Exploring Adversarially Robust Training for Unsupervised Domain Adaptation

Shao-Yuan Lo and Vishal M. Patel

Dept. of Electrical and Computer Engineering, Johns Hopkins University
{sylo, vpatel36}@jhu.edu

Abstract. Unsupervised Domain Adaptation (UDA) methods aim to transfer knowledge from a labeled source domain to an unlabeled target domain. UDA has been extensively studied in the computer vision literature. Deep networks have been shown to be vulnerable to adversarial attacks. However, very little focus is devoted to improving the adversarial robustness of deep UDA models, causing serious concerns about model reliability. Adversarial Training (AT) has been considered to be the most successful adversarial defense approach. Nevertheless, conventional AT requires ground-truth labels to generate adversarial examples and train models, which limits its effectiveness in the unlabeled target domain. In this paper, we aim to explore AT to robustify UDA models: How to enhance the unlabeled data robustness via AT while learning domain-invariant features for UDA? To answer this question, we provide a systematic study into multiple AT variants that can potentially be applied to UDA. Moreover, we propose a novel Adversarially Robust Training method for UDA accordingly, referred to as ARTUDA. Extensive experiments on multiple adversarial attacks and UDA benchmarks show that ARTUDA consistently improves the adversarial robustness of UDA models. Code is available at https://github.com/shaoyuanlo/ARTUDA

1 Introduction

Recent advances in image recognition have enjoyed remarkable success via deep supervised learning [12,17,41]. However, the domain shift problem is very common in real-world applications, *i.e.*, source and target domains have different data characteristics. Furthermore, it is costly and labor-intensive to collect the ground-truth labels of target data. To address this issue, Unsupervised Domain Adaptation (UDA) methods have been developed in which the objective is to transfer the knowledge from a labeled source dataset to an unlabeled target dataset. Most existing UDA approaches rely on minimizing distribution discrepancy between source and target domains to learn domain-invariant representations [7,8,22,23,24,34]. Although these approaches achieve impressive performance, they do not consider the robustness against adversarial attacks [3,32], which causes critical concerns.

Adversarial attacks pose serious security risks to deep networks. In other words, deep networks could suffer from dramatic performance degradation in the

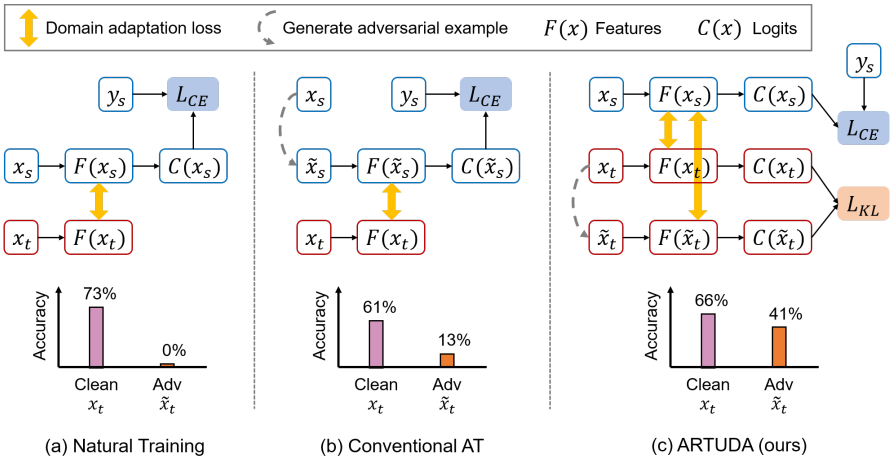


Fig. 1. Overview of the proposed ARTUDA and its importance. L_{CE}: Cross-entropy loss. L_{KL}: KL divergence loss. Compared to conventional AT [26], ARTUDA significantly improves adversarial robustness while maintaining decent clean accuracy. We use DANN [8] with ResNet-50 [12] backbone, the VisDA-2017 [29] dataset, and the PGD-20 [26] attack for this experiment.

presence of carefully crafted perturbations. To defend against adversarial attacks, various defense mechanisms have been proposed [10,11,16,20,26,30,42]. Currently, Adversarial Training (AT) based defenses [10,16,26,42] have been considered the most effective, especially under the white-box setting [1]. The core idea is to train a model on adversarial examples that are generated on the fly according to the model's current parameters. Nevertheless, conventional AT requires ground-truth labels to generate adversarial examples. This makes it not applicable to the UDA problem since UDA considers the scenario that label information is unavailable to a target domain. A nearly contemporary work [2] resorts to external adversarially pre-trained ImageNet models as teacher models to distill robustness knowledge. However, its performance is highly sensitive to the teacher models' perturbation budget, architecture, *etc.*, which limits the flexibility in a wide range of uses. Another very recent work [40] uses an external pre-trained UDA model to produce pseudo labels for doing AT on target data. Unfortunately, we show that it suffers from suboptimal accuracy and robustness against white-box attacks.

Given the above observations, intuitive questions emerge: *Can we develop an AT algorithm specifically for the UDA problem? How to improve the unlabeled data robustness via AT while learning domain-invariant features for UDA?* In this paper, we seek to answer these questions by systematically studying multiple AT variants that can potentially be applied to UDA. First, we apply a conventional AT [26] to an UDA model to see its effectiveness. In other words, the AT is performed on only the labeled source data. Second, inspired by [16,42], we attempt to train models by minimizing the difference between the output logits of clean target data and the corresponding adversarial examples. With this, we can conduct a kind of AT directly on the target data in a self-supervised manner. We call it *Self-Supervised Adversarial Training* or *Self-Supervised AT*. Next, we look into the effects of clean images and adversarial examples in the AT for UDA. We present the trade-off behind different AT variants. Last, we observe that Batch Normalization (BN) [13] plays an important role in the AT for UDA. The feature statistic estimations at training time would affect an UDA model's robustness.

Through these investigations, we propose a novel Adversarially Robust Training method for UDA accordingly, referred to as ARTUDA. It uses both source and target data for training and does not require target domain labels, so it is feasible for UDA. Moreover, it does not need guidance from external models such as adversarially pre-trained models and pre-trained UDA models. Fig. 1 illustrates an overview and the importance of the proposed ARTUDA. The naturally trained (*i.e.*, train with only clean data) model's accuracy decreases to 0% under an adversarial attack. Conventional AT [26] improves robust accuracy to 13% but sacrifices clean accuracy. As can be seen, ARTUDA significantly increases robust accuracy to 41% while maintaining better clean accuracy. This shows that our method can improve unlabeled data robustness and learn domain-invariant features simultaneously for UDA. To the best of our knowledge, ARTUDA is the first AT-based UDA defense that is robust against white-box attacks. In Sec. 5, we extensively evaluate ARTUDA on five adversarial attacks, three datasets and three different UDA algorithms. The results demonstrate its wide range of effectiveness.

Our main contributions can be summarized as follows: (i) We provide a systematic study into various AT methods that are suitable for UDA. We believe that such experimental analysis would provide useful insight into this relatively unexplored research direction. (ii) We propose ARTUDA, a new AT method specifically designed for UDA. To the best of our knowledge, it is the first AT-based UDA defense method that is robust against white-box attacks. (iii) Comprehensive experiments show that ARTUDA consistently improves UDA models' adversarial robustness under multiple attacks and datasets.

2 Related Work

Unsupervised domain adaptation. UDA considers the scenario that a source dataset contains images with category labels, while label information is unavailable to a target dataset. Most popular approaches attempt to transfer knowledge from the labeled source domain to the unlabeled target domain [7,8,22,23,24,34]. DANN [8] proposes to use a domain discriminator that distinguishes between source and target features, and the feature extractor is trained to fool it via GAN [9] learning scheme. ADDA [34] combines DANN with discriminative feature learning. CDAN [23] extends DANN using a class-conditional adversarial game. JAN [24] aligns the joint distributions of domain-specific layers between two domains. Nevertheless, these works do not take adversarial robustness into consideration.

RFA [2] and ASSUDA [40] are the most related works in the literature, which are nearly contemporary with our work. They are the first to focus on UDA's adversarial robustness, but we would like to point out the clear differences from our work. RFA leverages external adversarially pre-trained ImageNet models as teacher models to distill robustness knowledge. Its performance is highly sensitive to the teacher models' setup, such as perturbation budget, architecture and the number of teachers. AT on ImageNet is very expensive, so it is not always easy to obtain the preferred teacher models. In contrast, we propose a method that directly performs AT on a given UDA task, enjoying maximum flexibility. ASSUDA aims at semantic segmentation and considers only weak black-box attacks. It employs an external pre-trained UDA model to produce pseudo labels for target data, then uses the pseudo labels to do AT. However, we show that this approach has suboptimal accuracy and robustness against white-box attacks. In contrast, our method is robust under both black-box and white-box settings.

Adversarial attack and defense. Previous studies reveal that deep networks are vulnerable to adversarial examples [3,32]. Many adversarial attack algorithms have been proposed, such as Fast Gradient Sign Method (FGSM) [10], Projected Gradient Descent (PGD) [26], Momentum Iterative FGSM (MI-FGSM) [6] and Multiplicative Adversarial Example (MultAdv) [21].

Various adversarial defense mechanisms have also been introduced, where AT-based defenses [10,16,26,42] are considered the most effective, especially under the white-box setting [1]. AT trains a model on adversarial examples that are generated on the fly according to the model's current parameters. The most commonly used AT approaches include Madry's AT scheme (PGD-AT) [26] and TRADES [42]. PGD-AT formulates AT as a min-max optimization problem and trains a model with only adversarial examples. TRADES minimizes a regularized surrogate loss to obtain a better trade-off between robustness and performance, where both clean data and adversarial examples are used for training. However, AT requires the labels of input images to generate the corresponding adversarial examples, which can not be directly applied to the UDA problem. In this work, we propose a new AT method specifically designed for UDA.

3 Preliminary

UDA. Given a labeled source dataset $\mathbb{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and an unlabeled target dataset $\mathbb{D}_t = \{x_t^i\}_{i=1}^{n_t}$ with n_s and n_t number of samples, respectively, a typical UDA model learns a feature extractor F and a classifier C on top of F. Given an input image x, we express its feature space representation as $F(x)$ and its output logits as $C(x)$, where we use $C(x)$ as a simplification of the formal expression $C(F(x))$. The objective function of an UDA model can be written as:

$$\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{DA}\big(x_s, x_t\big), \tag{1}$$

where $\mathcal{L}_{CE}$ is the standard cross-entropy loss, and $\mathcal{L}_{DA}$ is the domain adaptation loss defined by each UDA approach. One of the most common $\mathcal{L}_{DA}$ is the adversarial loss introduced by DANN [8], which is defined as:

$$\mathcal{L}_{DA}\big(x_s, x_t\big) = \mathbb{E}[logD(F(x_s))] + \mathbb{E}[1 - (logD(F(x_t)))], \tag{2}$$

where D is a domain discriminator used to encourage domain-invariant features.
AT. PGD-AT [26] is one of the most commonly-used AT algorithm. It formulates AT as a min-max optimization problem and trains models on adversarial examples exclusively:

$$\min_{F,C} \mathbb{E}\left[\max_{\|\delta\|_p \le \epsilon} \mathcal{L}\big(C(\tilde{x}), y\big)\right], \tag{3}$$

where $\tilde{x} = x + \delta$ is the generated adversarial example of x, and ϵ denotes an L_p-norm perturbation budget. Here, δ corresponds to the adversarial noise which is added to x to make it adversarial. For image classification tasks, $\mathcal{L}$ is the cross-entropy loss $\mathcal{L}_{CE}$. PGD [26] is the most widely-used attack approaches. It generates $\tilde{x}$ in an iterative way:

$$x^{j+1} = \mathit{\Pi}_{\|\delta\|_p \le \epsilon}\left(x^j + \alpha \cdot sign(\triangledown_{x^j}\mathcal{L}(C(x^j), y))\right); \tag{4}$$

and $\tilde{x} = x^{j_{max}}$, where j_{max} denotes the maximum number of attack iterations. FGSM [10] can be treated as a single-step and non-random start version of PGD.

4 Exploring AT for UDA

In this section, we systematically study multiple variants of AT to explore suitable AT methods for UDA. Then we finalize the proposed ARTUDA accordingly. Here we conduct a set of experiments on the VisDA-2017 [29] dataset. We employ DANN [8] as the UDA algorithm with ResNet-50 [12] backbone. The white-box FGSM [10] attack with perturbation budget of $\epsilon = 3$ is used for both AT and testing. Following the practice of [2,40], we assume that attackers have the labels of the target dataset to generate adversarial examples. The rationale behind these settings is that (i) most existing UDA approaches [23,34] are based on DANN's key idea, so DANN is a fair representative; (ii) the white-box threat model is the strongest attack setting, which has been considered a standard evaluation protocol for defenses [26,42,1,39].

4.1 Conventional AT on UDA

We start with applying a conventional AT [26] to DANN to see its effectiveness. That is, the AT is performed on only the labeled source data, *i.e.*, apply Eq. (3) on source dataset $\mathbb{D}_s$. Therefore, the objective of the DANN model becomes:

$$\mathcal{L}_{CE}\big(C(\tilde{x}_s), y_s\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, x_t\big). \tag{5}$$

It is reasonable to expect that Conventional AT cannot fully benefit target domain robustness, as source domain robustness may not perfectly transfer to the target domain due to domain shift. As reported in Table 1, compared to the

Table 1. Results (%) of Conventional AT and our Self-Supervised AT on the VisDA-2017 dataset.

Training method	Clean	FGSM
Natural Training	73.2	21.2
Conventional AT [26]	62.9 (-10.3)	27.1 (+5.9)
Pseudo Labeling	33.1 (-40.1)	27.1 (+5.9)
Self-Supervised AT-L1	56.2 (-17.0)	15.8 (-5.4)
Self-Supervised AT-L2	51.3 (-21.9)	26.0 (+4.8)
Self-Supervised AT-KL	67.1 **(-6.1)**	**35.0 (+13.8)**

Natural Training baseline (*i.e.*, train with only clean data), Conventional AT indeed improves robustness to a certain extent but is not significant. Also, the clean accuracy is largely decreased. Hence, we argue that applying AT directly on the target data is important.

A naive way of applying AT on the target data is to produce pseudo labels y_t' using an external pre-trained UDA model. ASSUDA [40] resorts to this idea and applies it to the UDA semantic segmentation problem. Note that ASSUDA only evaluates black-box robustness. Here we implement the *Pseudo Labeling* idea on image classification and observe its white-box robustness. We use a naturally trained DANN as the pseudo labeler. The objective of Pseudo Labeling approach is as follows:

$$\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{CE}\big(C(\tilde{x}_t), y_t'\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big). \tag{6}$$

In Table 1, we find that Pseudo Labeling's robustness is not better than Conventional AT, and the clean accuracy drops dramatically. We believe that the label noise problem is inevitable in pseudo labels y_t' and limits model performance. This motivates us to explore a new AT method that can be directly performed on the target domain.

4.2 Self-Supervised AT

Inspired by [16,42], we seek to use clean target data's logits $C(x_t)$ as a self-supervision signal to generate adversarial examples $\tilde{x}_t$. Based on the min-max optimization for AT [26], we generate $\tilde{x}_t$ by maximizing the difference between $C(x_t)$ and $C(\tilde{x}_t)$, and minimize that difference to train a model. With this idea, we can generate adversarial examples via self-supervision and perform a kind of AT for the target domain. We call it *Self-Supervised Adversarial Training* or *Self-Supervised AT*. In other words, to generate $\tilde{x}_t$, Self-Supervised AT changes Eq. (4) to:

$$x_t^{j+1} = \mathit{\Pi}_{\|\delta\|_p \le \epsilon}\left(x_t^j + \alpha \cdot sign(\bigtriangledown_{x_t^j}\mathcal{L}(C(x_t^j), C(x_t)))\right), \tag{7}$$

Table 2. Results (%) of SS-AT variants on VisDA-2017. (x_s, x_t) denotes $\mathcal{L}_{DA}(x_s, x_t)$. ●: selected. —: not applicable.

Training method	x_s	$\tilde{x}_s$	x_t	$\tilde{x}_t$	(x_s, x_t)	$(x_s, \tilde{x}_t)$	$(\tilde{x}_s, x_t)$	$(\tilde{x}_s, \tilde{x}_t)$	Clean	FGSM
Natural Training	●		●		●	—	—	—	73.2	21.2
Conventional AT [26]		●	●		—	—	●	—	62.9	27.1
SS-AT-KL	●			●	—	●	—	—	67.1	35.0
SS-AT-s-t-t̃-1	●		●	●	●		—	—	67.3	27.5
SS-AT-s-t-t̃-2	●		●	●	●	●	—	—	73.0	39.4
SS-AT-s-s̃-t-t̃-1	●	●	●	●	●			●	63.4	41.6
SS-AT-s-s̃-t-t̃-2	●	●	●	●		●	●		62.8	42.3
SS-AT-s-s̃-t-t̃-3	●	●	●	●	●	●	●	●	61.3	41.6

and $\tilde{x}_t = x_t^{j_{max}}$. To adversarially train an UDA model, Self-Supervised AT changes Eq. (3) to:

$$\min_{F,C} \mathbb{E}\left[\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}\big(C(\tilde{x}_t), C(x_t)\big)\right]. \tag{8}$$

$\mathcal{L}$ is a loss function that encourages the logits to be similar. Possible choices include L1 loss, L2 loss, Kullback-Leibler (KL) divergence loss, *etc.* Taking KL divergence loss as an example, the objective of Self-Supervised AT for UDA can be written as follows:

$$\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big), \tag{9}$$

where $[\cdot]_{sg}$ denotes the stop-gradient operator [35] constraining its operand to be a non-updated constant. We do not expect that Self-Supervised AT is as robust as conventional supervised AT since the ground-truth labels y are always the strongest supervision. However, given that target domain labels y_t are unavailable, we believe that the clean logits $C(x_t)$ could be a good self-supervision signal.

Table 1 shows that Self-Supervised AT-L1 and Self-Supervised AT-L2 are not effective, while Self-Supervised AT-KL achieves excellent results. Self-Supervised AT-KL increases robust accuracy over Natural Training by 13.8%, which is much better than Conventional AT. It also maintains decent clean accuracy. These results demonstrate that our Self-Supervised AT strategy is effective, but the choice of the loss function is critical, where KL divergence loss is the preferred one.

4.3 On the Effects of Clean and Adversarial Examples in Self-Supervised AT.

Let us revisit the results of the last experiment from another perspective. We observe a trade-off between clean performance and robustness, and the upper part of Table 2 illustrates this point more clearly. Specifically, from Natural

Training and Conventional AT, we can see that replacing clean images x_s by adversarial examples $\tilde{x}_s$ increases robust accuracy but decreases clean accuracy. A similar trade-off can be found between Natural Training and Self-Supervised AT-KL, which train with x_t and $\tilde{x}_t$, respectively. This interests us to further investigate the usage of the four data types $\{x_s, \tilde{x}_s, x_t, \tilde{x}_t\}$ in the AT for UDA. Self-Supervised AT-KL outperforms Conventional AT in terms of both clean and robust accuracies, indicating that using $\tilde{x}_t$ is more efficient than $\tilde{x}_s$, so we start with Self-Supervised AT-KL as a baseline.

First, we add x_t to Self-Supervised AT-KL. This turn out SSAT-s-t-t̃-1 and SSAT-s-t-t̃-2, where SSAT-s-t-t̃-1's domain adaptation loss is $\mathcal{L}_{DA}(x_s, x_t)$, while SSAT-s-t-t̃-2 involves another term and becomes $\mathcal{L}_{DA}(x_s, x_t) + \mathcal{L}_{DA}(x_s, \tilde{x}_t)$. In other words, SSAT-s-t-t̃-1 explicitly transfers the supervised knowledge from x_s to only x_t, while SSAT-s-t-t̃-2 transfers to both x_t and $\tilde{x}_t$. We expect that SSAT-s-t-t̃-1 and SSAT-s-t-t̃-2 enjoy higher clean accuracy than Self-Supervised AT-KL because they involve x_t.

The lower part of Table 2 reports the results. We find that SSAT-s-t-t̃-1's robust accuracy drops significantly, but the clean accuracy does not improve much. In contrast, SSAT-s-t-t̃-2 largely increases both clean and robust accuracies by 5.9% and 4.4%, respectively. The improvement of clean performance matches our expectations, but we are surprised at that of robustness. We see this is due to our Self-Supervised AT's specific property. Self-Supervised AT leverages the objective $\mathcal{L}_{KL}\big(C(\tilde{x}_t), C(x_t)\big)$ to do AT, so $C(x_t)$'s quality is critical. Given that the labels y_t is unavailable, $\mathcal{L}_{DA}(x_s, x_t)$ can transfer the supervised knowledge to x_t and thus enhance $C(x_t)$'s quality. Therefore, adding x_t to Self-Supervised AT benefits robustness as well. This observation is different from the conventional supervised AT that exists the trade-off between performance and robustness [33,39,42]. We conclude that involving x_t into training does help, but an explicit supervised knowledge transfer to $\tilde{x}_t$ is needed. This is rational since $\tilde{x}_t$ plays the most important role in Self-Supervised AT, giving firm guidance to it is essential.

Second, we look into the effects of $\tilde{x}_s$ in Self-Supervised AT. We add $\tilde{x}_s$ and study three variants: SSAT-s-s̃-t-t̃-1, SSAT-s-s̃-t-t̃-2 and SSAT-s-s̃-t-t̃-3. Their differences are in their domain adaptation loss, which is also illustrated in Table 2. Intuitively, we expect that adding $\tilde{x}_s$ falls into the trade-off that leads to lower clean performance but better robustness, as $\tilde{x}_s$ is the conventional supervised adversarial example.

As shown in Table 2, all the three variants obtain lower clean accuracy and higher robust accuracy than SSAT-s-t-t̃-1 and SSAT-s-t-t̃-2, which matches our assumption. The results among these three are very close. Compared to SSAT-s-t-t̃-2, their clean accuracy drops 9.6%-11.7%, but robust accuracy only improves 2.2%-2.9%. This is consistent with Conventional AT's result, *i.e.*, source domain robustness is not easy to transfer to the target domain. Because training without $\tilde{x}_s$ achieves a better trade-off between performance and robustness, we use SSAT-s-t-t̃-2 as a baseline for the next investigation. To present our experiments more clear, in the following, we summarize the objective functions of each Self-Supervised AT variant discussed in this part:

- SSAT-s-t-$\tilde{t}$-1:

$$\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) + \mathcal{L}_{DA}\big(x_s, x_t\big). \tag{10}$$

- SSAT-s-t-$\tilde{t}$-2:

$$\begin{aligned} &\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) \\ &+ \mathcal{L}_{DA}\big(x_s, x_t\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big). \end{aligned} \tag{11}$$

- SSAT-s-$\tilde{s}$-t-$\tilde{t}$-1:

$$\begin{aligned} &\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) \\ &+ \mathcal{L}_{CE}\big(C(\tilde{x}_s), y_s\big) + \mathcal{L}_{DA}\big(x_s, x_t\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, \tilde{x}_t\big). \end{aligned} \tag{12}$$

- SSAT-s-$\tilde{s}$-t-$\tilde{t}$-2:

$$\begin{aligned} &\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) \\ &+ \mathcal{L}_{CE}\big(C(\tilde{x}_s), y_s\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, x_t\big). \end{aligned} \tag{13}$$

- SSAT-s-s-’t-$\tilde{t}$-3:

$$\begin{aligned} &\mathcal{L}_{CE}\big(C(x_s), y_s\big) + \mathcal{L}_{KL}\big(C(\tilde{x}_t), C([x_t]_{sg})\big) + \mathcal{L}_{CE}\big(C(\tilde{x}_s), y_s\big) \\ &+ \mathcal{L}_{DA}\big(x_s, x_t\big) + \mathcal{L}_{DA}\big(x_s, \tilde{x}_t\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, x_t\big) + \mathcal{L}_{DA}\big(\tilde{x}_s, \tilde{x}_t\big). \end{aligned} \tag{14}$$

4.4 On the Effects of BN in Self-Supervised AT

It has been well-known that the statistic estimation of BN [13] plays an important role in both the UDA [4,18] and the adversarial machine learning [19,39,37] fields. It is worth investigating the effects of BN given these two research fields meet together in this paper.

Recall that during training, BN computes the mean and variance of the feature space for each mini-batch, referred to as *batch statistics* [39]. Each mini-batch is normalized by its batch statistics at training time. Hence, the composition of a mini-batch defines its batch statistics, thereby affecting the normalized values of each data point's features. To observe the effects on Self-Supervised AT, we create four variants of SSAT-s-t-$\tilde{t}$-2. They involve the same data types $\{x_s, x_t, \tilde{x}_t\}$ into training but with different mini-batch compositions. Specifically, at each training step, Batch-st-$\tilde{t}$ has two mini-batches, $[x_s,\, x_t]$ and $[\tilde{x}_t]$; Batch-s-t$\tilde{t}$ has two mini-batches, $[x_s]$ and $[x_t,\, \tilde{x}_t]$; Batch-s-t-$\tilde{t}$ has three mini-batches, $[x_s]$, $[x_t]$ and $[\tilde{x}_t]$; and Batch-st$\tilde{t}$ has one mini-batch, $[x_s,\, x_t,\, \tilde{x}_t]$. Batch-st-$\tilde{t}$ is the original SSAT-s-t-$\tilde{t}$-2, which follows the setting of [15]. We expect that their batch statistics differences would cause different results.

Table 3 shows the results. As can be seen, Batch-st-$\tilde{t}$ achieves the highest clean accuracy, while Batch-st$\tilde{t}$ achieves the highest robust accuracy. We argue that in Batch-st$\tilde{t}$, x_s is with the same mini-batch as x_t and $\tilde{x}_t$, so it can also transfer the supervised knowledge through batch statistics. In other words, the batch statistics used to normalize x_t and $\tilde{x}_t$ contain x_s's information. This shares a similar spirit

Table 3. Results (%) of different mini-batch combinations on the VisDA-2017 dataset.

Method	Mini-batches	Clean	FGSM
Batch-st-t̃	$[x_s,\, x_t]$, $[\tilde{x}_t]$	73.0	39.4
Batch-s-tt̃	$[x_s]$, $[x_t,\, \tilde{x}_t]$	68.2	37.0
Batch-s-t-t̃	$[x_s]$, $[x_t]$, $[\tilde{x}_t]$	68.2	35.5
Batch-stt̃	$[x_s,\, x_t,\, \tilde{x}_t]$	69.0	41.4

with the domain adaptation loss $\mathcal{L}_{DA}(x_s, x_t) + \mathcal{L}_{DA}(x_s, \tilde{x}_t)$ discussed in Sec. 4.3, and we have known that it can improve robustness. For Batch-st-t̃, we see its high performance is due to the separation of x_t and $\tilde{x}_t$. Recall that clean and robust features have distinct characteristics [14,33], so putting them into the same mini-batch leads to suboptimal results [39]. Batch-s-t-t̃, however, achieves lower performance than Batch-st-t̃ though it has that separation as well. The reason is that in Batch-st-t̃, x_s and x_t are with the same mini-batch. This encourages the knowledge transfer from x_s to x_t, similar to the spirit of the domain adaptation loss $\mathcal{L}_{DA}(x_s, x_t)$.

Both Batch-st-t̃ and Batch-stt̃ achieve a good trade-off between performance and robustness. We can choose according to the downstream application's focus.

4.5 Summary

In this section, we explore four main aspects of AT for UDA, including Conventional AT, our Self-Supervised AT, the effects of clean and adversarial examples in Self-Supervised AT, and the effects of BN statistics. We progressively derive the best method from each investigation, then we take Batch-stt̃ as our final method, referred to as Adversarially Robust Training for UDA (ARTUDA). ARTUDA's training objective is Eq.(11), and Fig. 1 offers a visualized illustration. Note that some of the other variants also have their advantages, e.g., Batch-st-t̃, so they are still useful for certain focusses.

5 Experiments

We extensively evaluate the proposed ARTUDA on five adversarial attacks, three datasets and three different UDA algorithms. We further compare ARTUDA with the nearly contemporary work, RFA [2]. An analysis of feature space is also presented.

5.1 Experimental Setup

Datasets. We use three UDA datasets for evaluation: VisDA-2017 [29], Office-31 [31] and Office-Home [36]. VisDA-2017 contains two domains: Synthetic and Real.

There are 152,409 Synthetic and 55,400 Real images from 12 object categories in this large-scale dataset. Office-31 has three domains with 31 object categories. These are Amazon (A) with 2,817 images, Webcam (W) with 795 images, and DSLR (D) with 498 images. We employ the D → W task for our expeiment. Office-Home includes four domains with 65 categories: Art (Ar) with 2,427 images, Clipart (Cl) with 4,365, Product (Pr) with 4,439 images, and Real-World (Rw) with 4,375 images. We employ the Ar → Cl task for our experiment.

Attack setting. We test UDA models' adversarial robustness against four white-box attacks, including FGSM [10], PGD [26], MI-FGSM [6] and MultAdv [21], where PGD is the default attack unless stated otherwise. A black-box attack [27] is also considered. For AT, we use the PGD attack with $j_{max} = 3$ and $\epsilon = 3$ of L_∞-norm. If not otherwise specified, we set the same for all the attacks at testing time except that FGSM's j_{max} is 1.

Benchmark UDA algorithms. We apply ARTUDA to three common UDA algorithms, including DANN [8], JAN [24] and CDAN [23]. We use ResNet-50 [12] as a backbone for all of them. If not otherwise specified, DANN is the default UDA algorithm in our experiments.

Baseline defenses. We employ two commonly-used conventional AT algorithms, PGD-AT [26] and TRADES [42], to be our baseline defenses. To the best of our knowledge, RFA [2] might be the only approach aimming at the same problem as ours, and we also compare with it.

Implementation details. Our implementation is based on PyTorch [28]. We adopt Transfer-Learning-library [15] to set up UDA's experimental environment and follow the training hyper-parameters used in [15]. We also use the widely-used library, AdverTorch [5], to perform adversarial attacks. We will release our source code if the paper gets accepted.

5.2 Evaluation Results

White-box robustness. The robustness of multiple training methods against various white-box attacks is reported in Table 4. Without a defense, Natural Training's accuracy drops to almost 0% under the strong iterative attacks. PGD-AT and TRADES improve adversarial robustness though they are originally designed for the traditional classification task. However, they also reduce clean accuracy. The proposed method, ARTUDA, significantly increases robust accuracy. Specifically, on VisDA-2017, it achieves more than 10% and 20% higher robustness than TRADES and PGD-AT, respectively. On Office-31, its robust accuracy is higher than PGD-AT and TRADES by 25%-48% under white-box iterative attacks. On Office-Home, although TRADES is slightly more robust to white-box iterative attacks, ARTUDA has higher accuracy under clean data, FGSM and black-box attacks, leading by a decent margin. In general, ARTUDA is effective across all the five attacks on three datasets. ARTUDA's clean accuracy drops but is still the best among the defenses. It can greatly improve robustness and maintain decent clean performance simultaneously.

Black-box robustness. The robustness against black-box attacks is shown in the last column of Table 4. Here we consider a naturally trained DANN

Table 4. Results (%) of UDA models on multiple datasets under various adversarial attacks.

Dataset	Training method	Clean	FGSM	PGD	MI-FGSM	MultAdv	Black-box
VisDA-2017 [29]	Natural Training	73.2	21.2	0.9	0.5	0.3	58.3
	PGD-AT [26]	60.5	34.6	21.3	22.7	7.8	59.1
	TRADES [42]	64.0	42.1	29.7	31.2	16.4	62.6
	ARTUDA (ours)	65.5	**52.5**	**44.3**	**45.0**	**27.3**	**65.1**
Office-31 D → W[31]	Natural Training	98.0	52.7	0.9	0.6	0.1	95.0
	PGD-AT [26]	95.3	91.8	68.2	66.5	31.4	95.3
	TRADES [42]	88.4	85.3	66.4	67.0	28.2	88.2
	ARTUDA (ours)	96.5	**95.2**	**92.5**	**92.5**	**77.1**	**96.5**
Office-Home Ar → Cl [36]	Natural Training	54.5	26.4	4.7	2.8	2.0	53.1
	PGD-AT [26]	42.5	38.8	36.0	35.8	21.7	43.0
	TRADES [42]	49.3	45.1	**41.6**	**41.6**	**22.5**	49.4
	ARTUDA (ours)	54.0	**49.5**	41.3	39.9	21.6	**53.9**

Table 5. Results (%) of UDA models on the VisDA-2017 dataset under the PGD attack. Three UDA algorithms are considered.

UDA algorithm →		DANN [8]			JAN [24]			CDAN [23]	
Training method ↓	Clean	PGD	Drop	Clean	PGD	Drop	Clean	PGD	Drop
Natural Training	73.2	0.0	-73.2	64.2	0.0	-64.2	75.1	0.0	-75.1
PGD-AT [26]	60.5	13.3	-47.2	47.7	5.8	-41.9	58.2	11.7	-46.5
TRADES [42]	64.0	19.4	-44.6	48.7	8.5	-40.2	64.6	15.7	-48.9
Robust PT [2]	65.8	38.2	-27.6	55.1	32.2	**-22.9**	68.0	41.7	-26.3
RFA [2]	65.3	34.1	-31.2	63.0	32.8	-30.2	72.0	43.5	-28.5
ARTUDA (ours)	65.5	**40.7**	**-24.8**	58.5	**34.4**	-24.1	68.0	**43.6**	**-24.4**

with ResNet-18 as a substitute model and use MI-FGSM, which has better transferability, to generate black-box adversarial examples for target models. In general, the black-box attacks hardly fool the target models. However, we find that the conventional AT approaches have lower black-box accuracy than Natural Training in some cases. This is due to their lower clean accuracy. In contrast, ARTUDA has better clean accuracy and consistently achieves the best black-box robustness across all the datasets.

Generalizability. To compare with the results of [2], in this part, we evaluate robustness against the white-box PGD attack with $j_{max} = 20$ that used in [2]. Table 5 reports the adversarial robustness of multiple popular UDA algorithms. All of them are vulnerable to adversarial attacks. The state-of-the-art approaches, Robust PT and RFA, show excellent effectiveness in improving robustness. We apply our ARTUDA training method to these UDA models to protect them as well. As can be seen, ARTUDA uniformly robustfies all of these models. It consistently achieves low accuracy drops and the highest robust accuracy, which outperforms both Robust PT and RFA. This demonstrates that ARTUDA is generic and can be applied to multiple existing UDA algorithms.

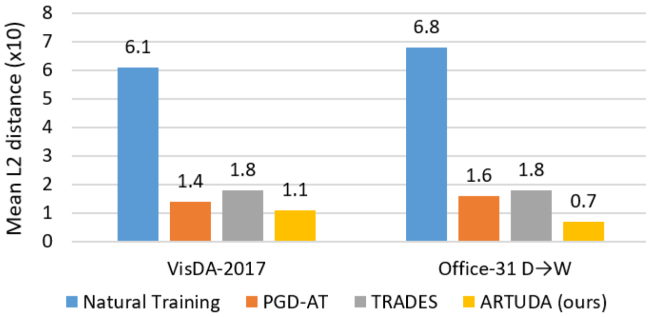
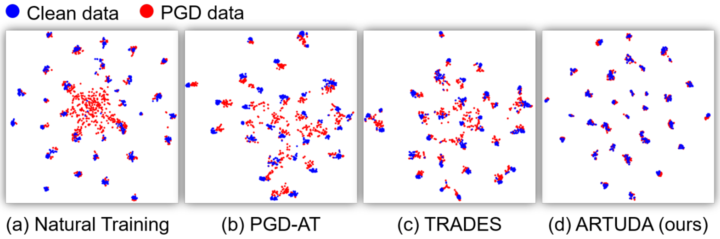


Fig. 2. Mean L_2-norm distance between the feature space of clean images and that of their adversarial examples. The values are the mean over an entire dataset.



(a) Natural Training (b) PGD-AT (c) TRADES (d) ARTUDA (ours)

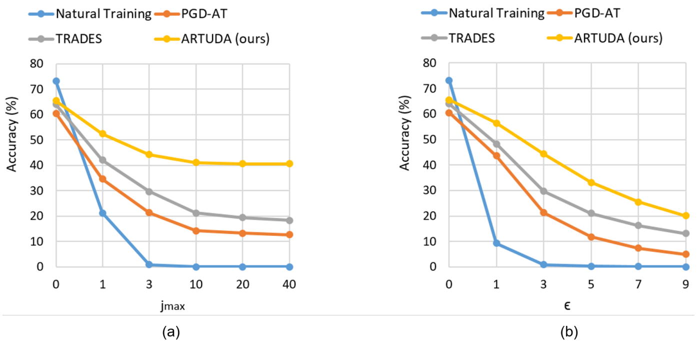
Fig. 3. The t-SNE visualization of the feature space on the Office-31 D→W task.

In terms of clean data accuracy, all the defenses lose clean accuracy to a certain extent. Still, the proposed ARTUDA achieves the best or the second-best clean accuracy among these defenses. Overall, it can significantly improve robustness and maintain decent clean performance simultaneously.

5.3 Analysis

Stability of feature space. Small adversarial perturbations on image space are enlarged considerably in feature space [38]. Hence, the stability of the feature space can reflect a model's robustness [20]. In other words, a robust model's feature space would hardly change under an adversarial example. We compute the mean L_2-norm distance between the feature space of clean images and that of their PGD examples for our models: $\| F(x_t) - F(\tilde{x}_t) \|_2$. The features from the last conv layer of the ResNet-50 backbone are used. As can be seen in Fig. 2, Natual Training has the largest distance, which means that its features are greatly changed when images are adversarially perturbed and thus cause wrong predictions. PGD-AT and TRADES can reduce the distance. ARTUDA attains the smallest distance on both datasets, showing that its feature space is not easily affected by adversarial perturbations.

Visualization of feature space. Fig. 3 visualizes the different methods' feature space on the Office-31 D→W task using t-SNE [25]. The features are from the last



(a)

(b)

Fig. 4. Accuracy of models under PGD attacks (a) with varied numbers of attack iterations j_{max} and (b) with varied perturbation sizes ϵ.

conv layer of the ResNet-50 backbone. The PGD data in the Natural Training model are disorderly scatter and do not align with clean data. PGD-AT and TRADES narrow the distribution gap to a certain extent. ARTUDA impressively align the feature space of PGD and clean data in which they almost overlap with each other. This implies that ARTUDA is effective in learning adversarially robust features. This result is consistent with the above stability analysis.

Attack budgets. We test our ARTUDA's scalability to various attack budgets. We vary the attack budgets by two aspects: the number of attack iterations j_{max} and the perturbation size ϵ. Fig. 4 shows the results. First, we can find that the attack strength does not increase apparently along with the increase of j_{max} when $j_{max} > 3$. This observation is consistent with that of [26]. The proposed ARTUDA demonstrates stable adversarial robustness and consistently performs better than Natural Training, PGD-AT [26] and TRADES [42] under varied j_{max}. On the other hand, the attack strength dramatically increases along with the increase of ϵ. It can be seen that ARTUDA consistently shows better robustness under varied ϵ. Obviously, ARTUDA is scalable to various attack budgets.

6 Conclusion

This paper explores AT methods for the UDA problem. Existing AT approaches require labels to generate adversarial examples and train models, but this does not apply to the unlabeled target domain. We provide a systematicac study into multiple AT variants that may suitable for UDA. This empirical contribution could offer useful insight to the research community. Based on our study, we propose ARTUDA, a novel AT method specifically designed for UDA. Our comprehensive experiments show that ARTUDA improves robustness consistently across multiple attacks and datasets, and outperforms the state-of-the-art methods.

Acknowledgements This work was supported by the DARPA GARD Program HR001119S0026-GARD-FP-052.

References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: International Conference on Machine Learning (2018) 2, 4, 5
2. Awais, M., Zhou, F., Xu, H., Hong, L., Luo, P., Bae, S.H., Li, Z.: Adversarial robustness for unsupervised domain adaptation. In: IEEE International Conference on Computer Vision (2021) 2, 3, 5, 10, 11, 12
3. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases (2013) 1, 4
4. Chang, W.G., You, T., Seo, S., Kwak, S., Han, B.: Domain-specific batch normalization for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 9
5. Ding, G.W., Wang, L., Jin, X.: AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. arXiv preprint arXiv:1902.07623 (2019) 11
6. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 4, 11
7. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning (2015) 1, 3
8. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research (2016) 1, 2, 3, 4, 5, 11, 12
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Conference on Neural Information Processing Systems (2014) 3
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015) 2, 4, 5, 11
11. Guo, C., Rana, M., Cisse, M., Van Der Maaten, L.: Countering adversarial images using input transformations (2018) 2
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE conference on Computer Vision and Pattern Recognition (2016) 1, 2, 5, 11
13. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (2015) 3, 9
14. Itazuri, T., Fukuhara, Y., Kataoka, H., Morishima, S.: What do adversarially robust models look at? arXiv preprint arXiv:1905.07666 (2019) 10
15. Jiang, J., Chen, B., Fu, B., Long, M.: Transfer-learning-library. https://github.com/thuml/Transfer-Learning-Library (2020) 9, 11
16. Kannan, H., Kurakin, A., Goodfellow, I.: Adversarial logit pairing. In: Conference on Neural Information Processing Systems (2018) 2, 4, 6
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Conference on Neural Information Processing Systems (2012) 1
18. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. In: International Conference on Learning Representations Workshop (2017) 9

19. Lo, S.Y., Patel, V.M.: Defending against multiple and unforeseen adversarial videos. IEEE Transactions on Image Processing (2021) 9
20. Lo, S.Y., Patel, V.M.: Error diffusion halftoning against adversarial examples. IEEE International Conference on Image Processing (2021) 2, 13
21. Lo, S.Y., Patel, V.M.: Multav: Multiplicative adversarial videos. IEEE International Conference on Advanced Video and Signal-based Surveillance (2021) 4, 11
22. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning (2015) 1, 3
23. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Conference on Neural Information Processing Systems (2018) 1, 3, 5, 11, 12
24. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: International Conference on Machine Learning (2017) 1, 3, 11, 12
25. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research (2008) 13
26. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018) 2, 3, 4, 5, 6, 7, 11, 12, 14
27. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: ACM Asia Conference on Computer and Communications Security (2017) 11
28. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Conference on Neural Information Processing Systems (2019) 11
29. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924 (2017) 2, 5, 10, 12
30. Raff, E., Sylvester, J., Forsyth, S., McLean, M.: Barrage of random transforms for adversarially robust defense. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 2
31. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: European Conference on Computer Vision (2010) 10, 12
32. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014) 1, 4
33. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: International Conference on Learning Representations (2019) 8, 10
34. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 1, 3, 5
35. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: Conference on Neural Information Processing Systems (2017) 7
36. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition (2017) 10, 12
37. Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A., Le, Q.V.: Adversarial examples improve image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2020) 9

38. Xie, C., Wu, Y., van der Maaten, L., Yuille, A., He, K.: Feature denoising for improving adversarial robustness. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 13
39. Xie, C., Yuille, A.: Intriguing properties of adversarial training at scale. In: International Conference on Learning Representations (2020) 5, 8, 9, 10
40. Yang, J., Li, C., An, W., Ma, H., Guo, Y., Rong, Y., Zhao, P., Huang, J.: Exploring robustness of unsupervised domain adaptation in semantic segmentation. In: IEEE International Conference on Computer Vision (2021) 2, 3, 5, 6
41. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference (2016) 1
42. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning (2019) 2, 4, 5, 6, 8, 11, 12, 14