SRoUDA: Meta Self-Training for Robust Unsupervised Domain Adaptation

Wanqing Zhu[1,2], **Jia-Li Yin**[1,2*], **Bo-Hao Chen**[3], **Ximeng Liu**[1,2*]

[1] Fujian Province Key Laboratory of Information Security and Network Systems, Fuzhou 350108, China
[2] College of Computer Science and Big Data, Fuzhou University, Fuzhou 350108, China
[3] Department of Computer Science and Engineering, Yuan Ze University, Taiwan
wqingzhu00@163.com, jlyin@fzu.edu, bhchen@saturn.yzu.edu.tw, snbnix@gmail.com

Abstract

As acquiring manual labels on data could be costly, unsupervised domain adaptation (UDA), which transfers knowledge learned from a rich-label dataset to the unlabeled target dataset, is gaining increasingly more popularity. While extensive studies have been devoted to improving the model accuracy on target domain, an important issue of model *robustness* is neglected. To make things worse, conventional adversarial training (AT) methods for improving model robustness are inapplicable under UDA scenario since they train models on adversarial examples that are generated by supervised loss function. In this paper, we present a new meta self-training pipeline, named SRoUDA, for improving adversarial robustness of UDA models. Based on self-training paradigm, SRoUDA starts with pre-training a source model by applying UDA baseline on source labeled data and taraget unlabeled data with a developed random masked augmentation (RMA), and then alternates between adversarial target model training on pseudo-labeled target data and fine-tuning source model by a meta step. While self-training allows the direct incorporation of AT in UDA, the meta step in SRoUDA further helps in mitigating error propagation from noisy pseudo labels. Extensive experiments on various benchmark datasets demonstrate the state-of-the-art performance of SRoUDA where it achieves significant model robustness improvement without harming clean accuracy.

Introduction

Deep neural networks (DNNs) have achieved impressive advances across a variety of machine learning tasks. However, these leaps come only when sufficient and well-labeled training data is available. For various label-scarce real-world scenarios, Unsupervised Domain Adaptation (UDA) (Long et al. 2015; Zhang et al. 2019b; Park et al. 2022; Fan et al. 2022) is widely studied and used. This process starts with a label-rich source dataset, and then transfers knowledge learned from the source domain to the target domain, usually by minimizing the distribution discrepancy between source and target domains.

While UDA approaches greatly reduce data requirements and are more practical for real-world scenarios, it neglects an important issue of model *robustness*. Recent studies (Szegedy et al. 2014; Madry et al. 2018; Sehwag et al. 2022) have revealed that the DNN models are vulnerable to adversarial attacks, i.e., maliciously hand-crafted images which are similar looking to the original images but can lead to dramatic changes in model predictive behavior. For instance, given the UDA task $\mathbf{A} \rightarrow \mathbf{W}$, the model trained with UDA baseline can achieve 94.50% accuracy on clean target data; however, the model robustness against adversarial examples is 0.00%, as shown in the upper row of Figure 1. Obviously, the robustness vulnerability of DNNs significantly hinders their applications in many real-world scenarios. Approaches to effectively improve the adversarial robustness of

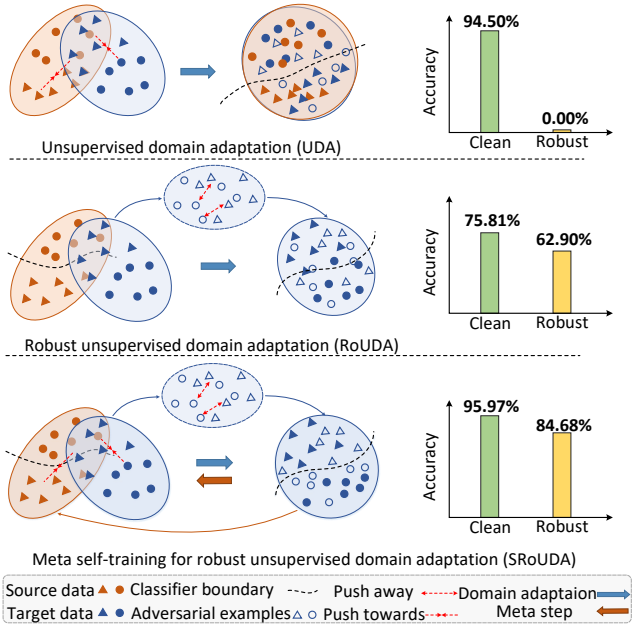

Figure 1: Overview of different UDA schemes and their performance. Upper row: Conventional UDA. Middle row: Naive self-training for injecting AT into UDA. Bottom row: Our proposed SRoUDA. The statics are tested on ResNet-50 backbone on A $\rightarrow$ W task in Office-31 dataset. We use MDD (Zhang et al. 2019b) as the UDA baseline, and PGD-20 for evaluating model robustness.

*Corresponding author.

models produced by UDA is highly demanded.

In contrast to the intensive studies on UDA, few efforts have been devoted to explore the robustness of UDA models. Recall that the most effective defense for adversarial attacks so far is adversarial training (AT), of which the core idea is to train the task model on adversarial examples that are online generated at each training epoch. However, it needs ground-truth labels to generate adversarial examples, which is inapplicable for target data under the UDA scenario. A recent work (Awais et al. 2021) proposed to utilize an external adversarially pre-trained model as a teacher model to distill robustness knowledge during UDA process. However, its performance is limited by the teacher model's perturbation budget and sensitive to the architecture of teacher model.

Recently, self-training (Xu et al. 2019; Yang et al. 2021b), which trains the model on unlabeled target data in a supervised manner by utilizing pseudo labels generated from a pre-trained source model, has become popular as means of learning potential training signals in target domain for UDA. The insights of pseudo label generation in self-training provide us an access for directly injecting AT in UDA where we can generate adversarial examples of target data with pseudo labels. However, we find that naive self-training does not work well in robust UDA training as shown in the middle row of Figure 1. Although the model robustness can be improved, the clean accuracy decreased dramatically due to the inevitable noisy labels generated from source model and the odds between clean and robustness accuracy native in AT (Zhang et al. 2019a).

In this paper, we propose SRoUDA, redesigning the self-training pipeline, for improving adversarial robustness of UDA models. First, in source model pre-training, we apply the UDA technique on source and target data with a developed random mask augmentation (RMA), to overcome the domain bias and initialize more proper pseudo labels for target data. Second, for target model training, we directly inject AT to train the target model using the adversarial examples generated from pseudo-labeled target data. The main challenge in this phase is how to improve pseudo label quality. Inspired by (Wang et al. 2021b; Zhang et al. 2022), instead of using arbitrary pseudo labels for target model training, we form the pseudo label generation as an optimization problem and employ meta-learning with a designed meta-objective: the best pseudo labels should make the target model the best. Thus we propose a meta step in this stage where the source model is progressively updated by learning from the feedback of how the target network performs. In our implementation, the feedback signal is the performance of the target model on the labeled source data. It can reflect how much and how well the target model learns from the source model and brings alignment of source data with adversarial examples of target data, which consequently improves the target model's performance. Specifically, the target and source models are trained alternatively: the target model learns robust knowledge from the pseudo-labeled target data, and then the source model is fine-tuned by the target model loss on the labeled source dataset.

We perform extensive experiments on various benchmarks and demonstrate the effectiveness of our approach, where the model robustness is improved by a large percentage ($0.00\% \rightarrow 84.68\%$ in the bottom row of Figure 1). Besides, the proposed method outperforms the state-of-the-art approach (Awais et al. 2021) by a notable margin on all evaluated settings. It is also noteworthy that our approach can even achieve a higher clean accuracy than the UDA baseline in several tasks.

Related Works

Unsupervised Domain Adaptation

Unsupervised domain adaptation aims to transfer the knowledge learned from a labeled source data to an unlabeled target data. Conventional UDA approaches explore domain-invariant information across source and target data by minimizing the learned distribution discrepancy. Long *et al*. proposed DAN (Long et al. 2015) and JAN (Long et al. 2017) to minimize the feature discrepancy using Maximum Mean Discrepancy (MMD). More recently, the emergence of GANs has brought new inspiration to the field of domain adaptation, the DANN (Ganin et al. 2016), CDAN (Long et al. 2018), and MCD (Saito et al. 2018) are proposed where a discriminator is equipped to force more discriminative domain-invariant feature generation. Despite the effectiveness of these methods, a drawback emerges that the potentially meaningful training information from the target domain are under-utilized. Thus, another line of works explored self-training scheme to generate pseudo labels for target domain and then re-train the model by pseudo-labeled target data. To improve the quality of pseudo labels, many efforts are devoted to reduce label noise by utilizing progressive generation strategy (Xu et al. 2019), curriculum learning (Choi et al. 2019), and voting scheme (Yang et al. 2021b; Fan et al. 2022).

Adversarial Robustness

Since Szegedy *et al*. (Szegedy et al. 2014) first reported that imperceptible perturbations can easily fool deep models, the adversarial vulnerability of deep models has gained increasing concerns. The work of (Goodfellow, Shlens, and Szegedy 2015) designed a Fast Gradient Sign Method (FGSM) to produce strong adversarial examples based on the investigation of CNN linear nature. Madry *et al*. (Madry et al. 2018) further proposed Projected Gradient Descent (PGD) attack by changing the one-step perturbation generation in FGSM into iterative perturbation generation, and has become one of the most classic adversarial attacks. In response to the threat of adversarial examples, AT has been developed as a paradigm to train robust models. It is formed as a min-max game between adversarial example generation and model training. The work (Madry et al. 2018) first formulated AT process, where they used PGD to generate adversarial examples and trained the model on these adversarial examples. Various modifications are developed to improve robustness accuracy of AT, with changes in the adversarial examples generation procedure (Tramer et al. 2018; Kannan, Kurakin, and Goodfellow 2018; Zheng et al. 2020), model parameter updating (Hwang et al. 2021) and feature adaption (Zhang et al. 2019a; Wang et al. 2021a). However,

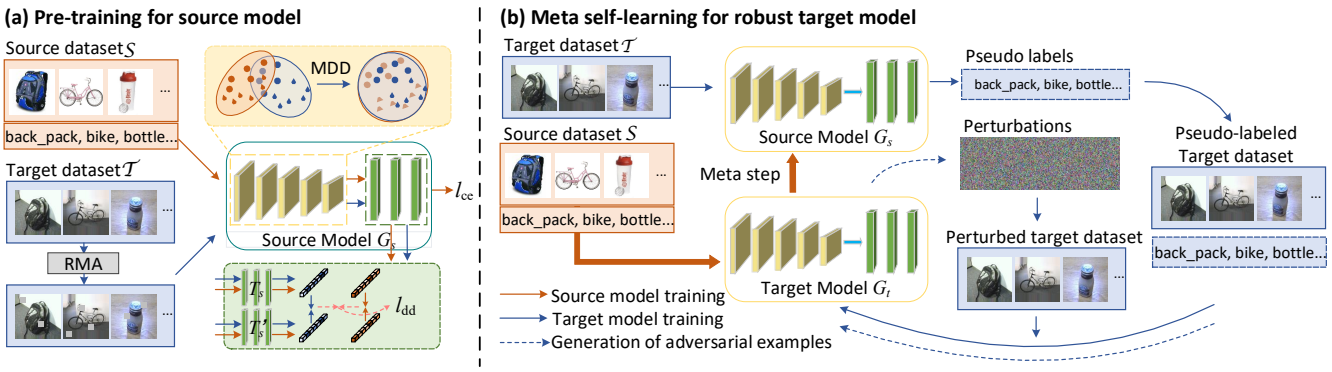

Figure 2: An overview of our SRoUDA pipeline, which consists of two phases: (a) Pre-train the source model G_s with RMA in target domain to mitigate the domain bias. (b) Train the target model on adversarial examples generated by pseudo-labeled target data, while fine-tuning the source model by employing a meta-step which is the feedback of the target model's performance on source labeled data.

AT requires labels and therefore not applicable under UDA settings.

Adversarial Robustness of UDA Models

In contrast to intensive studies on improving accuracy of UDA models, few efforts have been made to explore the adversarial robustness for UDA. Meanwhile, the main challenges for incorporating AT in UDA is the missing label information in target domain while AT needs ground-truth labels for generating adversarial examples. To address this issue, existing methods either skip the AT (Awais et al. 2021) or use self-supervised methods (Lo and Patel 2022) to generate adversarial examples. For instance, Awais *et al.* (Awais et al. 2021) directly explored the robustness transfer in UDA process instead of using the AT, and proposed to use an external pre-trained robust model for robust feature distillation during UDA process. Despite its effectiveness, its performance is limited by the teacher model's perturbation budget and sensitive to the architecture of teacher model. On the other hand, Lo *et al.* (Lo and Patel 2022) proposed to use a self-supervised adversarial example generation for injection of AT into UDA. Sadly, such adversarial example generation cannot guarantee the inner-maximization in AT, thus leading to unsatisfied model robustness. Another naive self-training based method was proposed in (Yang et al. 2021a) for image segmentation. Likewise, such simple pseudo label generation is not correct enough and might even bring an unsatisfied robustness, as we stated in previous section.

Preliminaries

Problem setting. We set our problem under UDA scenario, where we have accessed to a labeled source dataset $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^n$ and an unlabeled target dataset $\mathcal{T} = \{(x_i^t)\}_{i=1}^m$. Considering a classification model composed of a feature extractor, $F : x \to f$, where f is the feature representation of original input x, and a classification layer, $T : f \to z$, where z is the logit output, and $z \in \mathbb{R}^C$, C denotes the number of classes. The goal of UDA is to produce a target model with high accuracy on the unlabeled target dataset. Here we go further to achieve not only accurate but also robust model on target domain. We first revisit the conventional UDA and AT algorithms.

UDA. A typical UDA model is optimized by a source error plus a discrepancy metric between the target and the source as follows:

$$\min_{F,T} \mathcal{L}_{uda} = \ell_{ce}(T(F(x^s)), y^s) + \ell_{da}(F(x^s), F(x^t)), \quad (1)$$

where ℓ_{ce} denotes the cross-entropy loss for classification task, ℓ_{da} is the domain adaptation loss defined by different UDA approaches.

AT. An adversarial example is an perturbed image which is obtained by adding a perturbation δ over original data x:

$$\hat{x} = x + \delta, \quad \text{s.t.} \quad \hat{x} \in \mathcal{B}(x), \quad (2)$$

where $\mathcal{B}(x)$ denotes the ℓ_p-norm ball centered at x with radius ϵ, i.e., $\mathcal{B}(x) = \{\hat{x} : \|\hat{x} - x\|_p \leq \epsilon\}$. Correspondingly, the AT process directly takes adversarial examples as training data and has the following objective:

$$\min_{F,T} \mathcal{L}_{at} = \ell_{ce}(T(F(\hat{x})), y). \quad (3)$$

Typically, the adversarial examples in AT are generated using PGD attack in an iterative way as follows:

$$\hat{x}_{k+1} = \Pi_{x+\mathcal{B}}(\hat{x}_k + \alpha \operatorname{sign}(\nabla_x \ell_{ce}(T(F(x)), y)), \quad (4)$$

where $\hat{x}_k$ is initialized as the clean input x, and the final adversarial example $\hat{x} = \hat{x}_{k_{max}}$, where k_{max} is the maximum number of iterations; and Π refers to the projection operation for projecting the adversarial examples back to the norm-ball.

Methodology

We now introduce our SRoUDA for training a not accurate but also robust model under the UDA settings. An overview of our framework is shown in Figure 2. Starting from pre-training a source model G_s by applying UDA baseline on source labeled data $\mathcal{S}$ and taraget unlabeled data $\mathcal{T}$ with random masked augmentation in Figure 2 (a), our SRoUDA alternates between adversarial target model training on pseudo-labeled target data and fine-tuning source model by a meta step in Figure 2 (b).

Source Model Pre-Training

The source model G_s learns how to perform classification task on source labeled data $\mathcal{S}$ and is further adopted to produce pseudo labels $\bar{y}^t$ for the unlabeled target data $\mathcal{T}$. To better initialize pseudo labels for the second phase, we consider the design of source model pre-training from two aspects: 1) reduce the learning bias from the source domain $\mathcal{S}$; 2) improve the model generalization for target domain $\mathcal{T}$. To this end, instead of only training on source data, we adopt UDA baseline to utilize target data for reducing the model learning bias from source domain. Moreover, to improve the generalization of pre-trained model, inspired by the masked autoencoders (MAE) (He et al. 2022), we propose a very simple yet effective augmentation strategy, i.e., RMA, to facilitate the object-aware recognition.

Random masked augmentation. Given a target image $x^t \in \mathbb{R}^{h\times w\times c}$, we first divide the target image into non-overlapping patches $z \in \mathbb{R}^{1\times \frac{hw}{3^2}}$ of size 3×3 . RMA then samples a subset of the patches with a uniform distribution and masks the remaining patches, which can be presented as:

$$z_i^+ = \begin{cases} z_i, & i \in R \\ 0, & i \notin R \end{cases}, \tag{5}$$

where $R \sim U(1, \frac{hw}{3^2})$. Different from the image reconstruction task in MAE, where they set the sampling ratio as 20% for learning content reconstruction, here we set the sampling ratio as 85% to cover most of the image contents for data augmentation. The final augmented image x^{t+} is formed by grouping z_i^+ back into an image. In the pre-training process, the target data is the combination of the original and augmented data.

Pre-training. Before the interaction between source and target models, we first pre-train the source model for a better initialization of pseudo labels for target data. Here we adopt the basic UDA approach Margin Disparity Discrepancy (MDD) to pre-train a clean UDA model. Specifically, MDD first employs an auxiliary classifier T_s' to evaluate the disparity discrepancy as

$$\begin{aligned} \ell_{dd}(x^s, x^t) =& \ell_{ce}(T_s'(F_s(x^t)), T_s(F_s(x^t))) \\ & -\gamma \ell_{ce}(T_s'(F_s(x^s)), T_s(F_s(x^s))), \end{aligned} \tag{6}$$

where γ is the margin factor and is set to be 4. While auxiliary classifier T_s' is trained to maximize the disparity discrepancy loss ℓ_{dd} in Eq. (6), the source model is updated as:

$$\min_{F_s, T_s} \mathcal{L}_{MDD} = \ell_{ce}(T_s(F_s(x^s)), y^s) + \eta \ell_{dd}(x^s, x^t), \tag{7}$$

where η is a regularization factor which can be set to be 0.1 according to (Zhang et al. 2019b). Note that it is not compulsory to choose MDD as the UDA baseline for pre-training here. Please refer to the experimental section for more comparisons of using different UDA baselines in source model pre-training.

Meta Self-Training for Robust Target Model

With the pre-trained source model, the next step is to generate pseudo labels for the unlabeled target data and then the target model can be trained on the adversarial examples generated by the pseudo labeled target data, which can be presented as:

$$\min_{G_t} \mathcal{L} = \ell_{ce}(G_t(\hat{x}^t), \bar{y}^t), \tag{8}$$

where $\hat{x}^t$ is the adversarial example of target data and $\bar{y}^t = G_s(x^t)$ is the pseudo label for x^t. It is obvious that the optimal G_t heavily depends on the source model G_s via the pseudo labels. Considering the noisy labels are inevitable in the initial pseudo labels produced from pre-trained model, we need to progressively fine-tune the source model G_s for better pseudo labels that can make the target model best. We thus consider the pseudo label generation as an optimization problem, and transform Eq. (8) into:

$$\min_{G_t, G_s} \mathcal{L} = \ell_{ce}(G_t(\hat{x}^t), G_s(x^t)). \tag{9}$$

The object in Eq. (9) can be regarded as a joint learning of G_t and G_s. Inspired by meta learning (Pham et al. 2021; Wang et al. 2021b), we design a meta step for source model fine-tuning with a meta-objective that the best pseudo labels should improve the learning of target model, where the feedback signal is the performance of target model on source labeled data. Specifically, in each iteration, the target model G_t is trained on the adversarial examples generated by the target data with pseudo labels that are produced by the source model. Then the source network G_s is fine-tuned by learning from the feedback of the performance of target model on source dataset. By this way, the pseudo labels can be adjusted accordingly to further improve target model's performance. The training steps are as follows:

Algorithm 1: Meta self-training for robust unsupervised domain adaptation (SRoUDA)

Input : Source domain labeled data $\mathcal{S} = \{(x_i^s, y_i^s)\}$, and target domain unlabeled dataset $\mathcal{T} = \{(x_i^t)\}$. Source model G_s, target model G_t, batch size B, learning rate lr, and iteration number N.

Output: The adversarial robust model for target domain G_t.

Pre-train the source model G_s as detailed in Eqs.(5)-(7).
Initialize the target model G_t by copying parameters from the source model G_s.
for *epoch = 1 ... N* **do**
  Sampling a random mini-batch of training samples $x^t \leftarrow \{x_i^t\}_{i=1}^B$, $x^s \leftarrow \{x_i^s, y_i^s\}_{i=1}^B$;
  Generating pesudo labels $\bar{y}^t$ for target data x^t;
  Generating adversarial examples of target data $\hat{x}^t$ based on the pesudo labels by Eq. (10);
  Update G_t by optimizing adversarial training loss in Eq. (11);
  Utilize the current target model G_t to compute meta loss on source data x^s in Eq. (12);
  Update G_s by optimizing the meta loss;
end for

Methods	A → W		D → W		A → D		D → A		W → D		W → A		Avg.	
	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
Baseline (UDA) †	94.50	0.00	98.40	0.00	93.50	0.00	74.60	0.65	100.00	0.00	72.20	1.93	88.90	0.43
Source only	28.23	6.45	45.16	26.61	16.13	6.45	9.46	5.59	62.90	25.81	12.69	5.81	29.10	12.79
AT+UDA	74.09	34.47	91.19	70.57	73.49	19.28	39.65	24.46	98.59	66.87	55.31	35.39	72.02	42.21
UDA+AT	75.81	62.90	91.13	53.23	83.87	43.55	64.52	53.33	95.16	51.61	64.52	53.12	79.78	53.21
RFA (Awais et al. 2021) †	-	-	-	-	-	-	-	-	-	-	-	-	84.21	74.31
SRoUDA (Ours)	95.97	84.68	96.77	83.87	91.94	85.48	72.47	57.20	100.00	88.71	67.10	57.42	87.27	75.79

Table 1: Comparison of clean and robustness accuracy % of different UDA models produced by different methods on Office-31 dataset. Note that † denotes the results are directly copied from the original paper. We only show the average accuracy of RFA since they did not include detailed number in (Awais et al. 2021).

Methods	M → U		U→ M		S→ M		Avg.		CIFAR→STL		STL→CIFAR		Avg.	
	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.	Clean	Rob.
Baseline(UDA)	95.60	12.31	97.36	35.23	89.20	41.19	94.05	29.58	68.82	9.28	59.16	13.29	63.99	11.29
source only	76.13	57.90	54.63	48.59	22.63	15.14	51.13	40.54	42.71	24.08	35.53	21.41	39.12	22.75
AT+UDA	93.42	69.76	96.35	74.83	73.83	67.55	87.87	70.71	50.33	28.51	33.37	18.80	41.85	23.66
UDA+AT	92.43	83.26	97.85	93.56	65.09	63.64	85.12	80.15	58.64	36.86	40.72	25.53	49.68	31.20
SRoUDA (Ours)	95.02	87.59	98.50	96.44	88.72	87.16	94.08	90.40	50.75	31.57	62.04	38.50	56.40	35.04

Table 2: Comparison of clean and robustness accuracy % of different UDA models produced by different methods on Digits and CIFAR datasets.

Step 1. Given fixed pre-trained source model G_s, the target model G_t performs adversarial training on adversarial examples of target data that are generated by pseudo labels produced by G_s. In each iteration, the adversarial examples of target data are first generated by changing Eq. (4) to:

$$\hat{x}^t_{k+1} = \Pi_{\mathcal{B}}(\hat{x}^t_k + \alpha \operatorname{sign}(\nabla_{x^t} \ell_{ce}(G_t(x^t), G_s(x^t))), \quad (10)$$

where $\hat{x}^t = \hat{x}^t_{k_{max}}$. Typically, k_{max} is set as 10 during AT process. Then the target model is further trained on the generated adversarial examples as follows:

$$\min_{G_t} \mathcal{L}_{at} = \ell_{ce}(G_t(\hat{x}^t), G_s(x^t)). \quad (11)$$

Step 2. Next, to improve the quality of pseudo labels, we apply a meta-step to utilize the performance of G_t on source data for fine-tuning the source network. The meta loss is computed as:

$$\mathcal{L}_{meta} = \ell_{ce}(G_t(x^s), y_s). \quad (12)$$

The source model G_s is then fine-tuned by the gradient descent based on $\mathcal{L}_{meta}$. The algorithm of our SRoUDA is summarized in Algorithm 1.

Experiments

Experimental Setup

Datasets. We evaluate our method on the both main-stream UDA benchmark datasets and AT datasets: 1) **Office-31** dataset, which is a standard domain adaptation dataset with three domains: Amazon (**A**, 2,817 images), Webcam (**W**, 795 images), and DSLR (**D**, 498 images), It is imbalanced across domains. 2) **Digits** dataset containing 3 different domains: MNIST (**M**), USPS (**U**), and SVHN (**S**). Note that the images in **M**, **U** are gray-scale, whereas the images in **S** are colored. 3) **CIFAR** and **STL** datasets. Both datasets contain 10 categories, of which the overlapping categories are 9 categories. We remove the different categories in the two datasets and changing the 10-category classification task into 9-category task.

Compared methods. We compare our SRoUDA with four baselines: (1) UDA baseline: UDA method without considering model robustness; (2) Source only: the model is adversarially trained on only source data; (3) UDA+AT: injecting AT into UDA process based on the naive self-training pipeline, where a source model is first pre-trained by UDA, and then the target model is adversarially trained on pseudo-labeled target data; (4) AT+UDA: the source data is first transferred into adversarial examples, and then perform UDA on adversarial source data and clean target data. We also compare with the state-of-the-art method RFA using their reported results in (Awais et al. 2021).

Implementation details. We validate our proposed SRoUDA on three backbones for fair comparison with SOTAs. Specifically, we use ResNet-50 on **Office-31**, DTN on **Digits** dataset, and WideResNet-50-2 on **CIFAR** dataset. During the pre-training of source model, we adopt the training settings of the popular UDA codebase DALIB and train the source model for 20 epochs with a learning rate of 0.004. For the following meta self-training stage, we iteratively update the source and target models. In the AT process, we set $k_{max} = 10$, $\epsilon = 8/255$ in adversarial

Method	STL → CIFAR-10				
	Clean	FGSM	PGD-10	PGD-20	CW_∞
Baseline	59.16	32.07	16.73	13.29	3.59
Source only	36.89	20.61	13.47	12.11	12.23
AT+UDA	33.37	21.96	19.28	18.80	12.56
UDA+AT	40.72	40.15	25.54	25.53	24.21
SRoUDA (Ours)	**62.24**	**41.47**	**38.82**	**38.50**	**37.23**

Table 3: Robustness comparison against different adversarial attacks on STL → CIFAR-10 task.

Method	W → A		D → A	
	Clean	Rob.	Clean	Rob.
SRoUDA w/o pre-train	50.75	40.22	40.43	22.15
SRoUDA w/o meta-step	64.52	53.12	68.17	54.84
SRoUDA w/o RMA	63.44	51.40	72.04	52.90
SRoUDA	**67.10**	**57.42**	**72.47**	**57.20**

Table 4: Experimental results on component ablations of SRoUDA.

example generation, the Adam optimizer with learning rate 0.0015 is used to update the target model. We update the source model every epoch in this process. During both the pre-training and meta self-training processes, we also adopt the widely used data augmentation, including random flipping, and rotation for avoiding overfitting.

Main Results

Overall results. We present the comprehensively comparison with different methods in Table 1-2. Here we use MDD as the UDA baseline and PGD-20 attack to test the models' robustness.

From Table 1-2, we can have the following observations. First, the proposed SRoUDA can effectively improve adversarial robustness of UDA models, which significantly outperforms the other UDA baselines. Specifically, we improve the performance of model robustness on all the datasets by a large margin, e.g., 0.43% to 75.79% on average for **Office-31**; 29.58% to 90.40% on average for **Digits**. Furthermore, when tested on small-scale datasets such as **Digits**, our method can achieve both near-optimal accuracy and robustness simultaneously on target domain, i.e., 98.50% clean and 96.44% robustness accuracy in **U** → **M** task. These encouraging results validate that our SRoUDA can effectively enhance the adversarial robustness of UDA models and perform generally well on different domain adaptation tasks.

Second, compared to other schemes for improving UDA robustness, our method achieves the state-of-the-art model robustness without harming clean accuracy. Although the naive self-training UDA can improve the robustness, the clean accuracy is dramatically dropped, e.g., 88.90% to 79.78% on average for **Office-31**; and 94.05% to 85.12% on average for **Digits**. This risk comes from both the pseudo label noises and the odds between accuracy and robustness native in AT. On the contrary, our method can even improve the clean accuracy on some tasks, e.g., for **Office-31** dataset, SRoUDA improves clean accuracy over the UDA baseline on **A** → **W** (94.50% to 95.97%); for **Digits** dataset, our method improves **U** → **M** (97.36% to 98.50%). This benefits from the meta-step for fine-tuning source model in target model training with the objective to perform well on source data, which inherently aligns the source data with the adversarial examples of target data during target model training.

UDA baselines	W → D		D → A	
	Clean	Rob.	Clean	Rob.
DAN	91.94	82.26	59.78	43.23
DANN	98.39	83.87	68.60	49.25
JAN	96.77	87.10	62.58	48.82
CDAN	98.39	72.58	70.97	51.61
MDD	**100.00**	**88.71**	**72.47**	**57.20**

Table 5: Comparison of using different UDA baselines in source model pre-training on W → D and D → **A** tasks.

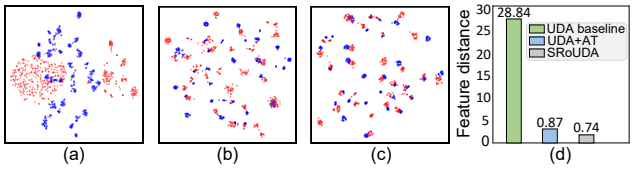

Figure 3: (a)-(c): The t-SNE visualization of extracted features from target models trained with UDA baseline, UDA+AT with naive self-training, and our method on W → D task, respectively. Theblue dots denote clean target data and red dots denote the adversarial examples of target data. (d): The mean L_2-norm distance of clean and adversarial examples in feature space.

We also test the model robustness against different adversarial attacks on **STL** → **CIFAR-10** task in Table 3. Here, the FGSM, PGD-10, PGD-20, CW_∞ attacks are used to evaluate the model robustness. It can be observed that the model produced by our SRoUDA can achieve higher robustness accuracy on all attack settings, demonstrating that our method can produce robust models against multiple attacks, which is crucial for practical employment of DNNs.

Analysis

We analyze our SRoUDA from four perspectives: (1) component ablations of the SRoUDA pipeline, (2) sensitivity to the UDA techniques in the pre-training, (3) feature space analysis, and (4) training convergence.

Component ablations. We examine the effectiveness of individual components in our SRoUDA, and conduct the comparison on **Office-31** **W** → **A** and **D** → **A** tasks following the experimental settings described previously. The results are shown in Table 4. Specifically, we testify (1) SRoUDA

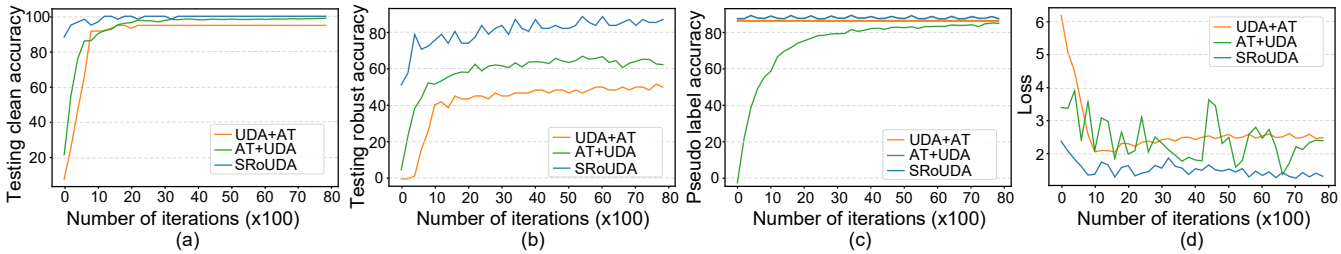

Figure 4: The training curve of different training schemes. (a) Testing clean accuracy convergence, (b) testing robust accuracy convergence, (c) pseudo label accuracy convergence, and (d) training loss convergence. Note that we only show the training convergence of target model training process.

w/o pre-train: removing the source model pre-training and randomly initialize the source model parameters.We can clearly see that without pre-training on the source model, both the clean and robustness accuracy drop by a large margin because of the misleading initialized pseudo labels, which indicates the pre-training of source model is crucial for self-training pipeline. (2) SRoUDA w/o meta-step: removing the meta learning step and use fixed pseudo labels for training robust target model. Using pre-trained source model but without fine-tuning it, SRoUDA w/o meta-step performs better than SRoUDA w/o pre-train but still suffers from over 10% degradation on both clean and adversarial accuracy compared with SRoUDA. (3) SRoUDA w/o RMA: removing the data augmentations. Without data augmentation, our SRoUDA suffers degradation both the clean and robustness accuracy in $\mathbf{W} \rightarrow \mathbf{A}$ task, and $\mathbf{D} \rightarrow \mathbf{A}$ task. In general, the full SRoUDA pipeline can achieve both the highest clean and robustness accuracy on target domain.

Different UDA techniques. Besides using MDD as the UDA baseline, we investigate our SRoUDA pipeline with different UDA techniques. Here we use DAN, DANN, JAN, and CDAN to replace the MDD in source model pre-training of our SRoUDA pipeline. The results are reported in Table 5. Both the clean and robustness accuracy vary as the change of UDA baselines in source model pre-training, which once again indicates that the model pre-training plays a crucial role in self-training pipelines. Based on the performance of different UDA techniques in natural training, we can see that better UDA baseline can help produce more robust target model since the pseudo labels are more reliable.

Feature space analysis. We first visualize the feature generalization in the target model trained by different methods using t-SNE embeddings in Figure 3 (a)-(c). The features are extracted from the last convolution layer of the target model. As we can see, the adversarial examples reside in a large region that are hard to distinguish in natural trained UDA model (Figure 3 (a)). By applying naive self-training scheme on UDA, the adversarial examples are discriminated better by incorporating adversarial training but are not well aligned with the clean data due to the bias of pseudo labels. In contrast, our method is evidently better and the categories are well discriminated. We further compute the mean L_2-norm distance between the clean target data and its corresponding adversarial examples in the feature space as $\|F_t(x^t) - F_t(\hat{x}^t)\|_2$. The results are given in Figure 3 (d). As we can observe, the UDA baseline has the largest distance as it does not consider the model robustness. Incorporating AT can greatly minimize the distance, while our SRoUDA achieves the lowest distance.

Convergence. We testify the convergence of UDA+AT, AT+UDA and SRoUDA with the testing clean accuracy, testing robustness accuracy, pseudo label accuracy, and loss function on task $\mathbf{W} \rightarrow \mathbf{D}$ shown in Figure 4. Note that we only plot the training convergence of target model in SRoUDA. With a sophisticated pre-trained source model, SRoUDA enjoys a faster convergence than UDA+AT and AT+UDA by taking the source model as initialization. For UDA+AT, the pseudo label generation is fixed thus the performance of clean and robustness accuracy is limited. For AT+UDA, as minimizing the discrepancy between source adversarial examples and clean data, the clean accuracy on target data can be improved. However, due to the inherent difference between source adversarial examples and target adversarial examples, the robustness accuracy on target data is less satisfied. In contrast, by equipping with a meta step to progressively adjust the pseudo labels, the clean and robustness accuracy of SRoUDA can be improved steadily as the training goes deeper shown in Figure 4 (a) and (b). Although the pseudo label accuracy does not improve that much during the fine-tuning of source model in Figure 4 (c), but the loss continues to be lower in Figure 4 (d). This phenomenon verifies the view of Zhang et al (Zhang et al. 2022), adversarial robustness can be enhanced by noisy label injection, they are optimized towards a better robust target model training.

Conclusion

In this paper, we tackle the problem of model robustness of unsupervised domain adaption models. We have presented SRoUDA, a redesigned self-training pipeline for robust unsupervised domain adaptation. SRoUDA involves a pre-training stage with a simple but effective random masked augmentation for source model, and a meta self-training stage for alternatively training robust target model and fine-tuning source model. Extensive experiments demonstrate that SRoUDA can effectively improve the robustness of UDA models by a large margin over baselines under various defense settings.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China under Grant Nos. 62202104, 62102422, 62072109 and U1804263; the Ministry of Science and Technology, Taiwan, under Grant MOST 111-2628-E-155-003-MY3; and Youth Foundation of Fujian Province, P.R.China, under Grant No. 2021J05129.

References

Awais, M.; Zhou, F.; Xu, H.; Hong, L.; Luo, P.; Bae, S.-H.; and Li, Z. 2021. Adversarial Robustness for Unsupervised Domain Adaptation. In *Proc. Int'l Conf. Computer Vision*, 8548–8557.

Choi, J.; Jeong, M.; Kim, T.; and Kim, C. 2019. Pseudo-Labeling Curriculum for Unsupervised Domain Adaptation. In *Proc. British Conf. Machine Vision*.

Fan, H.; Chang, X.; Zhang, W.; Cheng, Y.; Sun, Y.; and Kankanhalli, M. 2022. Self-Supervised Global-Local Structure Modeling for Point Cloud Domain Adaptation With Reliable Voted Pseudo Labels. In *Proc. Conf. Computer Vision and Pattern Recognition*, 6377–6386.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.*, 17(1): 2096–2030.

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *Proc. Int'l Conf. Learning Representations*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked Autoencoders Are Scalable Vision Learners. In *Proc. Conf. Computer Vision and Pattern Recognition*, 16000–16009.

Hwang, J.-w.; Lee, Y.; Oh, S.; and Bae, Y. 2021. Adversarial Training With Stochastic Weight Average. In *Proc. Int'l Conf. Image Processing*, 814–818.

Kannan, H.; Kurakin, A.; and Goodfellow, I. 2018. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*.

Lo, S.-Y.; and Patel, V. M. 2022. Exploring Adversarially Robust Training for Unsupervised Domain Adaptation. *arXiv preprint arXiv:2202.09300*.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning Transferable Features with Deep Adaptation Networks. In *Proc. Int'l Conf. Machine Learning*, 97–105.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional Adversarial Domain Adaptation. In *Proc. Neural Information Processing Systems*.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *Proc. Int'l Conf. Machine Learning*, 2208–2217.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *Proc. Int'l Conf. Learning Representations*.

Park, H.; Yessenbayev, A.; Singhal, T.; Adhikari, N. K.; Zhang, Y.; Borse, S. M.; Cai, H.; Pandey, N. P.; Yin, F.; Mayer, F.; Calidas, B.; and Porikli, F. 2022. Real-Time, Accurate, and Consistent Video Semantic Segmentation via Unsupervised Adaptation and Cross-Unit Deployment on Mobile Device. In *Proc. Conf. Computer Vision and Pattern Recognition*, 21431–21438.

Pham, H.; Dai, Z.; Xie, Q.; and Le, Q. V. 2021. Meta Pseudo Labels. In *Proc. Conf. Computer Vision and Pattern Recognition*, 11557–11568.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In *Proc. Conf. Computer Vision and Pattern Recognition*, 3723–3732.

Sehwag, V.; Mahloujifar, S.; Handina, T.; Dai, S.; Xiang, C.; Chiang, M.; and Mittal, P. 2022. Robust Learning Meets Generative Models: Can Proxy Distributions Improve Adversarial Robustness? In *Proc. Int'l Conf. Learning Representations*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *Proc. Int'l Conf. Learning Representations*.

Tramer, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2018. Ensemble adversarial training: Attacks and defenses. In *Proc. Int'l Conf. Learning Representations*.

Wang, T.; Zhang, R.; Chen, X.; Zhao, K.; Huang, X.; Huang, Y.; Li, S.; Li, J.; and Huang, F. 2021a. Adaptive Feature Alignment for Adversarial Training. *arXiv preprint arXiv:2105.15157*.

Wang, Y.; Mukherjee, S.; Chu, H.; Tu, Y.; Wu, M.; Gao, J.; and Awadallah, A. H. 2021b. Meta Self-Training for Few-Shot Neural Sequence Labeling. 1737–1747.

Xu, R.; Li, G.; Yang, J.; and Lin, L. 2019. Larger Norm More Transferable: An Adaptive Feature Norm Approach for Unsupervised Domain Adaptation. In *Proc. Int'l Conf. Computer Vision*.

Yang, J.; Li, C.; An, W.; Ma, H.; Guo, Y.; Rong, Y.; Zhao, P.; and Huang, J. 2021a. Exploring Robustness of Unsupervised Domain Adaptation in Semantic Segmentation. In *Proc. Int'l Conf. Computer Vision*, 9174–9183.

Yang, J.; Shi, S.; Wang, Z.; Li, H.; and Qi, X. 2021b. ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection. In *Proc. Conf. Computer Vision and Pattern Recognition*, 10363–10373.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019a. Theoretically principled trade-off between robustness and accuracy. In *Proc. Int'l Conf. Machine Learning*, 12907–12929.

Zhang, J.; Xu, X.; Han, B.; Liu, T.; Cui, L.; Niu, G.; and Sugiyama, M. 2022. NoiLin: Improving adversarial training and correcting stereotype of noisy labels. *J. Mach. Learn. Res.*

Zhang, Y.; Liu, T.; Long, M.; and Jordan, M. 2019b. Bridging Theory and Algorithm for Domain Adaptation. In *Proc. Int'l Conf. Machine Learning*, 7404–7413.

Zheng, H.; Zhang, Z.; Gu, J.; Lee, H.; and Prakash, A. 2020. Efficient adversarial training with transferable adversarial examples. In *Proc. Conf. Computer Vision and Pattern Recognition*, 1178–1187.