Supplementary Material
Adversarial Robustness for Unsupervised Domain Adaptation

Muhammad Awais [1, 2], Fengwei Zhou [1], Hang Xu [1], Lanqing Hong [1],
Ping Luo [3], Sung-Ho Bae [2], Zhenguo Li [1]

[1]Huawei Noah's Ark Lab
[2]Dept. of Computer Science, Kyung-Hee University, South Korea
[3]Dept. of Computer Science, The University of Hong Kong
awais@khu.ac.kr, {zhoufengwei, xu.hang, honglanqing}@huawei.com, pluo@cs.hku.hk,
shbae@khu.ac.kr, li.zhenguo@huawei.com

1. Implementation Details and Extended Results

Experimental Setup: Unless stated otherwise, we use ResNet-50 as the backbone model. We follow [2]'s experimental design. Specifically, we use 1) learning rate of 0.01; 2) a batch size of 64: 32 for source and 32 for target domain; 3) 30 epochs training; 4) 1000 iterations of data per epoch; 5) data augmentation: random horizontal flip for all datasets and center crop for VisDA-2017. We run each experiment 3 times and report the average value of both clean accuracy and robustness. All the experiments are implemented in PyTorch. We also use the same configuration for our proposed method (RFA). The only difference is that we forward passed input batch of data through one frozen teacher model per iteration and get intermediate activation to adapt robust features. We use three datasets for our UDA experiments: VisDA-2017, Office-31, and Office-Home.

Extended Results for Robust Pre-Training: We provide extended results for Section 4 of our main paper. Specifically, we replaced the normally pre-trained ResNet-50 model with adversarially robust ResNet-50 models pre-trained with different perturbation budgets (ϵ) on ImageNet. We conduct experiments with six UDA algorithms on aforementioned datasets. The results averaged over all possible tasks of each dataset are reported in Table 1. The robustness is tested with a PGD-20 attack and perturbation budget of $\epsilon = 3$. The results show that merely replacing the pre-trained model with the robust model can improve the robustness, but it also causes a significant drop in clean accuracy.

Extended Results for Our Method: We also report task-wise results on Office-Home and Office-31 in Figure 2 and Figure 1, respectively. We compare RFA with MDD Baseline (adopting normally pre-trained ImageNet model with MDD algorithm) and Robust PT (adopting adversarially pre-trained ImageNet model with MDD algorithm) with backbone model ResNet-50. It can be seen that our method improves the robustness significantly on all tasks.

References

[1] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2

[2] Mingsheng Long Junguang Jiang, Bo Fu. Transfer-learning-library. https://github.com/thuml/Transfer-Learning-Library, 2020. 1

[3] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 2

[4] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018. 2

[5] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017. 2

[6] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. Bridging theory and algorithm for domain adaptation. *arXiv preprint arXiv:1904.05801*, 2019. 2

Dataset	Robust PT	Source Only	DANN [1]	DAN [3]	CDAN [4]	JAN [5]	MDD [6]
VisDA-2017	$\ell_2(\epsilon = 0)$	43.05 / 0	71.34 / 0	61.79 / 0.01	74.23 / 0	63.70 / 0	72.20 / 4.03
	$\ell_2(\epsilon = 3)$	30.57 / 4.20	66.01 / 36.57	48.54 / 18.40	67.52 / 39.78	56.18 / 29.41	59.79 / 35.38
	$\ell_2(\epsilon = 5)$	31.24 / 5.27	63.95 / 35.50	44.87 / 18.16	67.43 / 41.20	54.14 / 29.58	59.80 / 35.35
	$\ell_\infty(\epsilon = 2)$	37.55 / 3.58	70.30 / 36.56	54.51 / 18.23	71.48 / 38.59	59.13 / 28.55	67.72 / 39.50
	$\ell_\infty(\epsilon = 4)$	34.47 / 4.72	65.79 / 38.21	48.65 / 19.99	68.00 / 41.67	55.08 / 32.15	60.97 / 37.46
	$\ell_\infty(\epsilon = 8)$	25.67 / 6.64	63.45 / 37.44	42.24 / 22.11	65.18 / 41.67	52.00 / 31.87	52.78 / 32.06
Office-31	$\ell_2(\epsilon = 0)$	77.80 / 0.02	85.79 / 0	81.72 / 0	86.90 / 0	85.68 / 0	88.31 / 1.70
	$\ell_2(\epsilon = 3)$	68.61 / 34.05	77.78 / 55.06	73.14 / 33.99	78.93 / 57.49	78.20 / 50.12	80.00 / 61.21
	$\ell_2(\epsilon = 5)$	64.08 / 30.55	73.70 / 55.39	69.34 / 34.05	74.75 / 58.43	73.38 / 49.03	75.72 / 60.87
	$\ell_\infty(\epsilon = 2)$	73.91 / 35.57	81.58 / 58.70	77.81 / 38.09	82.43 / 60.41	81.93 / 52.61	84.05 / 64.62
	$\ell_\infty(\epsilon = 4)$	69.51 / 41.11	77.30 / 62.38	73.71 / 42.29	79.67 / 65.53	78.88 / 57.85	80.72 / 67.54
	$\ell_\infty(\epsilon = 8)$	65.62 / 39.54	74.24 / 61.73	70.61 / 40.40	75.65 / 64.72	75.12 / 60.24	75.73 / 66.46
Office-Home	$\ell_2(\epsilon = 0)$	58.29 / 0.06	63.39 / 0.05	59.64 / 0.23	67.03 / 0.04	64.61 / 0.07	67.91 / 5.81
	$\ell_2(\epsilon = 3)$	51.45 / 24.03	56.82 / 30.39	52.98 / 18.45	61.08 / 35.77	58.84 / 24.92	62.04 / 38.06
	$\ell_2(\epsilon = 5)$	48.85 / 21.32	53.67 / 28.33	50.70 / 17.40	58.10 / 33.34	56.43 / 24.20	59.24 / 36.62
	$\ell_\infty(\epsilon = 2)$	56.02 / 27.74	60.76 / 34.44	57.43 / 21.47	65.08 / 41.15	63.37 / 29.65	65.85 / 42.93
	$\ell_\infty(\epsilon = 4)$	53.89 / 31.46	58.10 / 37.25	55.18 / 24.21	63.04 / 43.81	60.74 / 33.09	63.30 / 43.42
	$\ell_\infty(\epsilon = 8)$	49.87 / 28.89	54.79 / 36.01	51.48 / 23.20	59.10 / 42.80	57.10 / 32.99	59.56 / 42.66

Table 1: Effect of robust pre-training with varying perturbation budget (ϵ) on unsupervised domain adaptation. Reported results are shown as **clean accuracy / adversarial robustness (%)**.

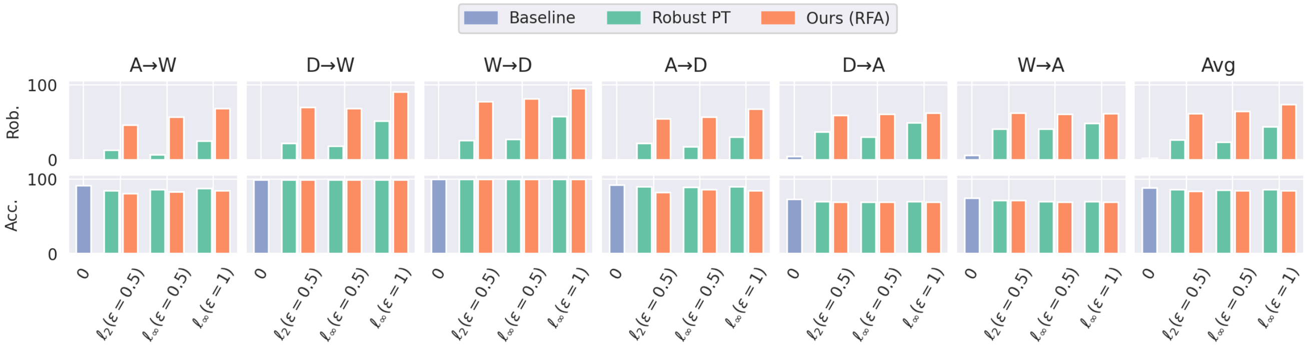

Figure 1: Comparison of robustness and accuracy (%) for MDD Baseline, Robust PT and RFA on six tasks from Office-31 dataset. The x-axis is the perturbation budget of the pre-trained model. RFA consistently improves robustness with a small drop in the clean accuracy.

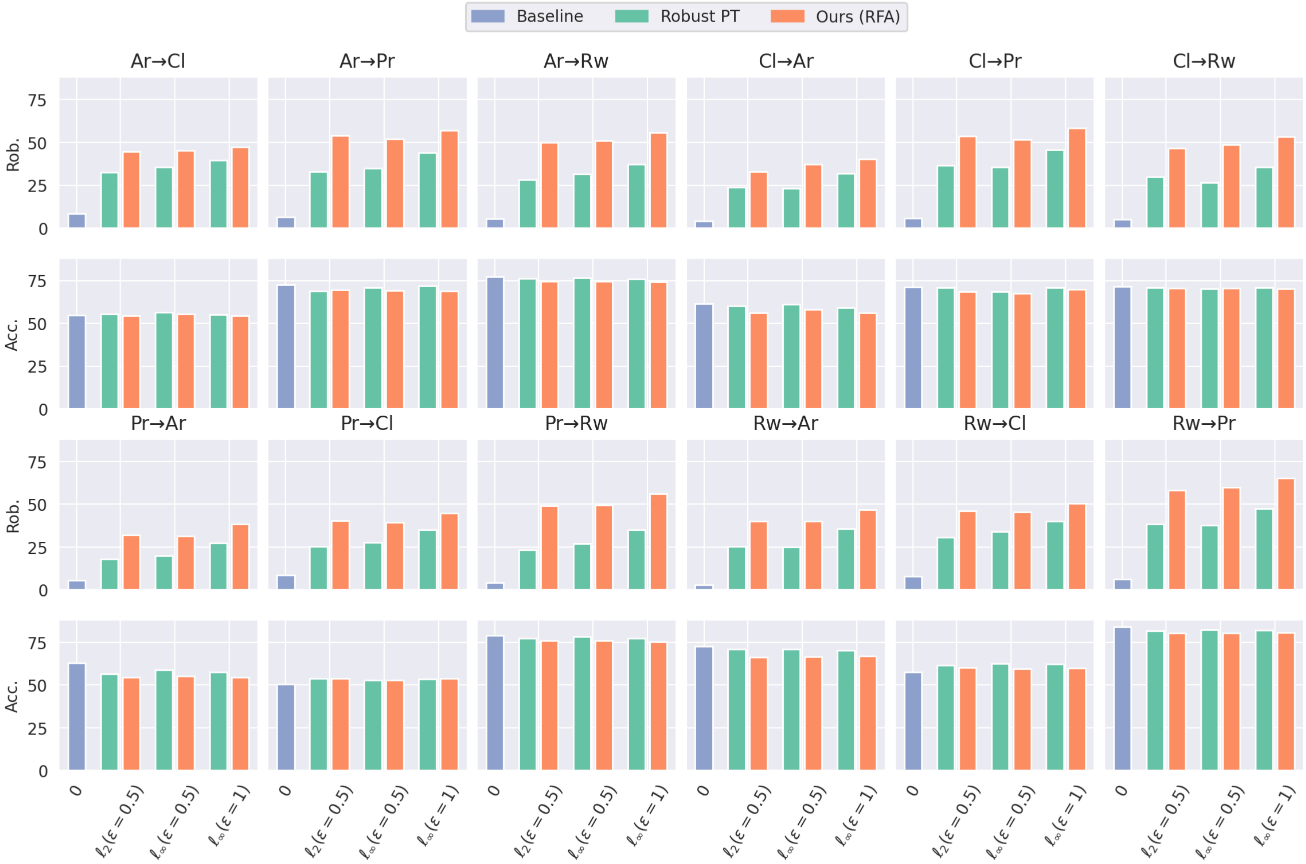

Figure 2: Comparison of robustness and accuracy (%) for MDD Baseline, Robust PT and RFA on twelve tasks from Office-Home dataset. The x-axis is the perturbation budget of the pre-trained model. RFA consistently improves robustness with a small drop in the clean accuracy.