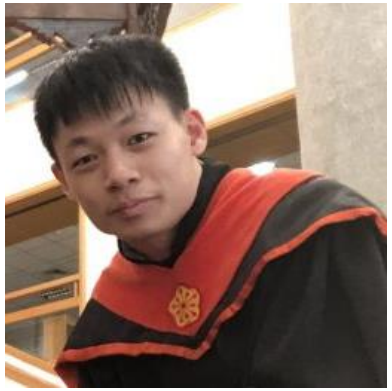
Exploring Adversarially Robust Training for Unsupervised Domain Adaptation

ACCV 2022



Shao-Yuan Lo and Vishal M. Patel

Johns Hopkins University

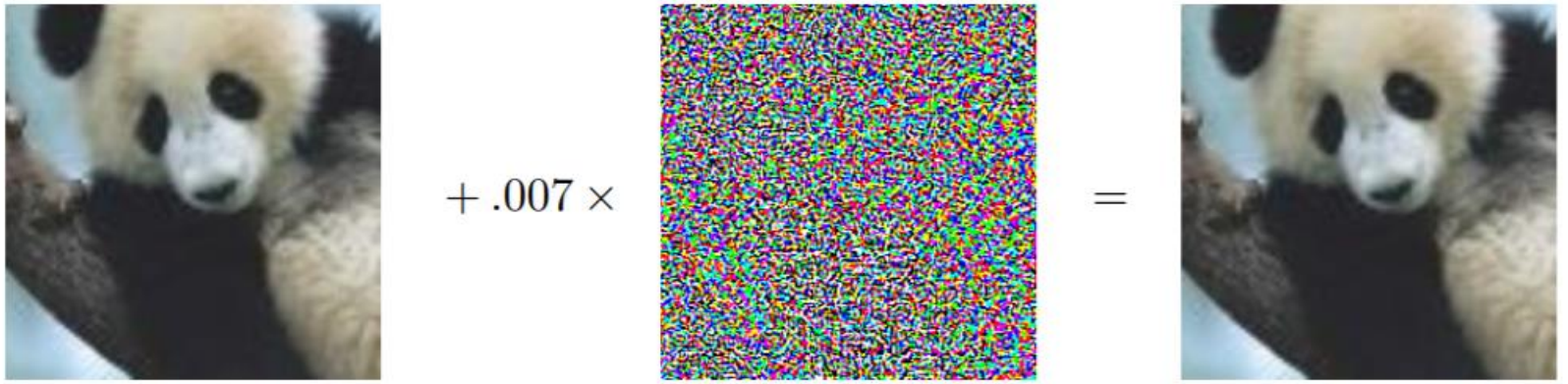
Adversarial Examples

$$x_{adv} = x + \delta$$

$$f(\boldsymbol{x}_{adv}) \neq y$$

Adversarial Examples

- Deep networks are **vulnerable** to adversarial examples.



Goodfellow et al. Explaining and Harnessing Adversarial Examples. ICLR'15.

Adversarial Defenses

- **Image transformation**: Remove perturbations from input images.

$$C(x_{adv}) \neq y.$$

$$C(T(x_{adv})) = y.$$

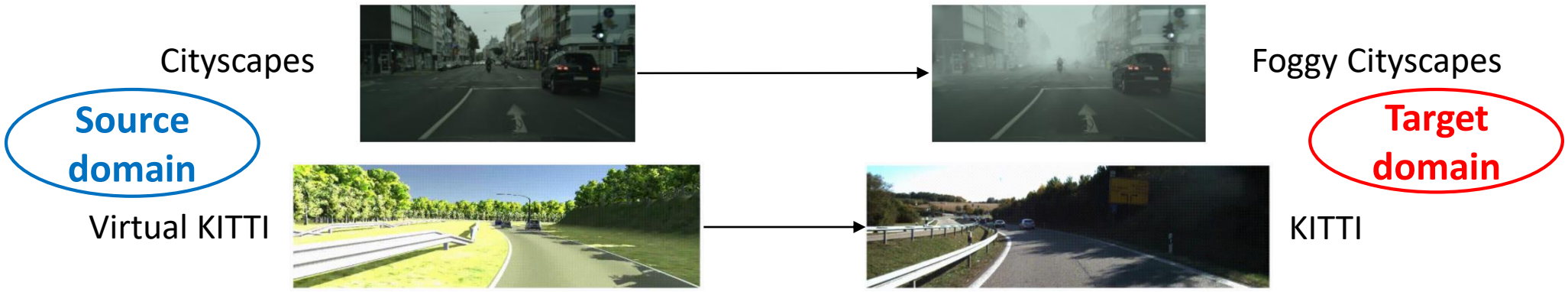
- **Adversarial training (AT)**: Enhance the robustness of networks itself.

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\max_{\delta\in\mathbb{S}} L(x+\delta, y; \theta) \right]$$

Madry et al. Towards deep learning models resistant to adversarial attacks. ICLR'18.

Unsupervised Domain Adaptation (UDA)

- **Scenario**: **Training (source)** data and **test (target)** data are from different domains (i.e. datasets).
  - Cause accuracy drop due to domain shift.
- **Setting**: Given a **labeled source** dataset and an **unlabeled target** dataset, learn a model for the **target** domain.



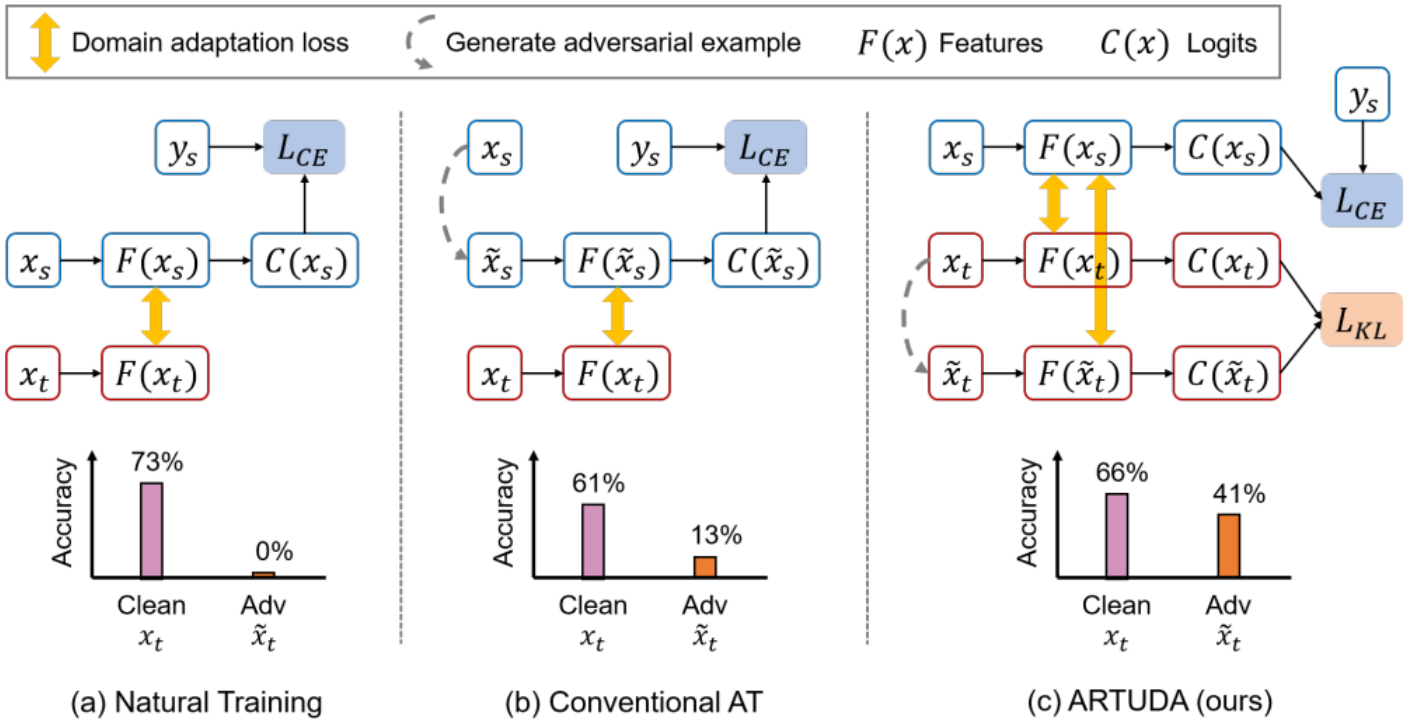
Challenges of AT for UDA

- Conventional AT requires ground-truth labels to generate adversarial examples and train models.

- However, UDA considers the scenario that label information is unavailable to a target domain.

Challenges of AT for UDA

- 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?



Conventional AT on UDA

- Natural Training

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

- Conventional AT on UDA

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

- Pseudo Labeling

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

Self-supervised AT

- **Conventional AT**: Generate adversarial examples **with** ground-truth labels (e.g., L: **cross-entropy loss**)

$$x^{j+1} = \Pi_{\|\delta\|_p \leq \epsilon} \left(x^j + \alpha \cdot sign(\nabla_{x^j} \mathcal{L}(C(x^j), y)) \right)$$

- **Self-supervised AT**: Generate adversarial examples **without** ground-truth labels (e.g., L: L1 loss, L2 loss, **KL divergence loss**)

$$x_t^{j+1} = \Pi_{\|\delta\|_p \leq \epsilon} \left(x_t^j + \alpha \cdot sign(\nabla_{x_t^j} \mathcal{L}(C(x_t^j), C(x_t))) \right)$$

Self-supervised AT

- Conventional AT (PGD-AT)

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

- Self-supervised AT

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

- Self-supervised AT on UDA

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

Self-supervised AT Results

- Dataset: VisDA-2017
- Attacks (white-box): FGSM [Goodfellow et al. 2015]

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)**

On the Effects of Clean and Adversarial Examples in Self-Supervise AT

– SSAT-s-t-$\tilde{t}$-1:

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

– SSAT-s-t-$\tilde{t}$-2:

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

– SSAT-s-$\tilde{s}$-t-$\tilde{t}$-1:

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

– SSAT-s-$\tilde{s}$-t-$\tilde{t}$-2:

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

– SSAT-s-s-'t-$\tilde{t}$-3:

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

On the Effects of Clean and Adversarial Examples in Self-Supervise AT

- Dataset: VisDA-2017
- Attacks (white-box): FGSM [Goodfellow et al. 2015]

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-$\tilde{t}$-1	•		•	•	•		—	—	67.3	27.5
SS-AT-s-t-$\tilde{t}$-2	•		•	•	•	•	—	—	73.0	39.4
SS-AT-s-$\tilde{s}$-t-$\tilde{t}$-1	•	•	•	•	•			•	63.4	41.6
SS-AT-s-$\tilde{s}$-t-$\tilde{t}$-2	•	•	•	•		•	•		62.8	42.3
SS-AT-s-$\tilde{s}$-t-$\tilde{t}$-3	•	•	•	•	•	•	•	•	61.3	41.6

On the Effects of Batch Normalization in Self-Supervise AT

- Dataset: VisDA-2017
- Attacks (white-box): FGSM [Goodfellow et al. 2015]

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

Results

- Comparison with baselines on multiple datasets and 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**

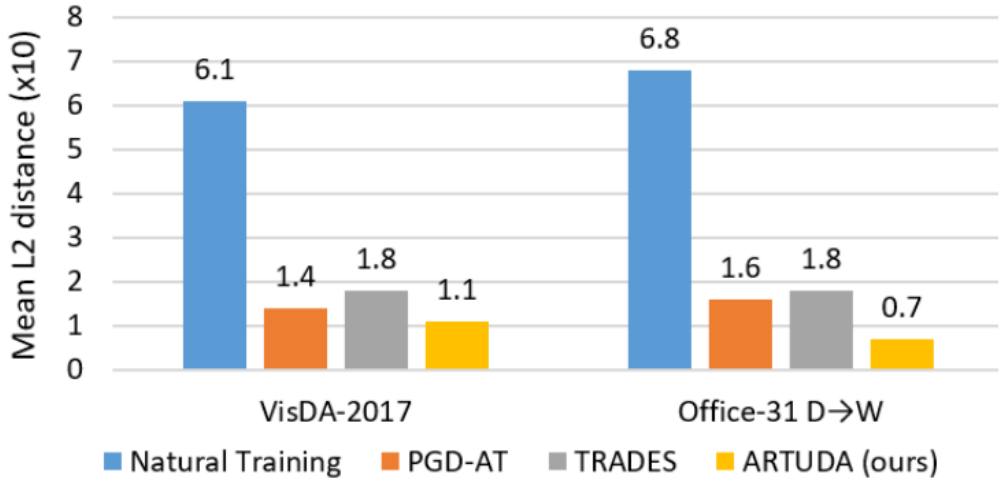
Results

- Comparison with baselines on multiple UDA algorithms

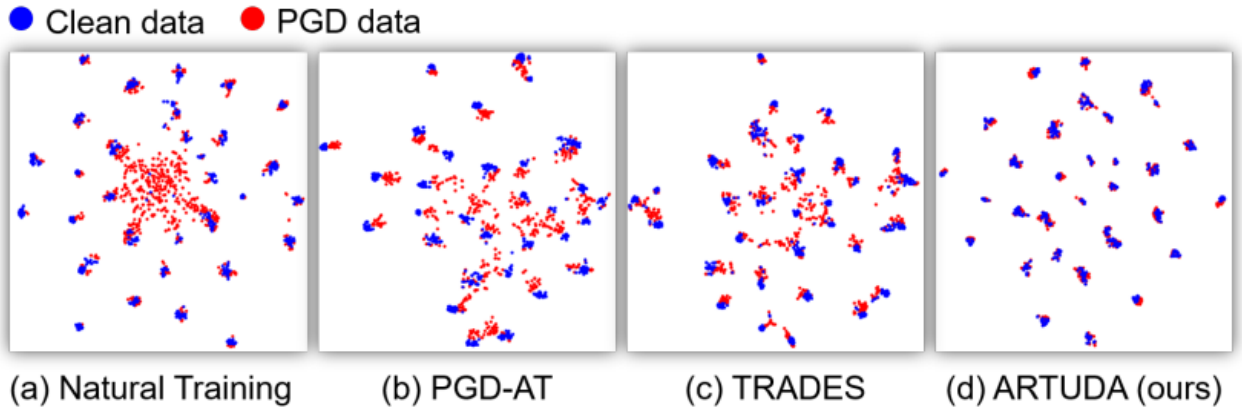
UDA algorithm → / Training method ↓	DANN [8]			JAN [24]			CDAN [23]		
	Clean	PGD	Drop	Clean	PGD	Drop	Clean	PGD	Drop
Natural Training	73.2	0.0	-73.2	64.2	0.0	-64.2	75.1	0.0	-75.1
PGD-AT [26]	60.5	13.3	-47.2	47.7	5.8	-41.9	58.2	11.7	-46.5
TRADES [42]	64.0	19.4	-44.6	48.7	8.5	-40.2	64.6	15.7	-48.9
Robust 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**

Feature Analysis

- Mean square differences between the features of clean images and the features of adversarial examples.



- t-SNE visualization



Conclusion

- We provide a systematic study into various AT methods that are suitable for UDA.

- 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.

- Comprehensive experiments show that ARTUDA consistently improves UDA models' adversarial robustness under multiple attacks and datasets.