uda}$. It is mainly because the robustnesss is brought from the target domain contrastive learning with QUAPL-assisted SAL loss.

*5) **QUAPL**:* The rows with 'Th=0.8' and 'Th=0.9' in Tab. V indicate the baseline models using naïve PL with threshold = 0.8 and 0.9 in UCAT. Compared with these two methods, UCAT using QUAPL improves on both clean data accuracy and adversarial robustness.

We further investigated the effects of switching label generators in QUAPL. The result is presented in the row 'w/o switch' in Tab. V. The performance drop suggests that a well-trained UDA model can generate better pseudo labels than a source domain model, which in turn improves the performance of UCAT.

TABLE IV
CLEAN DATA ACCURACY (ACC. (%)) AND ADVERSARIAL ROBUSTNESS (ROB. (%)) ON THE OFFICE-HOME. THE BEST RESULT IS MARKED IN **BOLD** IN EACH COLUMN (LAST ROW EXCLUDED).

AT methods	Ar→Cl		Ar→Pr		Ar→Rw		Cl→Ar		Cl→Pr		Cl→Rw	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
SATFT(0.8) +ADDA	36.4±0.1	18.2±0.1	40.6±0.2	18.3±0.1	48.9±0.2	24.5±0.1	41.0±0.1	18.7±0.1	43.7±0.1	20.8±0.1	50.2±0.1	24.5±0.2
SATFT(0.8) +SRDC	38.2±0.1	20.7±0.2	47.7±0.1	23.0±0.1	57.6±0.1	28.3±0.2	46.2±0.1	21.1±0.1	49.0±0.1	22.1±0.1	58.6±0.1	29.6±0.3
SATFT(QUAPL) +ADDA	38.6±0.1	21.0±0.2	45.8±0.2	22.6±0.1	53.2±0.1	26.7±0.1	44.5±0.1	20.4±0.1	45.9±0.2	21.6±0.1	54.9±0.1	28.3±0.2
SATFT(QUAPL) +SRDC	40.9±0.2	21.9±0.1	54.9±0.1	23.5±0.2	61.8±0.2	30.4±0.1	51.3±0.2	21.6±0.3	53.5±0.1	22.4±0.2	64.8±0.1	31.6±0.1
LFAT +ADDA	33.5±0.1	17.6±0.2	38.7±0.1	17.5±0.2	52.5±0.3	24.3±0.1	38.6±0.1	18.4±0.2	42.4±0.1	17.0±0.2	54.5±0.1	28.8±0.1
LFAT +SRDC	36.9±0.2	21.2±0.1	44.2±0.1	22.6±0.1	58.7±0.1	30.6±0.2	52.9±0.2	20.5±0.1	47.2±0.1	22.9±0.1	63.5±0.1	33.2±0.2
$\text{UCAT}_{\text{ADDA}}$	39.6±0.1	20.8±0.1	50.5±0.1	21.1±0.1	55.1±0.2	28.6±0.1	58.0±0.2	23.8±0.1	53.5±0.1	22.1±0.2	60.9±0.2	33.5±0.1
$\text{UCAT}_{\text{SRDC}}$	**47.5±0.1**	**24.3±0.1**	**64.6±0.2**	**27.6±0.1**	**67.8±0.1**	**32.9±0.1**	**70.7±0.2**	**27.9±0.1**	**66.1±0.2**	**28.6±0.1**	**72.4±0.1**	**36.8±0.2**
Fully supervised AT	50.3±0.1	28.6±0.1	71.7±0.1	31.2±0.1	79.6±0.1	36.7±0.1	69.5±0.1	27.4±0.1	72.3±0.1	27.9±0.1	71.0±0.1	34.6±0.1
AT methods	**Pr→Ar**		**Pr→Cl**		**Pr→Rw**		**Rw→Ar**		**Rw→Cl**		**Rw→Pr**	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
SATFT(0.8) +ADDA	38.6±0.2	19.7±0.2	36.6±0.2	13.9±0.2	41.3±0.3	17.8±0.2	36.9±0.2	14.5±0.2	39.7±0.3	17.8±0.2	39.5±0.2	18.5±0.2
SATFT(0.8) +SRDC	43.4±0.1	22.5±0.2	39.0±0.2	14.3±0.1	48.4±0.1	18.5±0.1	44.3±0.2	16.8±0.2	42.7±0.2	18.2±0.2	45.6±0.2	23.9±0.2
SATFT(QUAPL) +ADDA	40.3±0.2	21.0±0.1	40.4±0.1	14.8±0.1	53.0±0.3	20.1±0.1	50.5±0.2	18.9±0.2	44.2±0.1	16.9±0.3	51.1±0.3	21.3±0.1
SATFT(QUAPL) +SRDC	48.9±0.1	23.2±0.1	43.2±0.1	16.2±0.1	60.5±0.1	21.6±0.1	56.7±0.1	21.3±0.1	46.9±0.1	19.4±0.1	58.2±0.1	25.6±0.1
LFAT +ADDA	38.2±0.2	18.2±0.3	32.7±0.1	14.3±0.1	42.1±0.1	20.9±0.1	55.7±0.3	26.2±0.2	36.9±0.2	17.3±0.2	40.6±0.2	20.2±0.2
LFAT +SRDC	42.8±0.1	23.0±0.1	35.6±0.1	18.5±0.1	52.6±0.2	24.3±0.1	64.8±0.2	31.0±0.2	40.3±0.2	21.5±0.2	47.8±0.2	25.8±0.1
$\text{UCAT}_{\text{ADDA}}$	46.1±0.1	21.1±0.1	34.7±0.2	16.3±0.1	65.5±0.2	22.9±0.1	51.2±0.1	29.6±0.1	41.8±0.1	21.6±0.1	61.8±0.2	32.4±0.1
$\text{UCAT}_{\text{SRDC}}$	**64.3±0.1**	**26.6±0.1**	**46.1±0.1**	**21.1±0.1**	**74.4±0.1**	**31.2±0.1**	**74.6±0.1**	**40.7±0.1**	**53.5±0.1**	**23.6±0.1**	**75.3±0.1**	**32.7±0.1**
Fully supervised AT	63.3±0.1	24.5±0.1	50.5±0.1	24.8±0.2	78.6±0.1	31.9±0.1	73.7±0.1	39.7±0.1	54.6±0.1	26.1±0.1	75.5±0.1	31.2±0.2

TABLE V
ABLATION STUDIES ON ALL COMPONENTS IN UCAT EVALUATED ON DIGITS **M→U**, AND OFFICE-31 **A→W**, **D→A**. THE ROW '*w/o. switch*' MEANS QUAPL WITHOUT SWITCHING THE LABEL GENERATOR.

Methods		M→U		A→W		D→A	
		Clean	Rob.	Clean	Rob.	Clean	Rob.
$\text{UCAT}_{\text{ADDA}}$	w/o. $\mathcal{L}_{ctr}$	86.6±0.1	71.8±0.1	67.3±0.2	51.6±0.2	54.8±0.1	36.7±0.1
	w/o. $\mathcal{L}_{sal}$	83.3±0.2	56.1±0.1	67.2±0.1	48.5±0.2	50.1±0.1	32.9±0.1
	w/o. $\mathcal{L}_{lfat}$	86.1±0.1	72.9±0.1	66.2±0.1	54.2±0.1	57.3±0.2	40.8±0.1
	w/o. $\mathcal{L}_{uda}$	85.7±0.1	76.5±0.1	67.4±0.1	55.7±0.2	56.6±0.2	44.5±0.2
	Th=0.8	84.7±0.1	74.1±0.1	67.0±0.1	54.7±0.2	55.2±0.1	42.7±0.1
	Th=0.9	87.8±0.2	77.4±0.2	70.9±0.1	57.8±0.2	58.4±0.1	45.3±0.2
	w/o. switch	78.9±0.3	68.2±0.1	66.1±0.1	44.5±0.1	47.4±0.2	37.3±0.3
	Non-robust src model	87.3±0.1	74.6±0.2	68.2±0.2	56.8±0.1	57.8±0.1	43.8±0.2
	Adversarial src anchors	87.6±0.2	77.6±0.2	71.4±0.2	58.3±0.1	58.3±0.2	46.2±0.1
	Full model	**90.4**±0.2	**78.5**±0.2	**72.1**±0.2	**59.6**±0.1	**60.8**±0.2	**47.2**±0.1
$\text{UCAT}_{\text{SRDC}}$	w/o. $\mathcal{L}_{ctr}$	95.6±0.1	76.1±0.2	86.8±0.2	58.9±0.1	74.7±0.1	42.2±0.1
	w/o. $\mathcal{L}_{sal}$	88.7±0.2	66.5±0.1	67.4±0.1	50.3±0.1	49.6±0.1	36.9±0.1
	w/o. $\mathcal{L}_{lfat}$	96.0±0.1	77.2±0.1	85.9±0.1	63.8±0.1	75.2±0.2	44.0±0.1
	w/o. $\mathcal{L}_{uda}$	84.6±0.1	75.4±0.1	67.2±0.1	55.5±0.2	56.2±0.2	44.4±0.1
	Th=0.8	93.8±0.1	78.0±0.2	85.9±0.2	65.1±0.1	72.2±0.1	43.9±0.1
	Th=0.9	95.3±0.1	80.2±0.1	87.4±0.1	65.3±0.2	74.9±0.2	45.7±0.1
	w/o. switch	88.6±0.1	73.8±0.1	77.9±0.1	56.6±0.2	65.2±0.1	45.8±0.2
	Non-robust src model	96.6±0.1	80.6±0.1	87.9±0.2	61.5±0.2	75.5±0.2	41.2±0.2
	Adversarial src anchors	96.6±0.2	86.1±0.1	88.6±0.1	67.1±0.1	76.1±0.2	47.7±0.2
	Full model	**96.6**±0.2	**86.9**±0.2	**89.2**±0.2	**68.4**±0.1	**78.0**±0.2	**48.6**±0.1

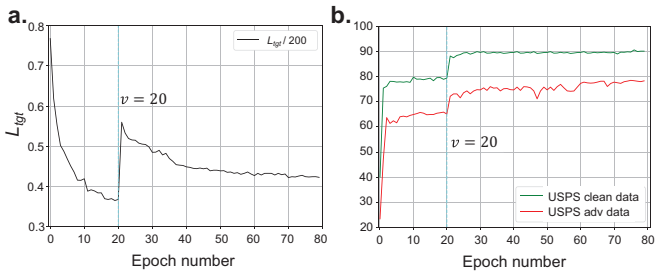

Fig. 5. Effects of switching label generators on **a.** training losses, and **b.** clean accuracy and robustness.

The training process on DIGITS **M→U** of switching the pseudo label generator is presented in Fig. 5. The converting epoch v is determined by the convergence of the training loss $\mathcal{L}_{tgt}$. As shown in Fig. 5a, with the robust source model as the label generator, the $\mathcal{L}_{tgt}$ converged in the first 20 epochs, indicating that the target model has been stable. Then the pseudo label generator is switched to the target model. There is a significant increase of the loss due to more validate sample are included for training. The target domain clean data accuracy and adversarial data robustness presented in Fig. 5b. shows that after the switching, both accuracy and robustness are improved. This results justify the necessity of switching the pseudo label generators, when the $\mathcal{L}_{tgt}$ is converged.

6) ***Robust source model****:* Comparing the rows 'Non-robust src model' with the Full model in Tab. V, we can see that using a robust source model indeed helps improve the model performance in the target domain. The main reason is that a robust model usually extracts/focuses on different features compared to a non-robust model [65]. Thus, the features extracted by a non-robust source model are not suitable for serving as anchors in the training of a robust target model.

7) ***Clean source anchors****:* The rows 'Adversarial src anchors' show that using adversarial images as source anchors leads to inferior performance. To further study how the adver-

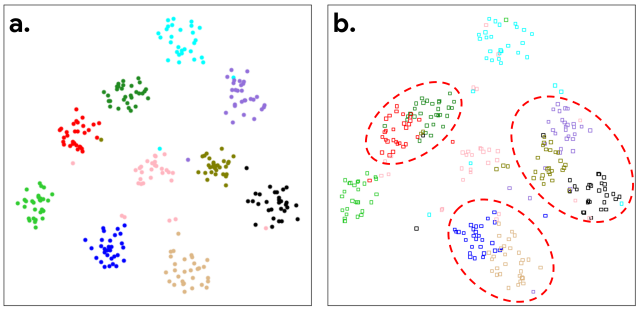

Fig. 6. Visualization by t-SNE on representations learned by a robust source model. The source model is trained with $\mathcal{L}_{src}$. **a.** The representations of the source clean images. **b.** The representations of the source adversarial images.

sarial source anchors jeopardize the target model performance, we visualize the representation space of a discriminative and robust source model with t-SNE as in Fig. 6. Here, different colors indicate different classes. Fig. 6a shows the representations of the source clean images and Fig. 6b presents the representations of source adversarial images. Compared with the compact clusters in Fig. 6a, the distributions of the adversarial source samples within the red dash circles in Fig. 6b are more dispersed and less discriminative. This less discriminative distribution will introduce extra noise and compromise the alignment of the target domain representations to the source domain.

F. Further Analyses on QUAPL

In this section, we further analyze and evaluate QUAPL from three perspectives: **a)** How does QUAPL balance the quality and the quantity of pseudo labels? **b)** Whether it is reasonable to select PL thresholds on the source test set and then directly apply these thresholds to the target domain? **c)** Will increasing the frequency of PL threshold selection significantly boost the performance?

*1) **Balancing Quality and Quantity**:* In Fig. 7, we present the class-wise quality-quantity scores $Q(\gamma^c)$ calculated over varying thresholds γ^c on DIGITS-**M**. As the threshold increases to improve the quality of pseudo labels, the score $Q(\gamma^c)$ increases initially, supported by the label *quality*. However, when the threshold gets greater than a certain value (around 0.9 on DIGITS-**M**), the score $Q(\gamma^c)$ starts to drop rapidly due to the reduction in label quantity.

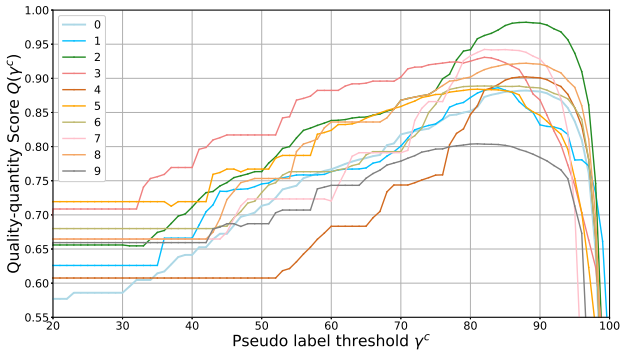

Fig. 7. The curves of the quantity-quality score as a function of PL threshold.

Tab. VI shows the quality (*Rat*, the ratio of the valid samples) and quantity (*Acc*, the accuracy of the valid samples) of pseudo labels in the target domain training set generated by different approaches. Compared with QUAPL, although naïve PL with a threshold = 0.8 selects more samples (*Rat* +7.1%, Th=0.8 v.s. QUAPL), the quality of the selected samples is significantly lower (*Acc* −7.0%, Th=0.8 v.s. QUAPL). When the threshold is increased to 0.9, naïve PL suffers a significant drop on sample quantity (*Rat* −9.0%, Th=0.9 v.s. Th=0.8) with improved on sample quality (*Acc* +6.8%, Th=0.9 v.s. Th=0.8). By adaptively applying the automatically determined thresholds to each class, QUAPL obtains a higher gain on quality (*Acc* +0.2%, QUAPL vs Th=0.9) with a lower loss on quantity (*Rat* +1.9%, QUAPL vs Th=0.9) without requiring manual tuning.

TABLE VI
COMPARISON OF DIFFERENT PLS ON DIGITS **M**→**U**. THE RESULTS ARE EVALUATED ON THE TARGET DOMAIN TRAINING SET. *Rat*: THE RATIO OF THE VALID SAMPLES IN THE TARGET DOMAIN. *Acc*: THE ACCURACY OF THE VALID SAMPLES IN THE TARGET DOMAIN.

Evaluation	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	Avg
$Rat_{th=0.8}$	**83.6**	**81.5**	**78.2**	**73.6**	**84.9**	**72.1**	**85.5**	**83.4**	**92.7**	**78.2**	**81.4**
$Rat_{th=0.9}$	74.7	74.6	63.5	55.6	79.8	65.5	71.6	77.4	84.1	77.0	72.4
Rat_{QUAPL}	79.2	73.4	64.2	66.8	77.3	62.0	78.9	75.8	89.1	76.0	74.3
$Acc_{th=0.8}$	83.6	83.2	83.5	73.3	73.8	73.9	82.3	77.8	74.5	69.4	77.5
$Acc_{th=0.9}$	87.5	**87.2**	**87.8**	**82.6**	**82.4**	78.6	**86.9**	83.4	**83.2**	**82.1**	84.3
Acc_{QUAPL}	**88.6**	**87.2**	87.7	82.0	82.3	**83.4**	85.3	**84.2**	**83.2**	80.6	**84.5**
Th_{QUAPL}	0.91	0.83	0.89	0.85	0.89	0.93	0.82	0.94	0.88	0.82	/

*2) **From Source to Target Domain**:* As mentioned in Sec. III-D, we select PL thresholds on the source domain because a model would have a similar quality-quantity relationship in both source and target domains. Here, we conduct experiments to verify this assumption. We independently compute the QUAPL-selected PL thresholds on the source domain and the target domain (given the ground truth labels) with the source model as the label generator. The consistency between the source and target PL thresholds is then evaluated. Fig. 8a-d present the QUAPL-selected PL thresholds of source (blue lines) and target (red lines) domains on four different UDA tasks, including **M**→**U**, **U**→**M**, **A**→**D**, and **D**→**A**. The x-axis shows the names of all classes in the dataset. From left to right, the classes are sorted according to the PL thresholds selected on the source domain. We can intuitively observe that, as the PL thresholds selected on the target domain follow the same increasing trend as their counterparts selected on the source domain. We noticed that most of the PL thresholds selected on the target domain are slightly higher than their source domain counterparts. It is because the source model performs relatively worse on the target domain, QUAPL automatically selects higher PL thresholds to achieve better pseudo label quality. Pearson correlations between the PL thresholds selected on the source and the target domain. The resulting r values are reported in the top left corner of each figure. The high r values verify the good consistency between the QUAPL-selected PL thresholds in the source domain and target domains, which indicates that it is reasonable to select the PL thresholds with the source domain samples.

*3) **Selection Frequencies**:* In the default settings, the PL thresholds are only selected two times in the whole training process. Here, we evaluate the effects of increasing the selection times on the UCAT performance. We gradually decrease the interval of updating the PL thresholds. The results are

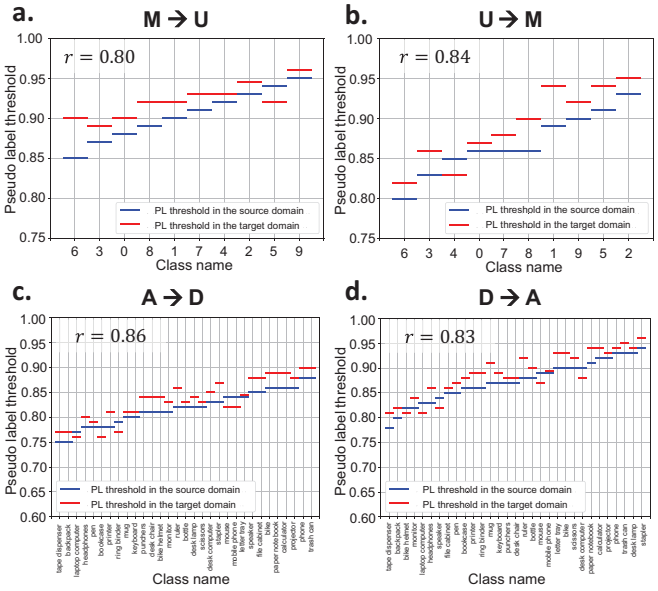

Fig. 8. Evaluating the consistency of QUAPL-based pseudo label thresholds on the source domain and the target domain.

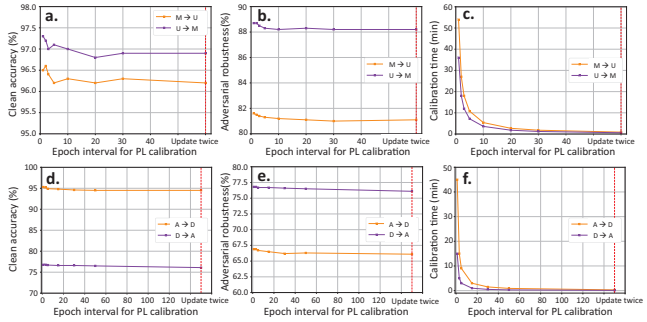

Fig. 9. Evaluating the target model performance with different epoch interval for the pseudo label threshold selection.

shown in Fig. 9. Fig. 9a, b, and c respectively present the variant of the clean data accuracy, the adversarial robustness, and the PL threshold selection time consumption on **M**→**U** and **U**→**M**. Compared with only updating the pseudo label generator twice (the rightmost point on each figure), higher updating frequency brings very limited improvement in accuracy ($< 0.5\%$ increment) and robustness ($< 0.4\%$ increment), but costs significantly more time (> 38 min). It is because the PL thresholds did not change much after the second selection when the target model take place the of the source model to serve as the label generator. Similar results on the task **A**→**D** and **D**→**A** are shown in Fig. 9d, e, and f. Considering the trade-off between model performance and time consumption, we keep the original experimental settings. However, for other different tasks in other future work, we still recommend following this analysis to select the optimal epoch interval.

G. Hyperparameter sensitivity.

The margin size m in $\mathcal{L}_{ctr}$ and $\mathcal{L}_{sal}$ is the only hyperparameter need to be tuned for different tasks. To select appropriate m, a grid search is performed in the source domain ($\mathcal{L}_{ctr}$) and then the selected value is directly adopted to the target domain

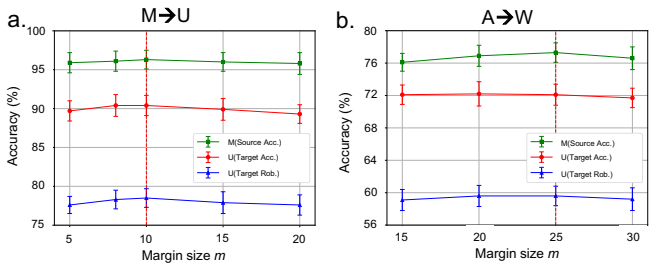

Fig. 10. The sensitivity of $\text{UCAT}_{\text{ADDA}}$ w.r.t. hyperparameter margin size m. The accuracy and robustness in the target domain are evaluated on the tasks of **a.** **M**→**U** and **b.** **A**→**W**.

TABLE VII
ROBUSTNESS(%) EVALUATION OF $\text{UCAT}_{\text{SRDC}}$ UNDER DIFFERENT NON-TARGETED ADVERSARIAL ATTACKS (BIM-40, PGD-40, PGD-100, FAB-100 AND CW-100).

Untargeted Attacks	DIGITS		Office-31		VisDA-2017
	M→U	U→M	A→W	W→A	Syn.→Real
BIM-40	86.3	88.2	66.9	48.0	60.3
PGD-40	86.4	88.3	67.1	47.7	59.8
PGD-100	85.5	87.9	67.2	46.7	58.3
FAB-100	84.9	86.6	65.5	46.0	57.4
C&W-100	82.8	85.6	64.4	45.2	56.1

TABLE VIII
ROBUSTNESS(%) EVALUATION OF $\text{UCAT}_{\text{SRDC}}$ UNDER DIFFERENT TARGETED ADVERSARIAL ATTACKS (BIM-40, PGD-40, PGD-100, FAB-100 AND CW-100).

Targeted Attacks	DIGITS		Office-31		VisDA-2017
	M→U	U→M	A→W	W→A	Syn.→Real
BIM-40	88.2	89.4	69.7	50.5	61.5
PGD-40	87.6	89.0	69.1	50.2	61.5
PGD-100	86.9	88.9	68.7	48.8	60.6
FAB-100	86.7	88.5	68.6	48.5	60.1
C&W-100	85.5	87.2	67.8	48.1	59.3

($\mathcal{L}_{sal}$). Fig. 10 presents the performance of $\text{UCAT}_{\text{ADDA}}$ on the two tasks (DIGITS **M**→**U** and Office-31 **A**→**W**) under varying m. It can be seen that the target model performance (clean data accuracy and adversarial robustness) do not vary a lot around the margin selected based on the source model. The performance on other tasks follows the similar trend.

H. Robustness Under Different Attacks

To evaluate the generalizability of the adversarial robustness of UCAT, we evaluated $\text{UCAT}_{\text{SRDC}}$ under multiple adversarial attacks, including BIM-40, FAB-40, PGD-40, PGD-100 and CW-100 (CW loss optimized by PGD-100). All other settings, such as perturbation radii and step size, are the same as those introduced in Sec. IV-C. The results in Tab. VII show that UCAT can effectively defend against various attacks.

In addition, we evaluate the UCAT robustness under targeted attacks with the same five adversarial attack methods. In our experiments, each class is randomly assigned a targeted label different from the ground truth label. For each attack method, we run the experiments five times and report the average value in Tab. VIII. The results show that compared with non-targeted attacks, UCAT generally performs better under the targeted attacks. Such results are reasonable. Since non-targeted attacks are less constrained, it is easier to generate adversarial samples to manipulate the original model prediction. This observation is also consistent with the previous work [3].

TABLE IX
ROBUSTNESS (%) OF $\text{UCAT}_{\text{SRDC}}$ WITH DIFFERENT SOURCE MODEL TRAINING STRATEGIES (BIM AND C&W). THE RESULTS ARE EVALUATED ON OFFICE-31 $\mathbf{A}\rightarrow\mathbf{W}$.

Training method	BIM-40	PGD-40	PGD-100	FAB-100	CW-100
BIM-10	66.9	67.1	66.6	66.2	65.6
C&W-10	66.7	66.7	66.5	66.7	65.8

Another thing worth of mentioning is that we adopt PGD for source model training because PGD is the one of the most efficient method for improving the model robustness. We also conduct additional experiments using BIM and C&W in the training phase with other settings kept unchanged. Tab. IX shows the results on the Office-31 $\mathbf{A}\rightarrow\mathbf{W}$. We can see that, as long as the attack used in the training is strong, the model can achieve good robustness.

V. CONCLUSION

In this paper, we present a new problem of *adversarial training in unlabeled target domain*, which concerns training an adversarial robust model in a target domain without data annotation. Correspondingly, we propose a novel framework of *Unsupervised Cross-domain Adversarial Training* (UCAT) to tackle this problem by effectively utilizing knowledge from the labeled source domain to enhance the representation learning in the unlabeled target domain. Experiments on four public benchmarks demonstrate that the proposed UCAT can efficiently train a high performance model resilient to various adversarial attacks in an unlabeled target domain. The effectiveness of each components in the proposed framework has been thoroughly validated through ablation studies. Domain adaptation and deep learning robustness have been highly valued in various applications, including autopilot, finance, and medical image analysis. With both high accuracy and strong robustness, our work has great potential in these applications.

VI. ACKNOWLEDGEMENT

This research was partially supported by the National Science Foundation (NSF) under the CAREER award OAC 2046708 and the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network (http://ibm.biz/AIHorizons).

REFERENCES

[1] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE SSP*, 2017, pp. 39–57.
[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.
[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.
[4] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *ICML*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, 2019, pp. 7472–7482.
[5] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8725–8735.
[6] J.-B. Alayrac, J. Uesato, P.-S. Huang, A. Fawzi, R. Stanforth, and P. Kohli, "Are labels required for improving adversarial robustness?" in *NeurIPS*, 2019, pp. 12 214–12 223.
[7] A. Shafahi, P. Saadatpanah, C. Zhu, A. Ghiasi, C. Studer, D. Jacobs, and T. Goldstein, "Adversarially robust transfer learning," in *International Conference on Learning Representations*, 2019.
[8] J. Zhang, H. Chao, and P. Yan, "Robustified domain adaptation," *arXiv preprint arXiv:2011.09563*, 2020.
[9] J. J. Hull, "A database for handwritten text recognition research," *IEEE TPAMI*, vol. 16, no. 5, pp. 550–554, 1994.
[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *JCMR*, vol. 17, no. 1, pp. 2096–2030, 2016.
[12] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.
[13] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "Visda: The visual domain adaptation challenge," *arXiv preprint arXiv:1710.06924*, 2017.
[14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.
[15] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR*, 2017.
[16] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *CVPR*, 2018, pp. 9185–9193.
[17] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *CVPR*, 2015, pp. 427–436.
[18] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *EuroS&P*, 2016, pp. 372–387.
[19] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *ICML*, 2020.
[20] D. Wu, S.-T. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," *Advances in Neural Information Processing Systems*, vol. 33, 2020.
[21] H. Kannan, A. Kurakin, and I. J. Goodfellow, "Adversarial logit pairing," vol. abs/1803.06373, 2018.
[22] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," in *NeurIPS*, 2019, pp. 11 192–11 203.
[23] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, "Do adversarially robust imagenet models transfer better?" *arXiv preprint arXiv:2007.08489*, 2020.
[24] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of CVPR*, 2017, pp. 7167–7176.
[25] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8004–8013.
[26] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3801–3809.
[27] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 285–10 295.
[28] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Adversarial dropout regularization," *arXiv preprint arXiv:1711.01575*, 2017.
[29] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3723–3732.
[30] V. K. Kurmi, S. Kumar, and V. P. Namboodiri, "Attending to discriminative certainty for domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 491–500.
[31] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5345–5352.
[32] J. Wen, R. Liu, N. Zheng, Q. Zheng, Z. Gong, and J. Yuan, "Exploiting local feature patterns for unsupervised domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5401–5408.
[33] G. Kang, L. Zheng, Y. Yan, and Y. Yang, "Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 401–416.

[34] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8503–8512.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[36] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *International conference on machine learning*. PMLR, 2019, pp. 1081–1090.

[37] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *International workshop on artificial intelligence and statistics*. PMLR, 2005, pp. 57–64.

[38] S. Cicek and S. Soatto, "Unsupervised domain adaptation via regularized conditional alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1416–1425.

[39] A. Rastrow, F. Jelinek, A. Sethy, and B. Ramabhadran, "Unsupervised model adaptation using information-theoretic criterion," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 190–197.

[40] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulo, N. Sebe, and E. Ricci, "Unsupervised domain adaptation using feature-whitening and consensus loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9471–9480.

[41] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," *arXiv preprint arXiv:1206.6438*, 2012.

[42] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," *arXiv preprint arXiv:1802.08735*, 2018.

[43] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1426–1435.

[44] Y. Zhang, B. Deng, H. Tang, L. Zhang, and K. Jia, "Unsupervised multi-class domain adaptation: Theory, algorithms, and practice," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[45] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014.

[46] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015, pp. 97–105.

[47] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *ECCV*, 2016, pp. 443–450.

[48] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[49] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *CVPR*, 2019, pp. 4893–4902.

[50] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9944–9953.

[51] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 556–12 565.

[52] E. Hosseini-Asl, Y. Zhou, C. Xiong, and R. Socher, "Augmented cyclic adversarial learning for low resource domain adaptation," *arXiv preprint arXiv:1807.00374*, 2018.

[53] J. Yang, C. Li, W. An, H. Ma, Y. Guo, Y. Rong, P. Zhao, and J. Huang, "Exploring robustness of unsupervised domain adaptation in semantic segmentation," *arXiv preprint arXiv:2105.10843*, 2021.

[54] A. Arnab, O. Miksik, and P. H. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 888–897.

[55] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013, p. 896.

[56] W. Shi, Y. Gong, C. Ding, Z. M. Tao, and N. Zheng, "Transductive semi-supervised deep learning using min-max features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 299–315.

[57] K. Yi and J. Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7017–7025.

[58] G.-H. Wang and J. Wu, "Repetitive reprediction deep decipher for semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6170–6177.

[59] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 687–10 698.

[60] M. Nayeem Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," *arXiv e-prints*, pp. arXiv–2101, 2021.

[61] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[62] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[64] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, vol. 2, 2006, pp. 1735–1742.

[65] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.