Bilateral Adversarial Training: Towards Fast Training of More Robust Models Against Adversarial Attacks

Jianyu Wang
Baidu Research USA
wjyouch@gmail.com

Haichao Zhang
Baidu Research USA
hczhang1@gmail.com

ICCV’19 paper review

202111278 컴퓨터공학부 김환희

목차

- Introduction
- 3 Motivation
- 4 Formulation
- 5 Experiments

Introduction

Abstract

- We propose to perturb both the image and the label during training

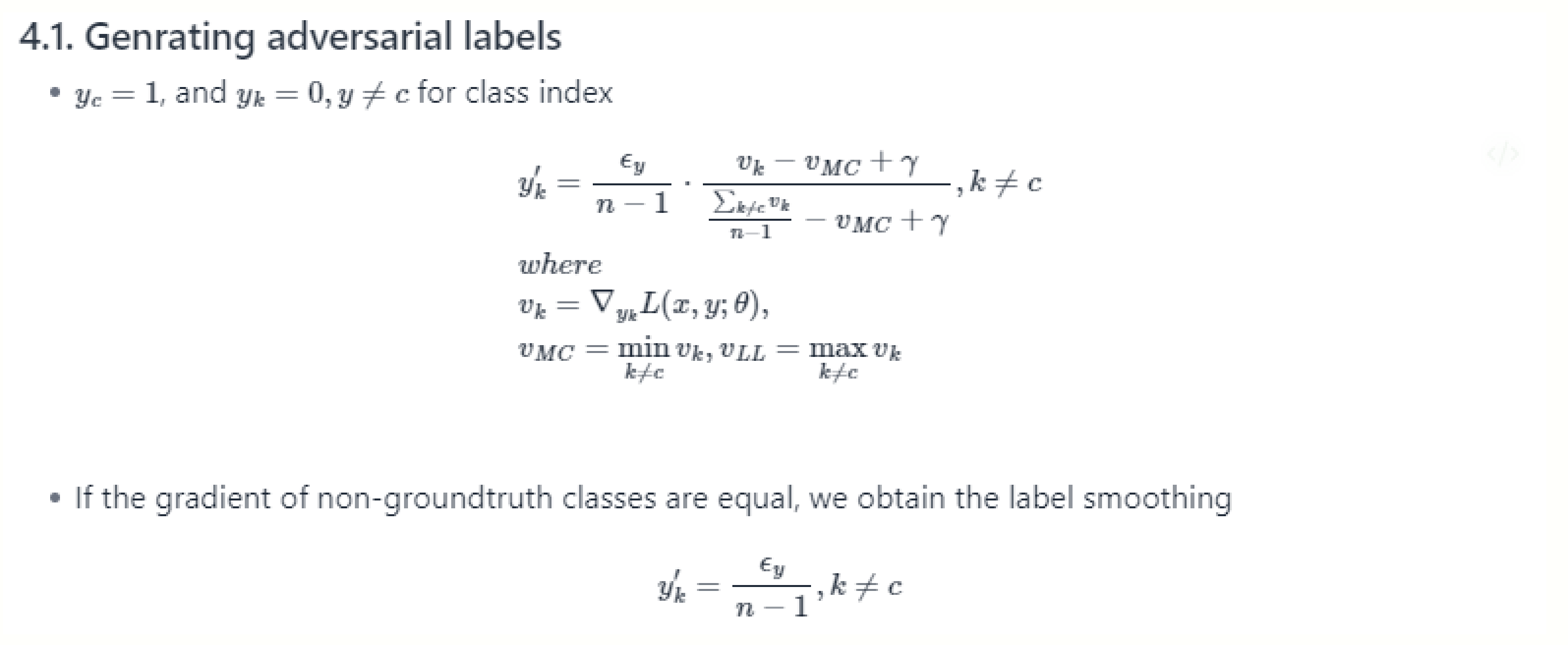
Introduction

- We propose a formulation to achieve two conditions: (1)low loss, (2)small gradient magnitude by perturbing both input images and labels during training
- We uses targeted attack to most confusing class
- We add random uniform noise to the original image
- We derive a forumla to perturb the groundtruth label

Motivation

- For adversarial robustness problem, we performed two experiments
- First experiment: PGD2-8(2 iterations and 8 step sizes, weaker attack) is largely as robust as PGD7-2(7 iterations and 2 step sizes, stronger attack)
  → robustness may not be achieved by simply fitting sufficient adversarial examples. There is more essential ingredients that directly relate to network robustness.
- Second experiment: gradient magnitude of adversarially trained models is much smaller than that of undefended models
  → gradient-based adversarial attacks fail with small gradient

Formulation



4.1. Genrating adversarial labels

- $y_c = 1$, and $y_k = 0, y \neq c$ for class index

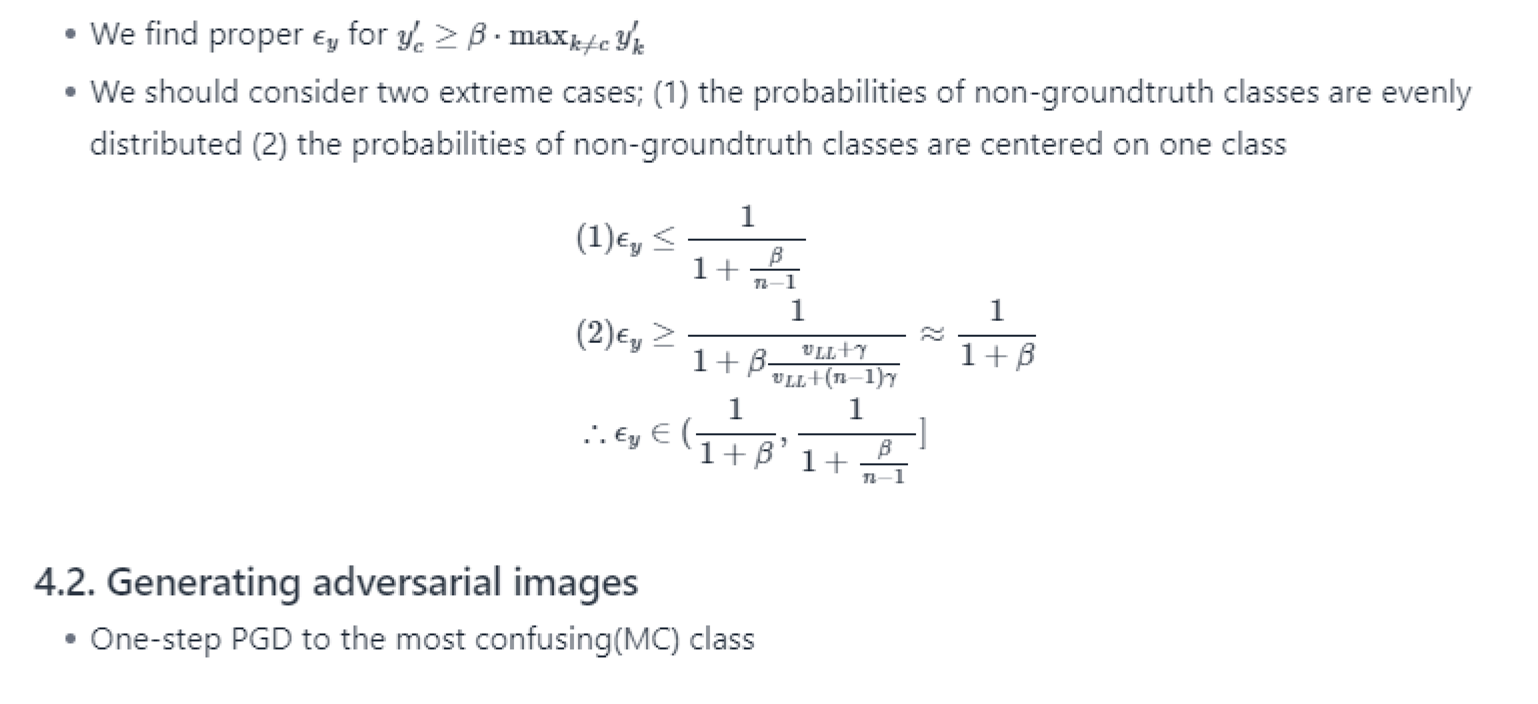
$$y'_k = \frac{\epsilon_y}{n-1} \cdot \frac{v_k - v_{MC} + \gamma}{\frac{\sum_{k \neq c} v_k}{n-1} - v_{MC} + \gamma}, k \neq c$$

$$\begin{aligned}
&where \\
&v_k = \nabla_{y_k} L(x, y; \theta), \\
&v_{MC} = \min_{k \neq c} v_k, v_{LL} = \max_{k \neq c} v_k
\end{aligned}$$

- If the gradient of non-groundtruth classes are equal, we obtain the label smoothing

$$y'_k = \frac{\epsilon_y}{n-1}, k \neq c$$

Formulation



- We find proper ϵ_y for $y'_c \geq \beta \cdot \max_{k \neq c} y'_k$
- We should consider two extreme cases; (1) the probabilities of non-groundtruth classes are evenly distributed (2) the probabilities of non-groundtruth classes are centered on one class

$$(1)\epsilon_y \leq \frac{1}{1+\frac{\beta}{n-1}}$$

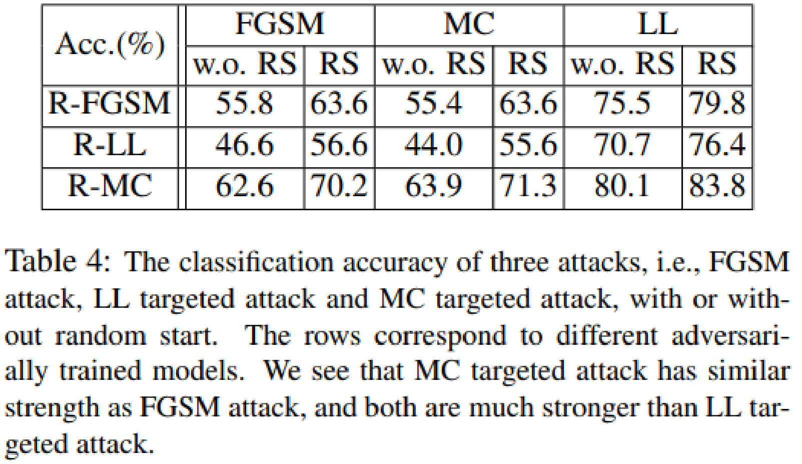
$$(2)\epsilon_y \geq \frac{1}{1+\beta\frac{v_{LL}+\gamma}{v_{LL}+(n-1)\gamma}} \approx \frac{1}{1+\beta}$$

$$\therefore \epsilon_y \in (\frac{1}{1+\beta}, \frac{1}{1+\frac{\beta}{n-1}}]$$

4.2. Generating adversarial images

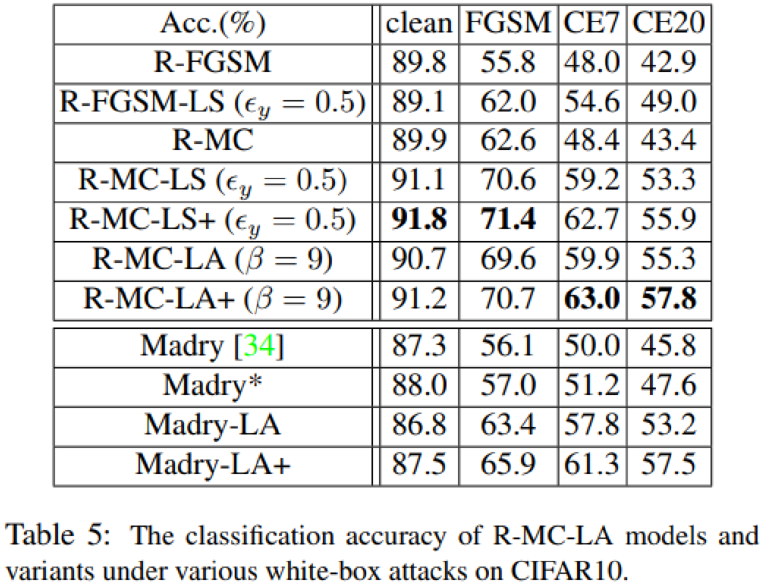
- One-step PGD to the most confusing(MC) class

Experiments



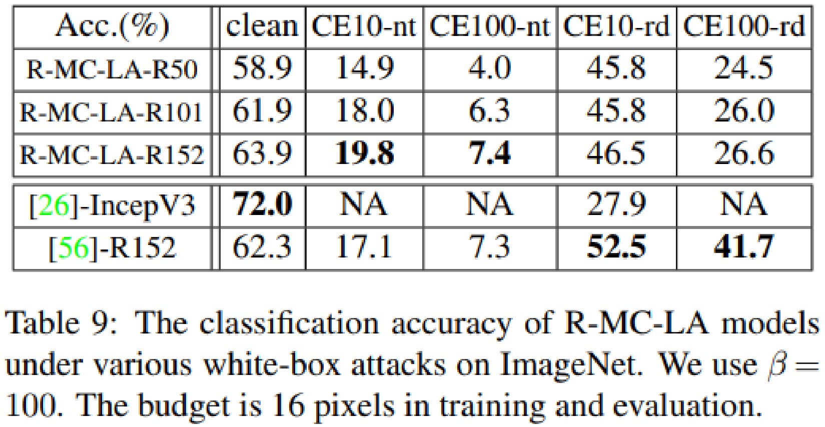
Acc.(%)	FGSM		MC		LL	
	w.o. RS	RS	w.o. RS	RS	w.o. RS	RS
R-FGSM	55.8	63.6	55.4	63.6	75.5	79.8
R-LL	46.6	56.6	44.0	55.6	70.7	76.4
R-MC	62.6	70.2	63.9	71.3	80.1	83.8

Table 4: The classification accuracy of three attacks, i.e., FGSM attack, LL targeted attack and MC targeted attack, with or without random start. The rows correspond to different adversarially trained models. We see that MC targeted attack has similar strength as FGSM attack, and both are much stronger than LL targeted attack.



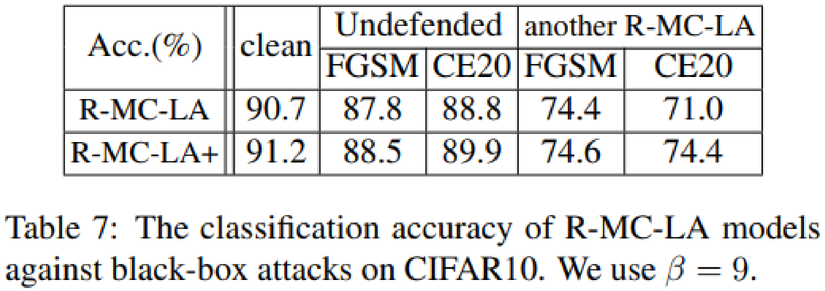
Acc.(%)	clean	FGSM	CE7	CE20
R-FGSM	89.8	55.8	48.0	42.9
R-FGSM-LS ($\epsilon_y = 0.5$)	89.1	62.0	54.6	49.0
R-MC	89.9	62.6	48.4	43.4
R-MC-LS ($\epsilon_y = 0.5$)	91.1	70.6	59.2	53.3
R-MC-LS+ ($\epsilon_y = 0.5$)	**91.8**	**71.4**	62.7	55.9
R-MC-LA ($\beta = 9$)	90.7	69.6	59.9	55.3
R-MC-LA+ ($\beta = 9$)	91.2	70.7	**63.0**	**57.8**
Madry [34]	87.3	56.1	50.0	45.8
Madry*	88.0	57.0	51.2	47.6
Madry-LA	86.8	63.4	57.8	53.2
Madry-LA+	87.5	65.9	61.3	57.5

Table 5: The classification accuracy of R-MC-LA models and variants under various white-box attacks on CIFAR10.



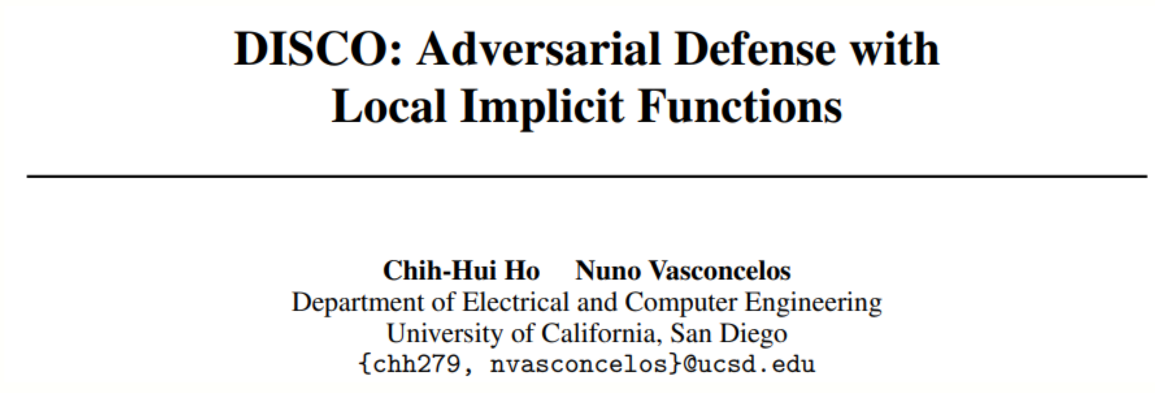
Acc.(%)	clean	CE10-nt	CE100-nt	CE10-rd	CE100-rd
R-MC-LA-R50	58.9	14.9	4.0	45.8	24.5
R-MC-LA-R101	61.9	18.0	6.3	45.8	26.0
R-MC-LA-R152	63.9	**19.8**	**7.4**	46.5	26.6
[26]-IncepV3	**72.0**	NA	NA	27.9	NA
[56]-R152	62.3	17.1	7.3	**52.5**	**41.7**

Table 9: The classification accuracy of R-MC-LA models under various white-box attacks on ImageNet. We use $\beta = 100$. The budget is 16 pixels in training and evaluation.



Acc.(%)	clean	Undefended		another R-MC-LA	
		FGSM	CE20	FGSM	CE20
R-MC-LA	90.7	87.8	88.8	74.4	71.0
R-MC-LA+	91.2	88.5	89.9	74.6	74.4

Table 7: The classification accuracy of R-MC-LA models against black-box attacks on CIFAR10. We use $\beta = 9$.



DISCO: Adversarial Defense with Local Implicit Functions

Chih-Hui Ho **Nuno Vasconcelos**
Department of Electrical and Computer Engineering
University of California, San Diego
{chh279, nvasconcelos}@ucsd.edu

NeurIPS'22 paper review

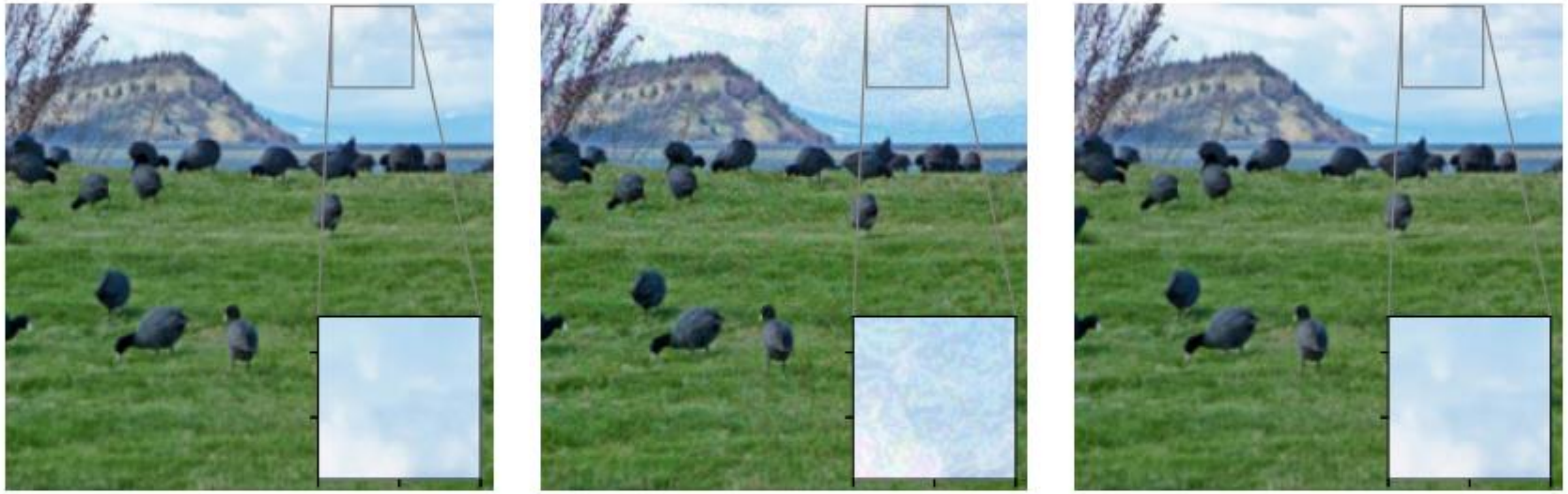
202111278 컴퓨터공학부 김환희

목차

- Abstract
- 3 Method
- 4 Experiments
- 5 Discussion, societal impact and limitations

Introduction

- *aDversarIal defenSe with local impliCit functiOns*
- We propose to remove adversarial perturbations by localized manifold projections
- It outperforms prior defenses, regardless of whether the defense is know to the attacker



(a) Clean Image (b) Adversarial Image (c) DISCO Output

Figure 1: Qualitative performance of DISCO output of a randomly selected ImageNet [23] image.

Method

3.1 Motivation

- DNN을 이용한 implicit function이 realistic image나 3d shape의 분포를 기억하고 합성할 수 있다.
- DISCO는 합성된 이미지가 manifold thickening을 위해 사용될 수 있다고 말한다.

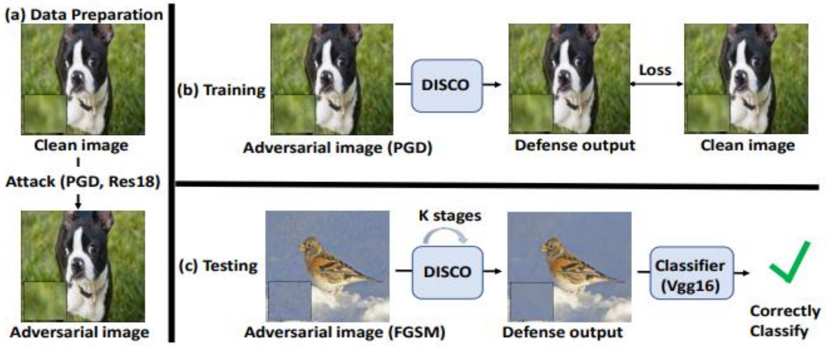


Figure 3: (a) Data preparation, (b) training and (c) testing phase of DISCO. DISCO supports different configurations of attack and classifier for training and testing. For cascade DISCO, $K > 1$.

Method

3.2 Model architecture and training

- (H * W * 3) 크기의 adversarial image가 들어오면 encoder E가 (H * W * C) 크기의 feature map을 만든다. 이후 MLP implicit module L이 특정 위치 좌표를 input으로 받으면 feature map의 특정 위치 좌표 주변의 feature map을 추출하여 해당 좌표의 픽셀을 복원한다. 이들을 합성해 (H' * W' * 3) 크기의 복원된 이미지가 만들어진다.

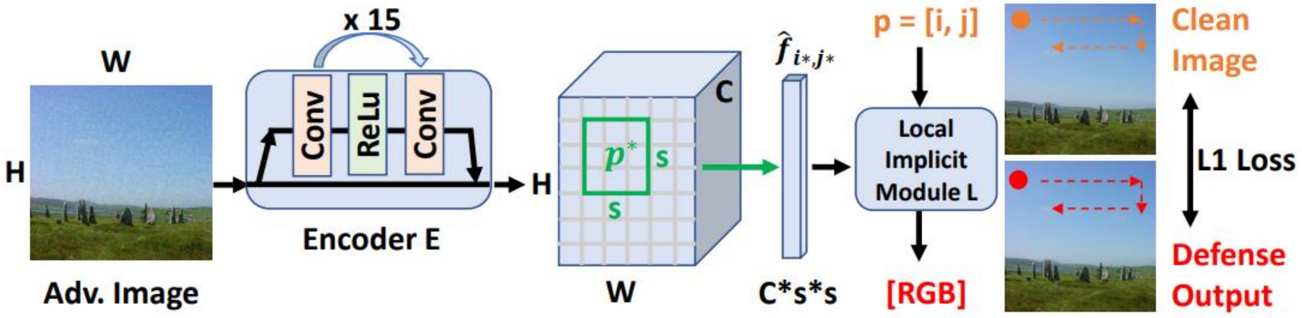


Figure 5: The DISCO architecture includes an encoder and a local implicit module. The network is trained to map Adversarial into Defense images, using an L_1 loss to Clean images.

Method

3.3 Inference

- Classifier, attack, dataset의 종류가 달라져도 유연하게 성능을 내는 모습을 보였다.

3.4 DISCO cascades

- 모든 dataset에 대해 image를 만드는 adversarial training보다 효율적이다.
- Classifier에 의존하지 않고 학습을 진행할 수 있어 효율적이다.

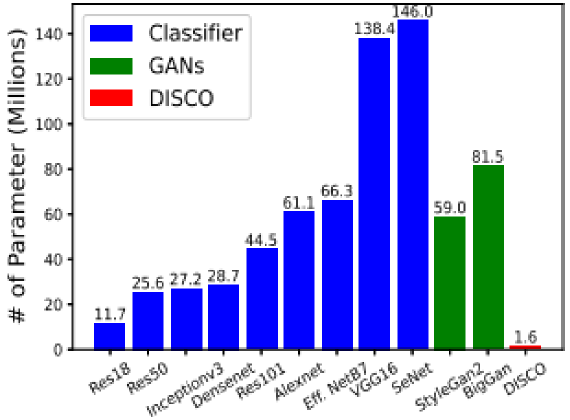


Figure 4: Number of parameters (Millions) of recent classifiers, GANs and DISCO.

Experiments

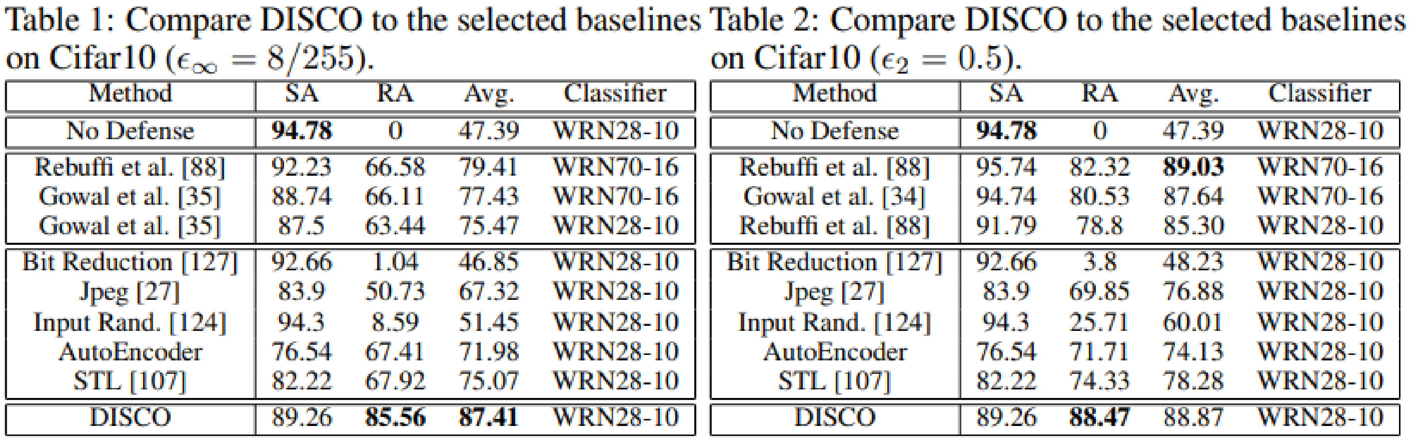


Table 1: Compare DISCO to the selected baselines on Cifar10 ($\epsilon_\infty = 8/255$).

Method	SA	RA	Avg.	Classifier
No Defense	**94.78**	0	47.39	WRN28-10
Rebuffi et al. [88]	92.23	66.58	79.41	WRN70-16
Gowal et al. [35]	88.74	66.11	77.43	WRN70-16
Gowal et al. [35]	87.5	63.44	75.47	WRN28-10
Bit Reduction [127]	92.66	1.04	46.85	WRN28-10
Jpeg [27]	83.9	50.73	67.32	WRN28-10
Input Rand. [124]	94.3	8.59	51.45	WRN28-10
AutoEncoder	76.54	67.41	71.98	WRN28-10
STL [107]	82.22	67.92	75.07	WRN28-10
DISCO	89.26	**85.56**	**87.41**	WRN28-10

Table 2: Compare DISCO to the selected baselines on Cifar10 ($\epsilon_2 = 0.5$).

Method	SA	RA	Avg.	Classifier
No Defense	**94.78**	0	47.39	WRN28-10
Rebuffi et al. [88]	95.74	82.32	**89.03**	WRN70-16
Gowal et al. [34]	94.74	80.53	87.64	WRN70-16
Rebuffi et al. [88]	91.79	78.8	85.30	WRN28-10
Bit Reduction [127]	92.66	3.8	48.23	WRN28-10
Jpeg [27]	83.9	69.85	76.88	WRN28-10
Input Rand. [124]	94.3	25.71	60.01	WRN28-10
AutoEncoder	76.54	71.71	74.13	WRN28-10
STL [107]	82.22	74.33	78.28	WRN28-10
DISCO	89.26	**88.47**	88.87	WRN28-10

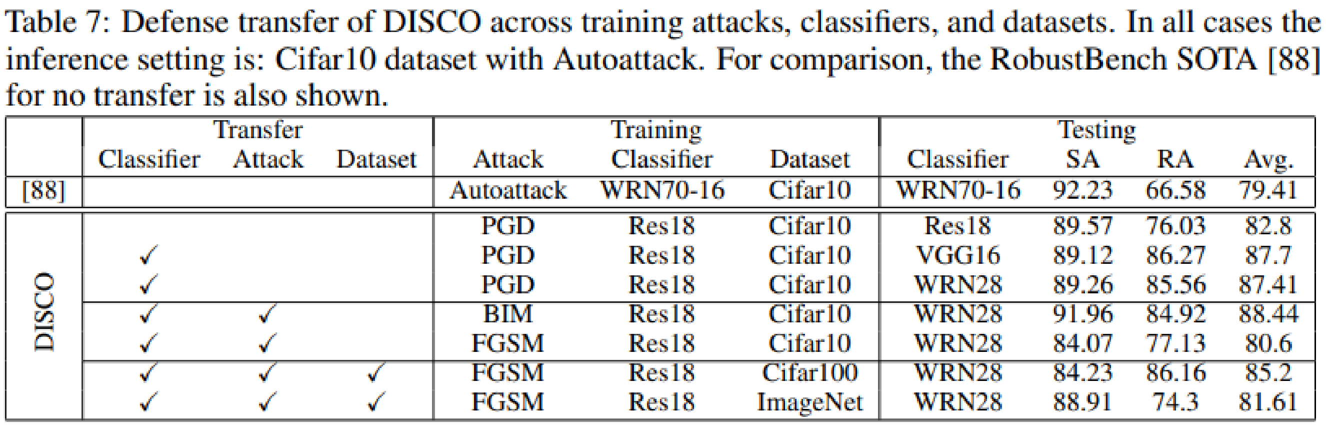


Table 7: Defense transfer of DISCO across training attacks, classifiers, and datasets. In all cases the inference setting is: Cifar10 dataset with Autoattack. For comparison, the RobustBench SOTA [88] for no transfer is also shown.

	Transfer			Training			Testing			
	Classifier	Attack	Dataset	Attack	Classifier	Dataset	Classifier	SA	RA	Avg.
[88]				Autoattack	WRN70-16	Cifar10	WRN70-16	92.23	66.58	79.41
DISCO				PGD	Res18	Cifar10	Res18	89.57	76.03	82.8
	✓			PGD	Res18	Cifar10	VGG16	89.12	86.27	87.7
	✓			PGD	Res18	Cifar10	WRN28	89.26	85.56	87.41
	✓	✓		BIM	Res18	Cifar10	WRN28	91.96	84.92	88.44
	✓	✓		FGSM	Res18	Cifar10	WRN28	84.07	77.13	80.6
	✓	✓	✓	FGSM	Res18	Cifar100	WRN28	84.23	86.16	85.2
	✓	✓	✓	FGSM	Res18	ImageNet	WRN28	88.91	74.3	81.61

Discussion, societal impact and limitations

- Limitations: 더 다양한 attack, dataset, 방법론에 대해서 실험이 필요하다.
- Societal impact: local implicit function을 사용하는 것이 의의가 있다.