

# Adversarial Defense

Lecturer: Dr. Xingjun Ma

School of Computer Science, Fudan University

Autumn, 2022

# Recap: week 4

## 1. Adversarial Example Detection

- ❑ Secondary Classification Methods (二级分类法)
- ❑ Principle Component Analysis (主成分分析法, PCA)
- ❑ Distribution Detection Methods (分布检测法)
- ❑ Prediction Inconsistency (预测不一致性)
- ❑ Reconstruction Inconsistency (重建不一致性)
- ❑ Trapping Based Detection (诱捕检测法)



# Skip: week 5

- ❑ We will skip week 5 for the national holiday (10.1)
- ❑ We will replace week 6's content with week 5's content

5	To appear	对抗防御（中）：介绍早期的对抗样本防御方法以及它们的不可靠性，介绍对抗训练方法，使学生了解对抗训练的理论基础、训练方法和技巧，以及尚存在的挑战。	课堂对抗攻击赛（ <a href="#">click here to participate</a> ）
6	To appear	<del>对抗防御（下）：介绍解理论对抗防御工作，使学生了解可证明的对抗防御方法。</del>	<del>课堂对抗攻击赛（<a href="#">click here to participate</a>）</del>



# Skip: week 6

## □ Adversarial Defense

5	To appear	对抗防御（中）：介绍早期的对抗样本防御方法以及它们的不可靠性，介绍对抗训练方法，使学生了解对抗训练的理论基础、训练方法和技巧，以及尚存在的挑战。	课堂对抗攻击赛（ <a href="#">click here to participate</a> ）
6	To appear	<del>对抗防御（下）：介绍解理论对抗防御工作，使学生了解可证明的对抗防御方法。</del>	<del>课堂对抗攻击赛（<a href="#">click here to participate</a>）</del>



# Adversarial Attack Competition

RESULTS							
#	User	Entries	Date of Last Entry	Score ▲	Error Rate ▲	Efficiency Score ▲	Detailed Results
1	songtianwei	3	10/11/22	0.5051 (1)	0.5010 (3)	0.4067 (8)	<a href="#">View</a>
2	miaojie	9	10/08/22	0.5047 (2)	0.5030 (2)	0.1662 (10)	<a href="#">View</a>
3	liuhuan	9	10/06/22	0.5046 (3)	0.5030 (1)	0.1563 (11)	<a href="#">View</a>
4	AliceNCsyuk	10	10/11/22	0.4968 (4)	0.4920 (4)	0.4770 (7)	<a href="#">View</a>
5	Kasia2222	16	10/09/22	0.4954 (5)	0.4890 (6)	0.6381 (4)	<a href="#">View</a>
6	Yuxuan_Wang	10	10/09/22	0.4938 (6)	0.4900 (5)	0.3762 (9)	<a href="#">View</a>
7	hsj576	3	10/10/22	0.4900 (7)	0.4850 (8)	0.5050 (6)	<a href="#">View</a>
8	terrytengli	11	10/07/22	0.4879 (8)	0.4790 (9)	0.8911 (1)	<a href="#">View</a>
9	xinwang22	29	10/10/22	0.4867 (9)	0.4780 (10)	0.8713 (2)	<a href="#">View</a>
10	Shadow_H	8	10/10/22	0.4861 (10)	0.4860 (7)	0.0099 (12)	<a href="#">View</a>
11	strawberryXia	16	10/10/22	0.4830 (11)	0.4750 (12)	0.8020 (3)	<a href="#">View</a>
11	yong_xie	10	10/08/22	0.4830 (11)	0.4750 (12)	0.8020 (3)	<a href="#">View</a>
12	WeijianMa	7	10/05/22	0.4811 (12)	0.4760 (11)	0.5093 (5)	<a href="#">View</a>
13	BlaBlaWang	1	09/30/22	0.4449 (13)	0.4360 (13)	0.8911 (1)	<a href="#">View</a>

**Link:** [https://codalab.lisn.upsaclay.fr/competitions/7556?secret\\_key=d4a3b1fa-66e2-4a80-8ce6-b5f99e518979](https://codalab.lisn.upsaclay.fr/competitions/7556?secret_key=d4a3b1fa-66e2-4a80-8ce6-b5f99e518979)

**Starting kit:** [https://codalab.lisn.upsaclay.fr/competitions/7556?secret\\_key=d4a3b1fa-66e2-4a80-8ce6-b5f99e518979#learn\\_the\\_details-get\\_starting\\_kit](https://codalab.lisn.upsaclay.fr/competitions/7556?secret_key=d4a3b1fa-66e2-4a80-8ce6-b5f99e518979#learn_the_details-get_starting_kit)



# Adversarial Defense vs Detection

- ❑ The weird relationship between defense and detection
  - ✓ Detection **IS** defense
  - ✓ But... when we say **defense**, we (most of the time) mean **the model is secured**, yet detection cannot do that...
  - ✓ In survey papers: detection is defense
  - ✓ In technical papers: defense is defense not detection
- ❑ Differences
  - ✓ Defense is to secure the model or the system
  - ✓ Detection is to identify potential threats, which should be followed by a defense strategy, e.g., query rejection (but mostly ignored)
  - ✓ By defense, it mostly means robust training methods



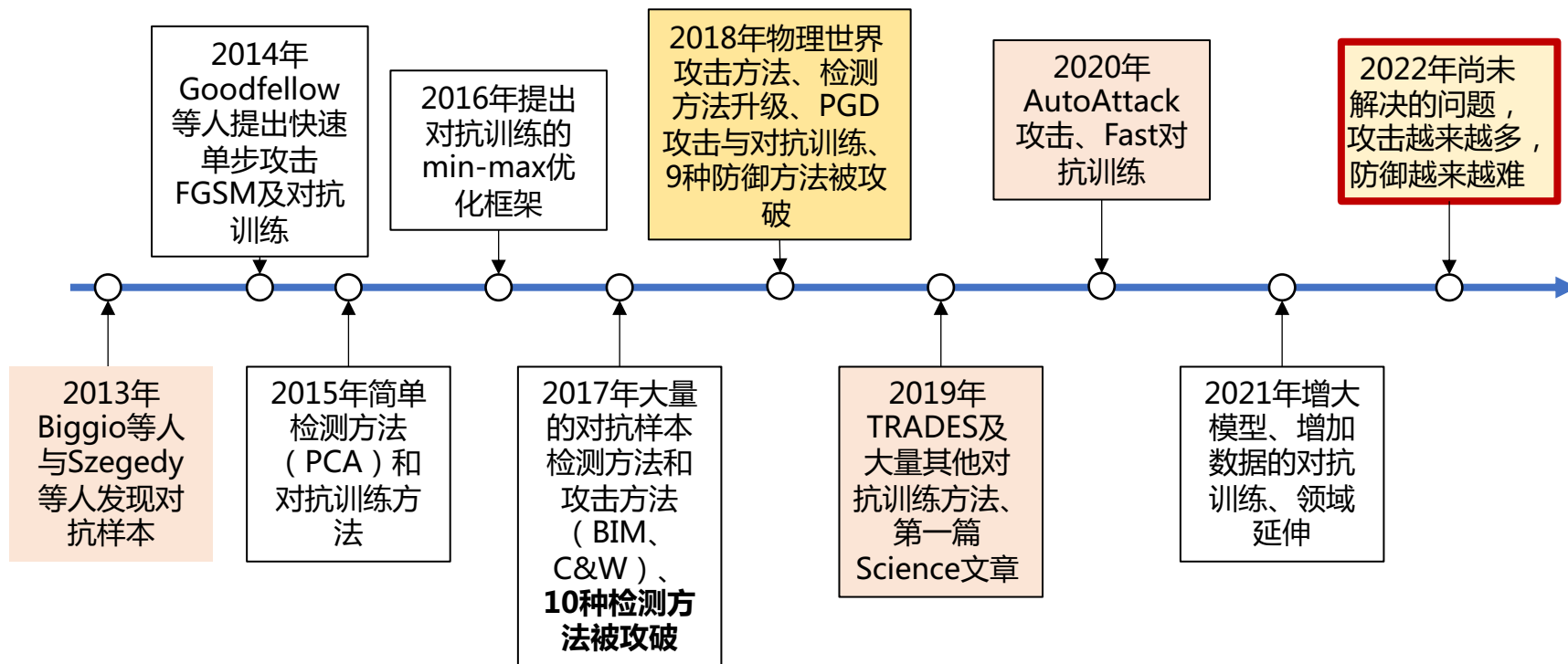
# Defense Methods

---

- ❑ Early Defense Methods
- ❑ Early Adversarial Training Methods
- ❑ Advanced Adversarial Training Methods
- ❑ Remaining Challenges and Recent Progress



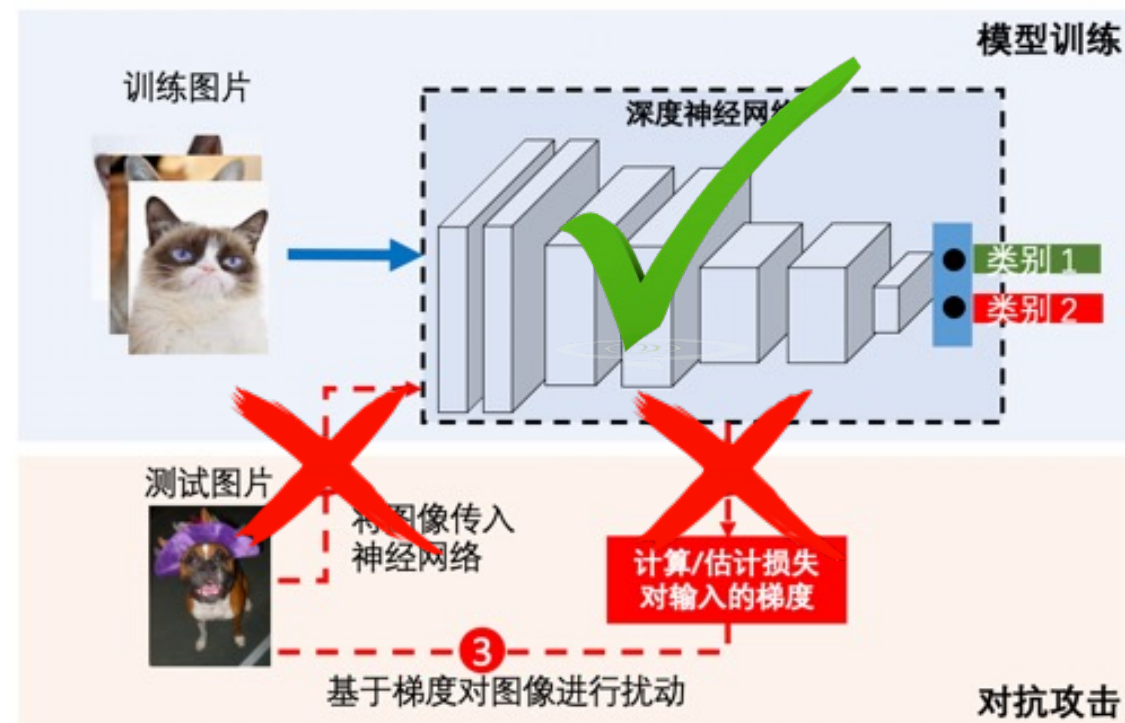
# A Recap of the Timeline





# Principles of Defense

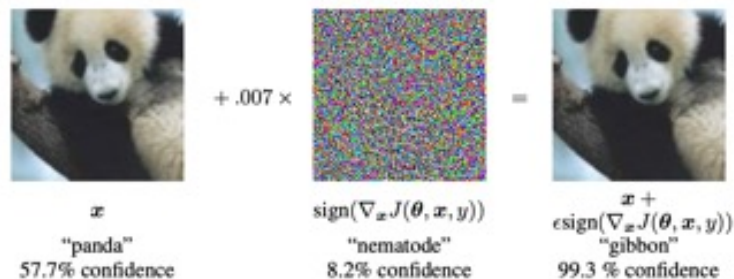
- ❑ Block the attack (卡住出入口)
  - Mask the input gradients
  - Regularize the input gradients
  - Distill the logits
  - Denoise the input
- ❑ Robustify the model (增强中间)
  - Smooth the decision boundary
  - Reduce the Lipschitzness of the model
  - Smooth the loss landscape



# Adversarial Attack

模型训练:

$$\min_{\theta} \sum_{(x_i, y_i) \in D_{train}} L(f_{\theta}(x_i), y_i)$$



对抗攻击:

$$\max_{x'} L(f_{\theta}(x'), y) \text{ subject to } \|x' - x\|_p \leq \epsilon \text{ for } x \in D_{test}$$

分类错误

扰动很小

测试阶段攻击

扰动上限:  $\|x' - x\|_{p=1, 2 \text{ or } \infty}$ , e.g.,  $\|\cdot\|_{\infty} \leq \frac{8}{255}$

Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[J]. ICLR 2014.  
Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples[J]. ICLR 2015.



# Performance Metrics

- Measurement of clean performance:

$$\text{accuracy (clean accuracy)} = \frac{\text{correctly classified clean samples}}{\text{total clean samples}}$$

- Measurement of adversarial robustness:

$$\text{robustness (robust accuracy)} = \frac{\text{correctly classified advs samples}}{\text{total advs samples}}$$

- Other metrics: maximum perturbation for 100% attack success rate



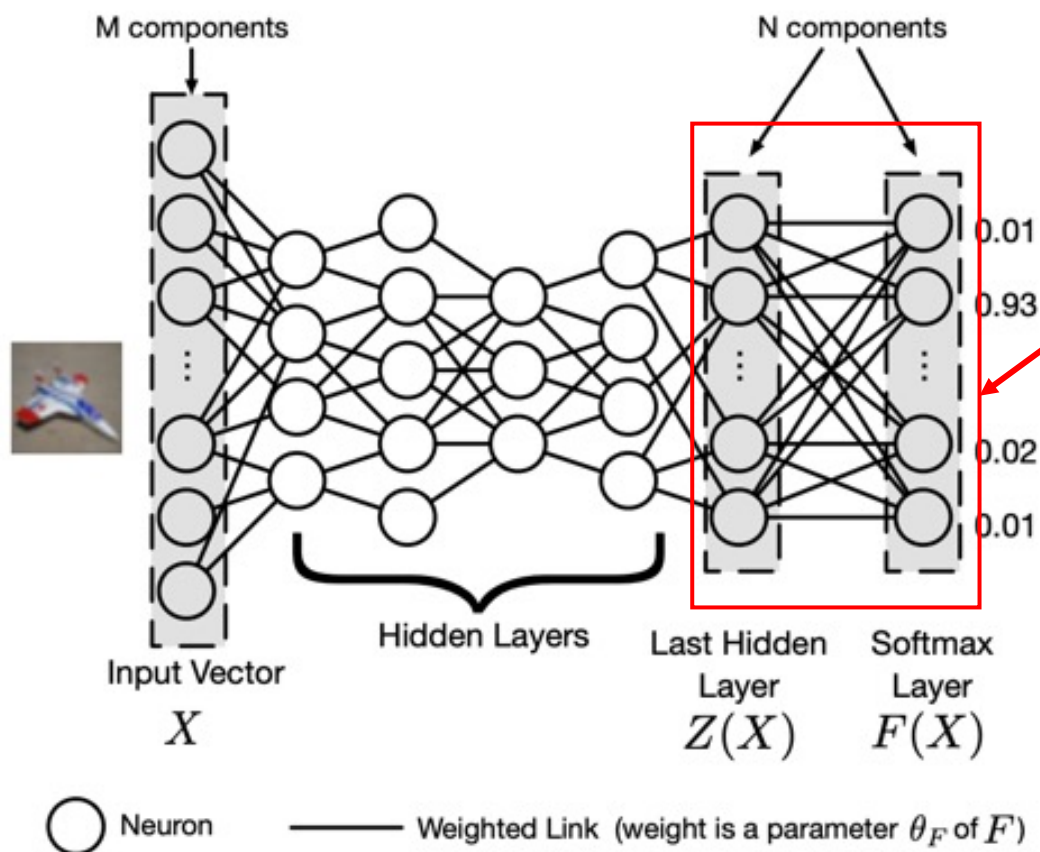
# Defense Methods

---

- ❑ Early Defense Methods
- ❑ Early Adversarial Training Methods
- ❑ Advanced Adversarial Training Methods
- ❑ Remaining Challenges and Recent Progress



# Defensive Distillation

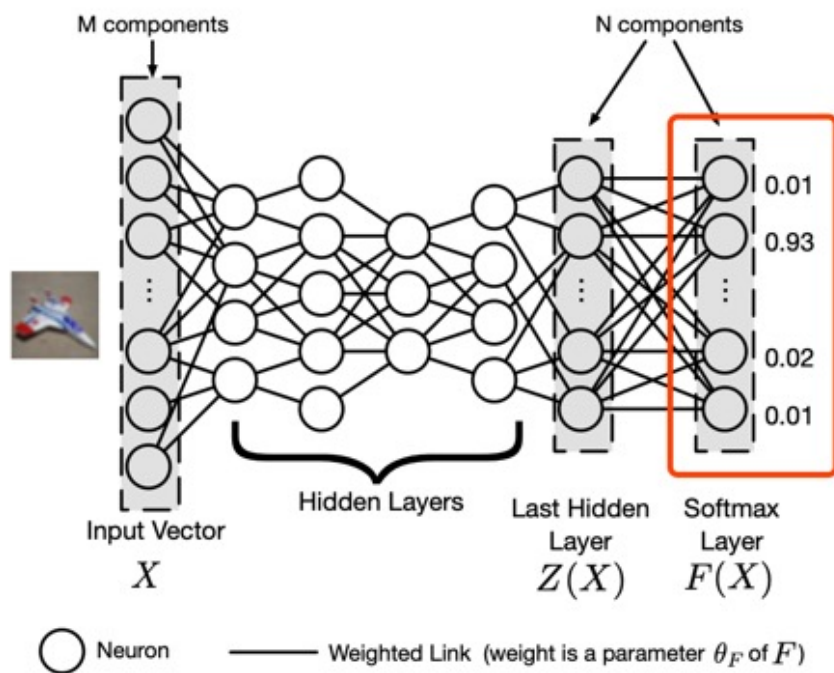


- **Making large logits change to be "small"**
  - Scaling up logits by a few magnitudes;
  - Retrain the last layer with scaled logits;

Papernot et al. Distillation as a defense to adversarial perturbations against deep neural networks, S&P, 2016



# Defensive Distillation



$$f_{\theta}^i(x) = \frac{e^{z_i(x)}}{\sum_{j=1}^K e^{z_j(x)}}$$

1

Distillation with temperature  $T$

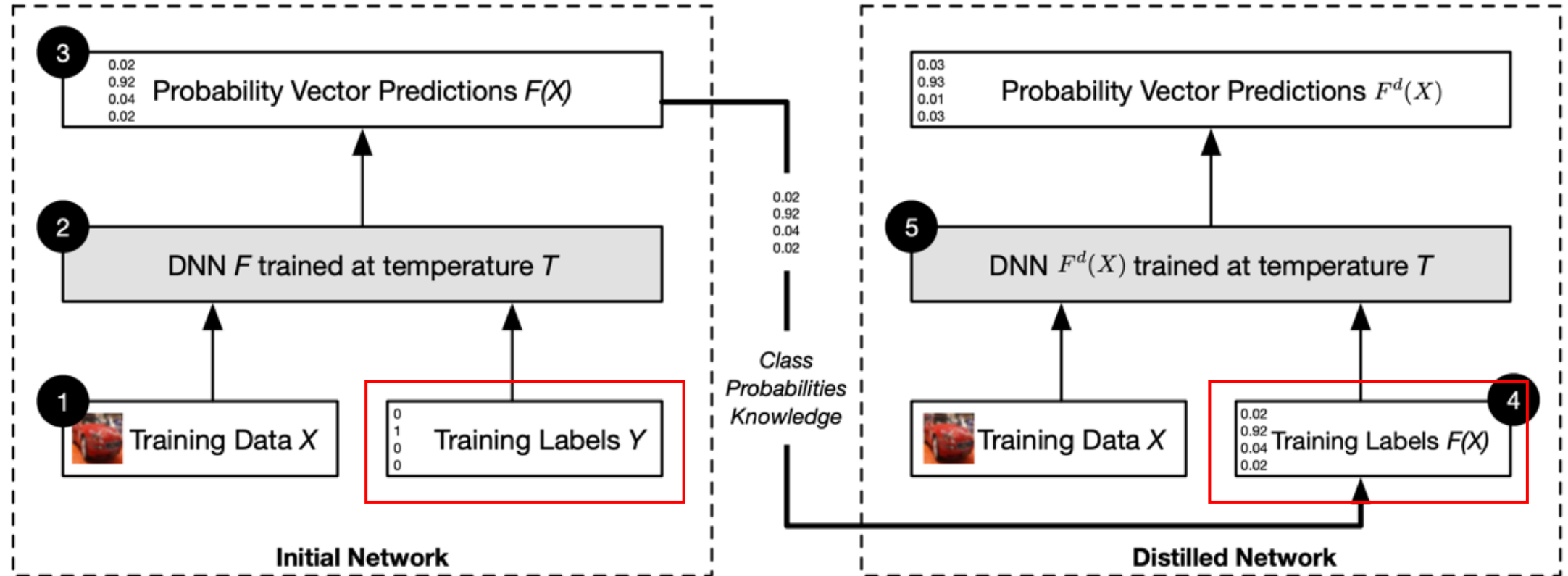
$$f_{\theta}^i(x) = \frac{e^{z_i(x)/T}}{\sum_{j=1}^K e^{z_j(x)/T}}$$

2

Papernot et al. Distillation as a defense to adversarial perturbations against deep neural networks, S&P, 2016



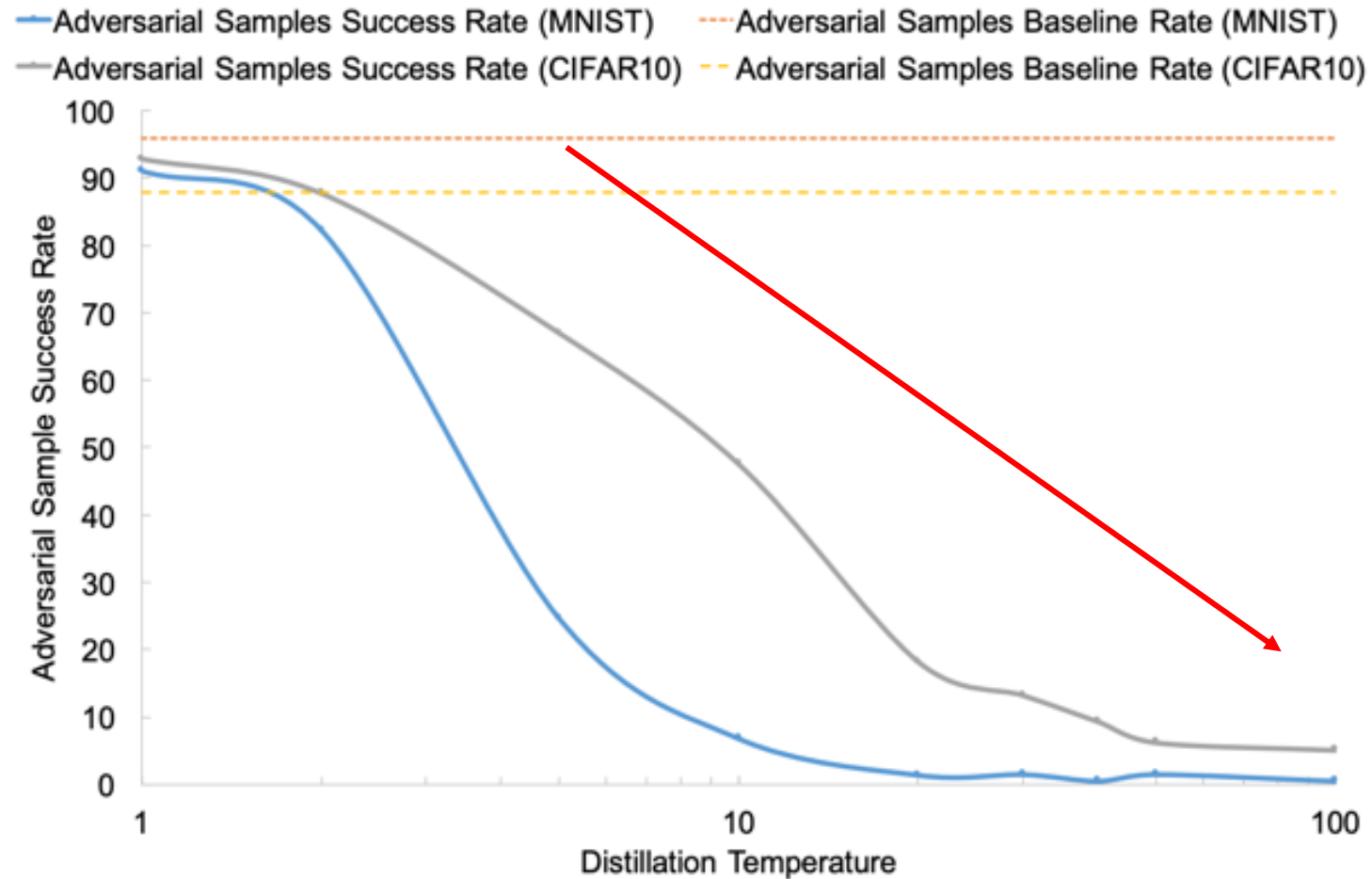
# Defensive Distillation



Papernot et al. Distillation as a defense to adversarial perturbations against deep neural networks, S&P, 2016



# Defensive Distillation



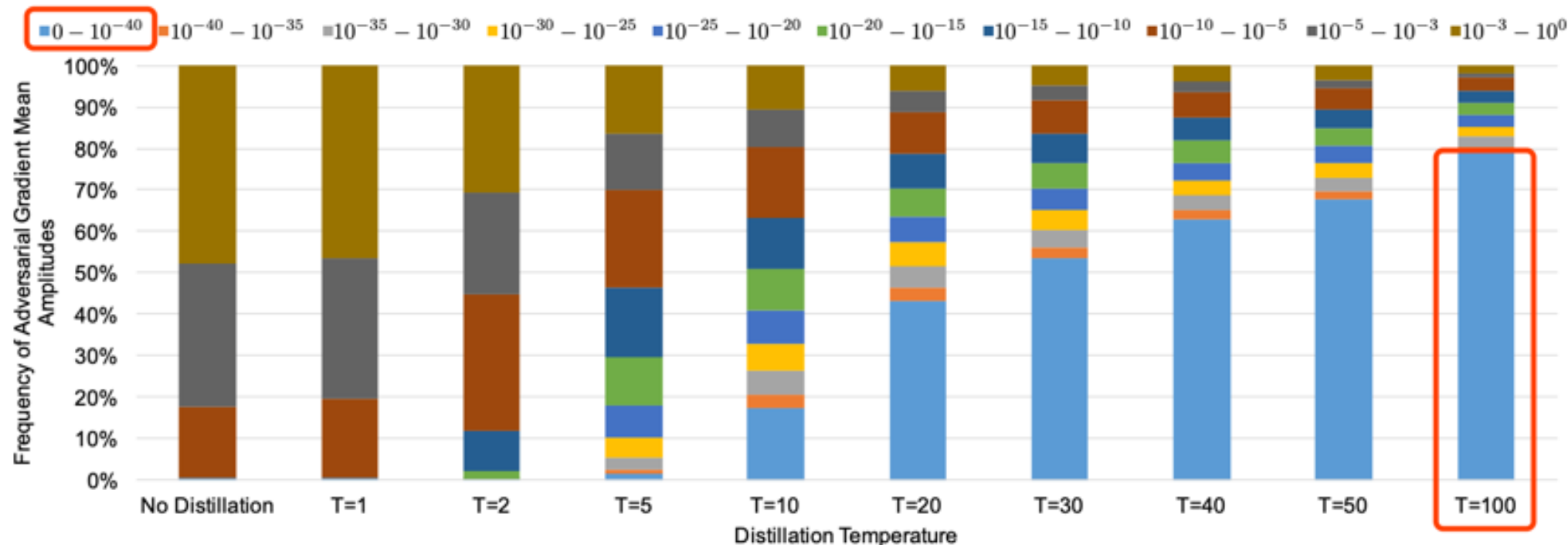
Papernot et al. Distillation as a defense to adversarial perturbations against deep neural networks, S&P, 2016





# Defensive Distillation

Distillation makes input gradients  $\nabla_x L(f_\theta(x), y)$  to be small!



Papernot et al. Distillation as a defense to adversarial perturbations against deep neural networks, S&P, 2016

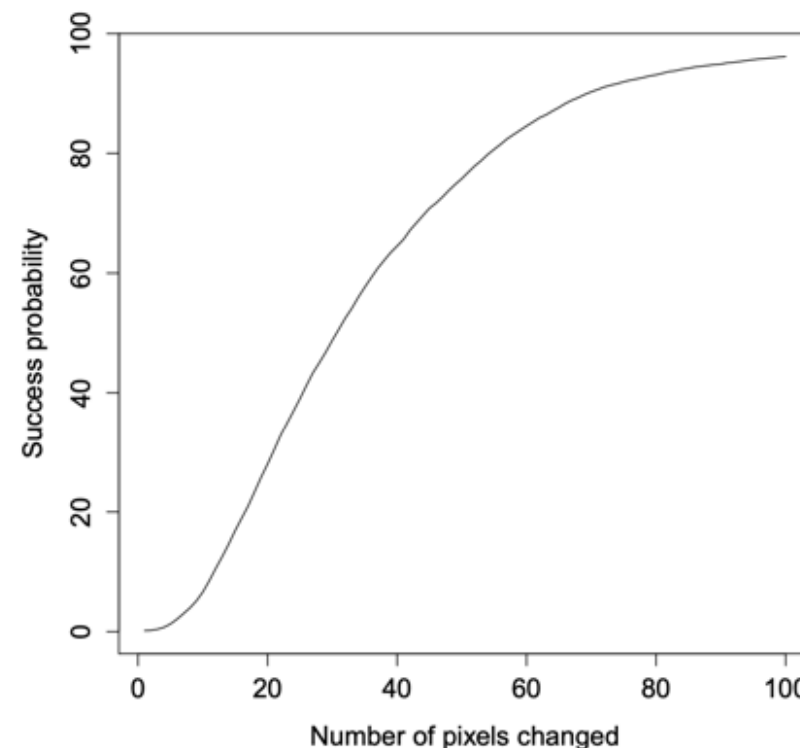


# Defensive Distillation Is Not Robust

It can be evaded by attacking the distilled network with the temperature T.

$$x' = x + \varepsilon \cdot \text{sign } \nabla_x L(\hat{f}_\theta(x), y)$$

$$\hat{f}_\theta(x) = \text{softmax}(z(x)/T)$$



Carlini, Nicholas, and David Wagner. "Defensive distillation is not robust to adversarial examples." arXiv preprint arXiv:1607.04311 (2016).



# Lessons Learned

- ❑ Distillation is not a good solution for adversarial robustness
- ❑ Vanishing input gradients can still be recovered by a reverse operation
- ❑ A defense should be evaluated against the adaptive attack to prove real robustness

Carlini, Nicholas, and David Wagner. "Defensive distillation is not robust to adversarial examples." arXiv preprint arXiv:1607.04311 (2016).



# Input Gradients Regularization

Directly regularize the input gradients  $\nabla_x L(f_\theta(x), y)$  to be small

$$L_{reg} = \underbrace{L(f_\theta(x), y)}_{\text{Classification loss}} + \underbrace{\lambda \|\nabla_x L(f_\theta(x), y)\|_2^2}_{\text{Input gradients regularization}}$$

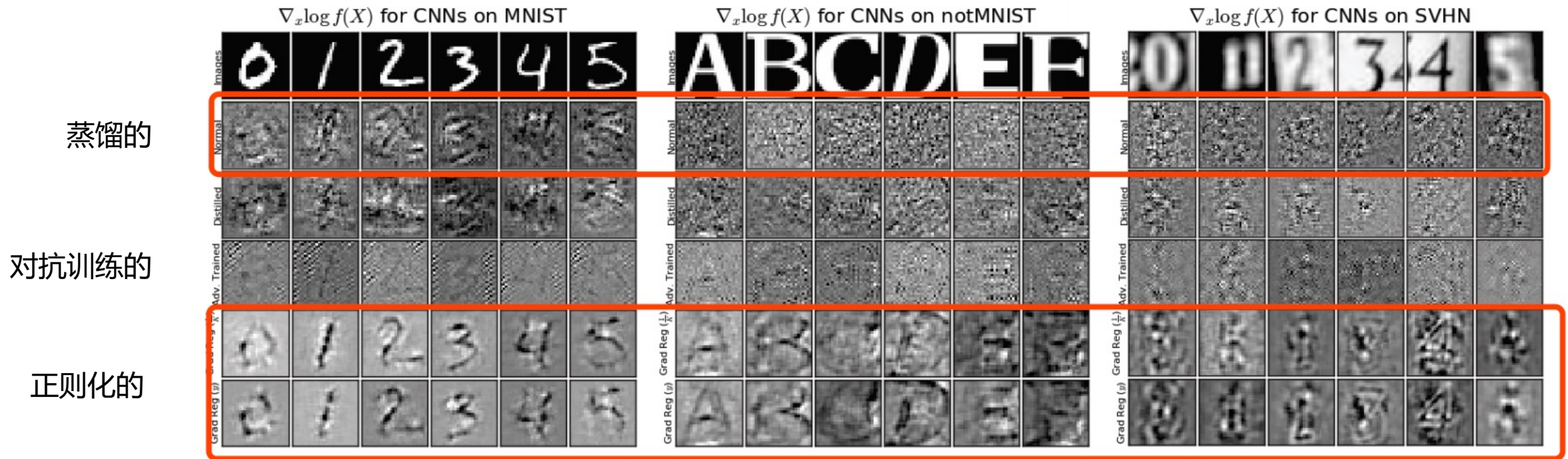
Related to the **double backpropagation** proposed by Drucker and Le Cun (1992):

$$\arg \min_{\theta} H(y, \hat{y}) + \lambda \|\nabla_x H(y, \hat{y})\|_2^2$$

Ross et al. "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients." *AAAI*, 2018.  
Drucker, Harris, and Yann Le Cun. "Improving generalization performance using double backpropagation." *TNN*, 1992.



# Input Gradients Regularization



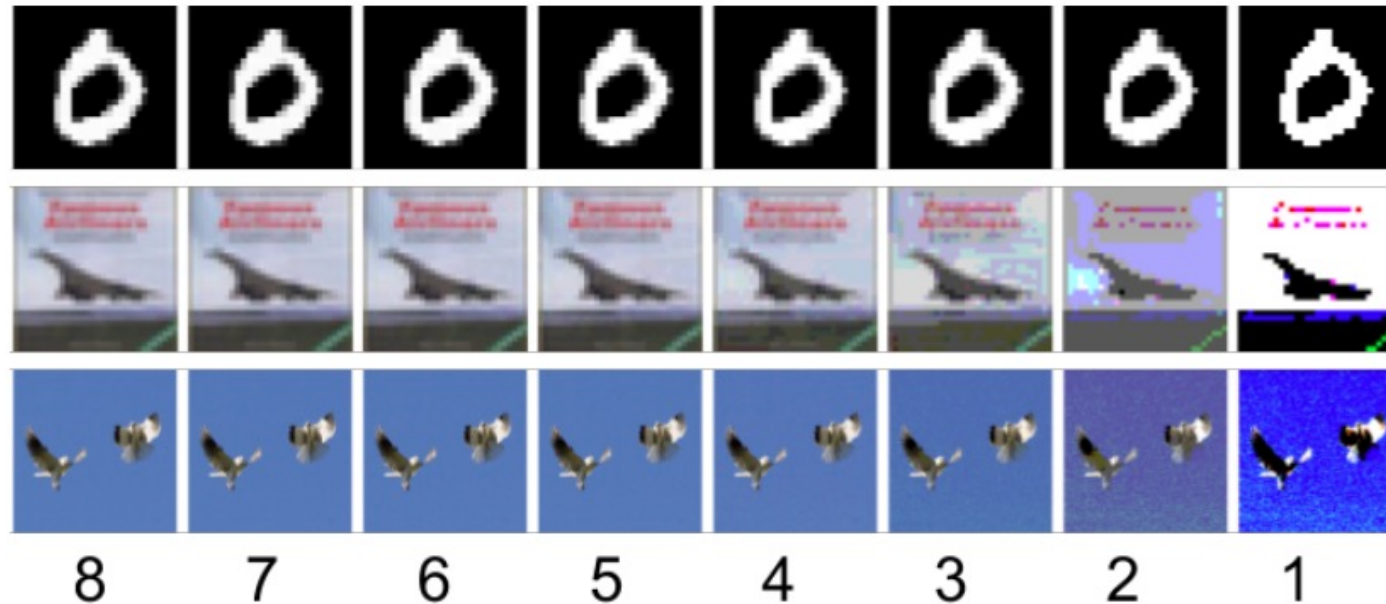
Issues: 1) limited adversarial robustness, 2) hurts learning

Ross et al. "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients." AAAI, 2018.



# Feature Squeezing

Compress the input space



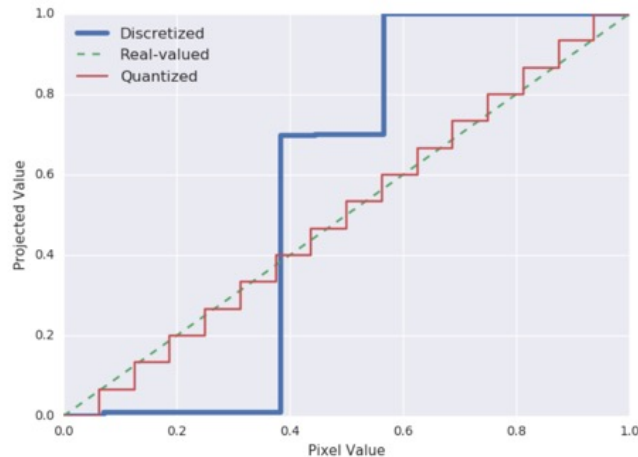
It also hurts performance on large-scale image datasets.

Xu et al. "Feature squeezing: Detecting adversarial examples in deep neural networks." *NDSS*, 2018.

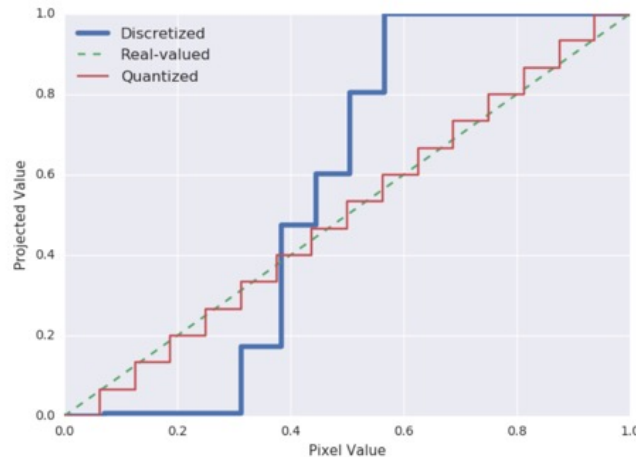
# Thermometer Encoding

Discretize the input to break small noise

Real-valued	Quantized	Discretized (one-hot)	Discretized (thermometer)
0.13	0.15	[0100000000]	[0111111111]
0.66	0.65	[0000001000]	[0000001111]
0.92	0.95	[0000000001]	[0000000001]



(a) One hot encoding



(b) Thermometer encoding

$$\tau(x_{i,j,c})_k = 1 \text{ if } x_{i,j,c} > k/l,$$

$$\tau(0.66) = 1111110000$$

**Proposed Thermometer Encoding**

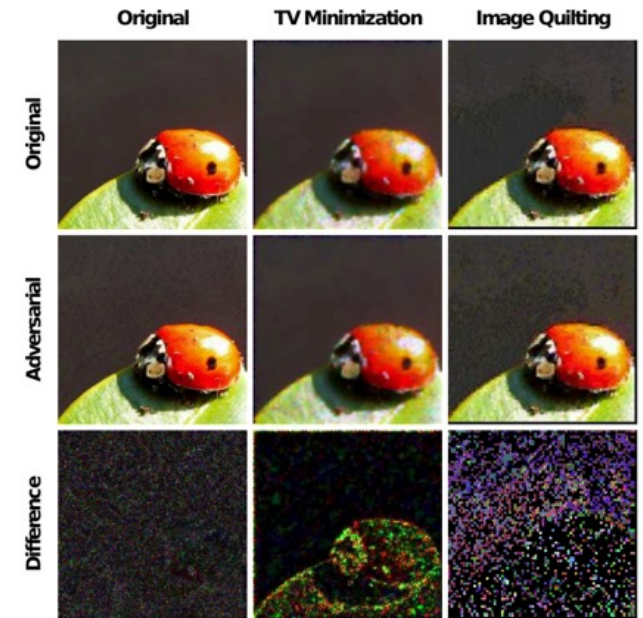
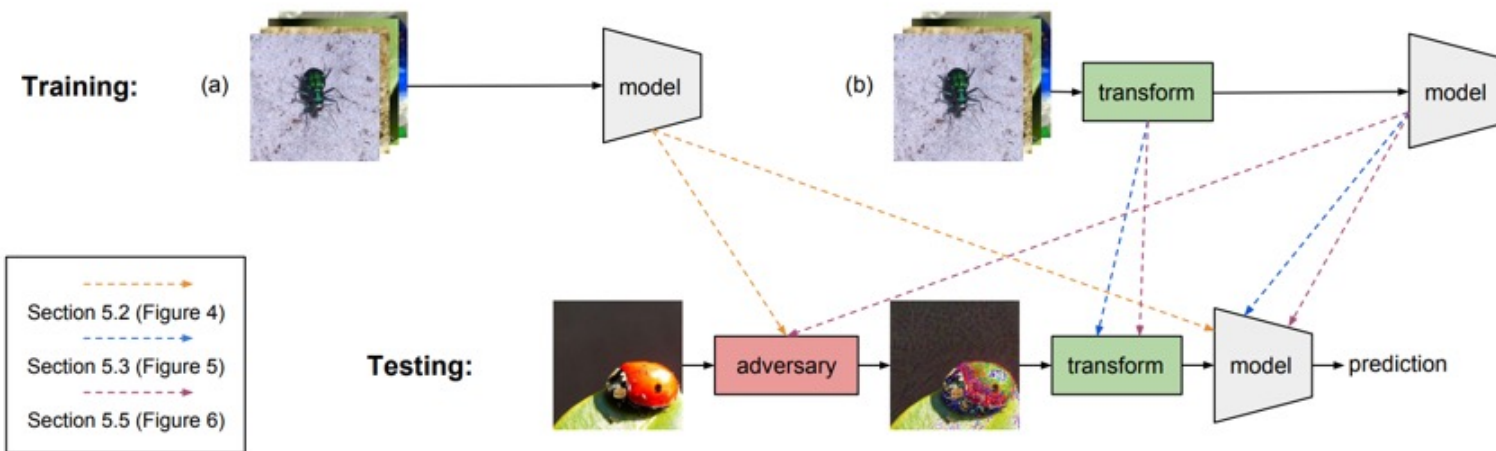
Buckman et al. "Thermometer encoding: One hot way to resist adversarial examples." *ICLR*, 2018.





# Input Transformations

- ❑ Image cropping and rescaling
- ❑ Bit-depth reduction
- ❑ JPEG compression
- ❑ Total variance minimization
- ❑ Image quilting



Guo et al. "Countering Adversarial Images using Input Transformations." *ICLR*, 2018.



# Obfuscated Gradients = Fake Robustness

Backward Pass Differentiable Approximation (BPDA): can break **non-differentiable operation** based defenses

$$g(x) \approx f^i(x)$$

find a linear approximation of the non-differentiable operations, e.g., discretization, compression etc.

Expectation Over Transformation (EOT) can break **randomization** based defenses

$$\arg \max_{x'} \mathbb{E}_{t \sim T} [\log P(y_t | t(x'))]$$

$$\begin{aligned} \text{subject to } & \mathbb{E}_{t \sim T} [d(t(x'), t(x))] < \epsilon \\ & x \in [0, 1]^d \end{aligned}$$

T: a set of randomized transformations

Athalye et al. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples." *ICML*, 2018.  
Athalye et al. Synthesizing robust adversarial examples. *ICML*, 2018.



# BPDA+EOT breaks 7 defenses published at ICLR 2018

Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	0%*
Ma et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	5%
Guo et al. (2018)	ImageNet	0.005 ( $\ell_2$ )	0%*
Dhillon et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	0%
Xie et al. (2018)	ImageNet	0.031 ( $\ell_\infty$ )	0%*
Song et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	9%*
Samangouei et al. (2018)	MNIST	0.005 ( $\ell_2$ )	55%**
Madry et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	47%
Na et al. (2018)	CIFAR	0.015 ( $\ell_\infty$ )	15%

We got a survivor!



Athalye et al. “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples.” *ICML*, 2018.  
Athalye et al. Synthesizing robust adversarial examples. *ICML*, 2018.



# How to Properly Evaluate a Defense?

- ❑ Do not blindly apply multiple (similar) attacks
- ❑ Try at least one gradient-free attack and one hard-label attack
- ❑ Perform a transferability attack using a similar substitute model.
- ❑ For randomized defenses, properly ensemble over randomness
- ❑ For non-differentiable components, apply differentiable techniques (BPDA)
- ❑ Verify that the attacks have converged under the selected hyperparameters
- ❑ Carefully investigate attack hyperparameters and report those selected
- ❑ Compare against prior work and explain important differences
- ❑ Test broader threat models when proposing general defenses

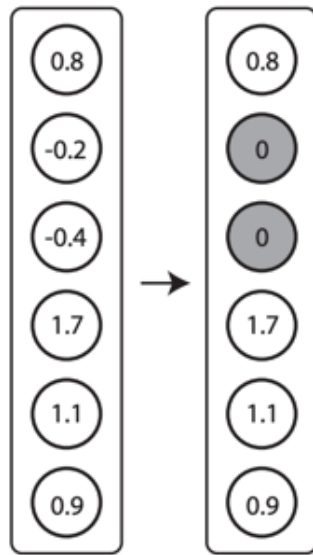


Carlini, Nicholas, et al. “On evaluating adversarial robustness.” *arXiv preprint arXiv:1902.06705* (2019). Athalye et al. “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples.” *ICML*, 2018



# Robust Activation Functions

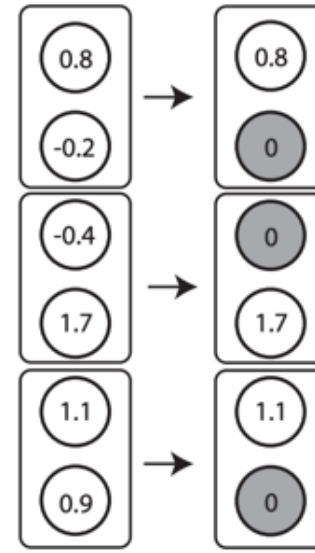
Block the internal activation: break the continuity



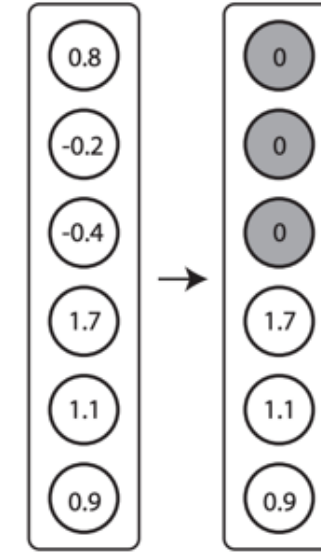
ReLU



Maxpool



LWTA



k-WTA

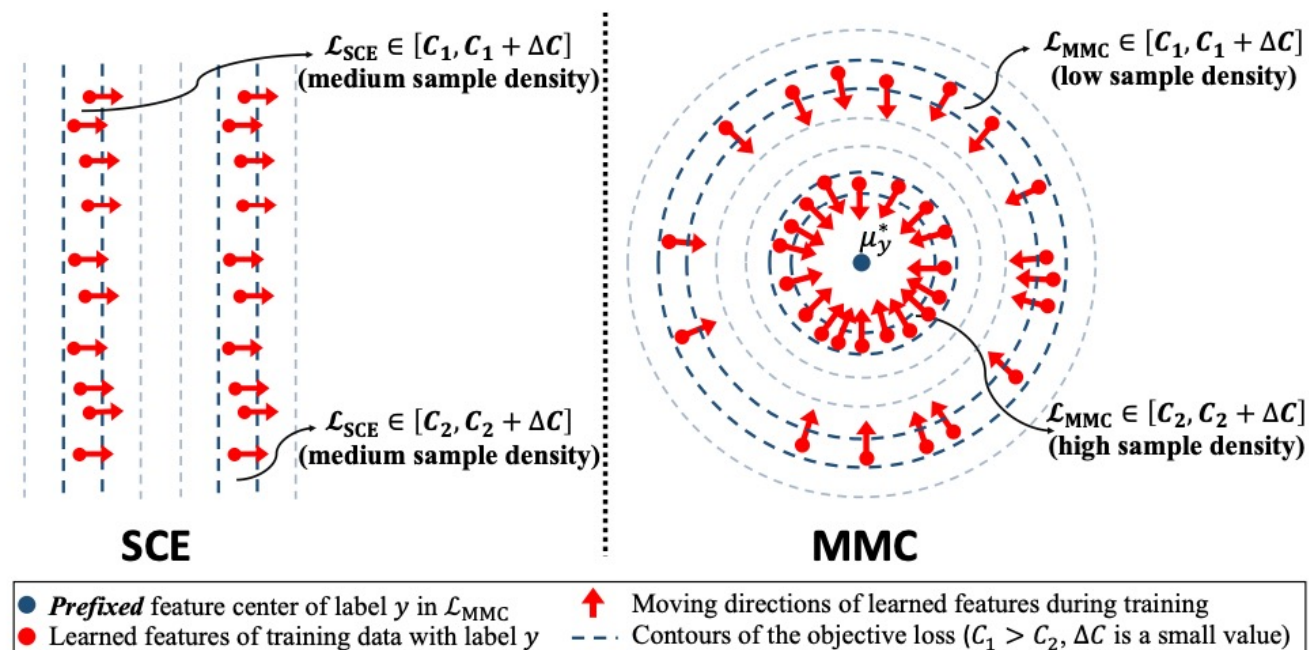
k-Winners-Take-All (k-WTA) activation

Xiao et al. "Enhancing Adversarial Defense by k-Winners-Take-All." ICLR, 2020.



# Robust Loss Function

## Max-Mahalanobis center (**MMC**) loss



Max-Mahalanobis center (MMC); SCE: softmax cross entropy

# Robust Inference

## Mixup Inference (MI)

$$\hat{x}_k = \lambda x + (1 - \lambda) s_k$$

---

### Algorithm 1 Mixup Inference (MI)

---

**Input:** The mixup-trained classifier  $F$ ; the input  $x$ .

**Hyperparameters:** The sample distribution  $p_s$ ; the mixup ratio  $\lambda$ ; the number of execution  $N$ .

Initialize  $F_{\text{MI}}(x) = \mathbf{0}$ ;

**for**  $k = 1$  **to**  $N$  **do**

    Sample  $y_{s,k} \sim p_s(y_s)$ ,  $x_{s,k} \sim p_s(x_s|y_{s,k})$ ;

    Mixup  $x$  with  $x_{s,k}$  as  $\tilde{x}_k = \lambda x + (1 - \lambda)x_{s,k}$ ;

    Update  $F_{\text{MI}}(x) = F_{\text{MI}}(x) + \frac{1}{N} F(\tilde{x}_k)$ ;

**end for**

**Return:** The prediction  $F_{\text{MI}}(x)$  of input  $x$ .

---



# New Adaptive Attacks Break These Defenses

Defense		Attack Themes					
		T1	T2	T3	T4	T5	T6
Appendix B	<i>k-Winners Take All</i> [XZZ20]	●				●	●
Appendix C	<i>The Odds are Odd</i> [RKH19]			●	●		
Appendix D	<i>Generative Classifiers</i> [LBS19]		●	●			
Appendix E	<i>Sparse Fourier Transform</i> [BMV18]	●	●				
Appendix F	<i>Rethinking Cross Entropy</i> [PXD <sup>+</sup> 20]			●		●	
Appendix G	<i>Error Correcting Codes</i> [VS19]	●	●				
Appendix H	<i>Ensemble Diversity</i> [PXD <sup>+</sup> 19]					●	
Appendix I	<i>EMPIR</i> [SRR20]		●			●	
Appendix J	<i>Temporal Dependency</i> [YLCS19]	●		●	●	●	
Appendix K	<i>Mixup Inference</i> [PXZ20]	●					
Appendix L	<i>ME-Net</i> [YZKX19]	●	●		●		●
Appendix M	<i>Asymmetrical Adv. Training</i> [YKR20]			●	●		●
Appendix N	<i>Weakness into a Strength</i> [HYG <sup>+</sup> 19]	●	●	●	●		

**T1:** Attack the full defense  
**T2:** Target important defense parts  
**T3:** Simplify the attack  
**T4:** Ensure consistent loss function  
**T5:** Optimize with different methods  
**T6:** Use strong adaptive attacks

Tramer et al. "On adaptive attacks to adversarial example defenses." *NeurIPS*, 2020.

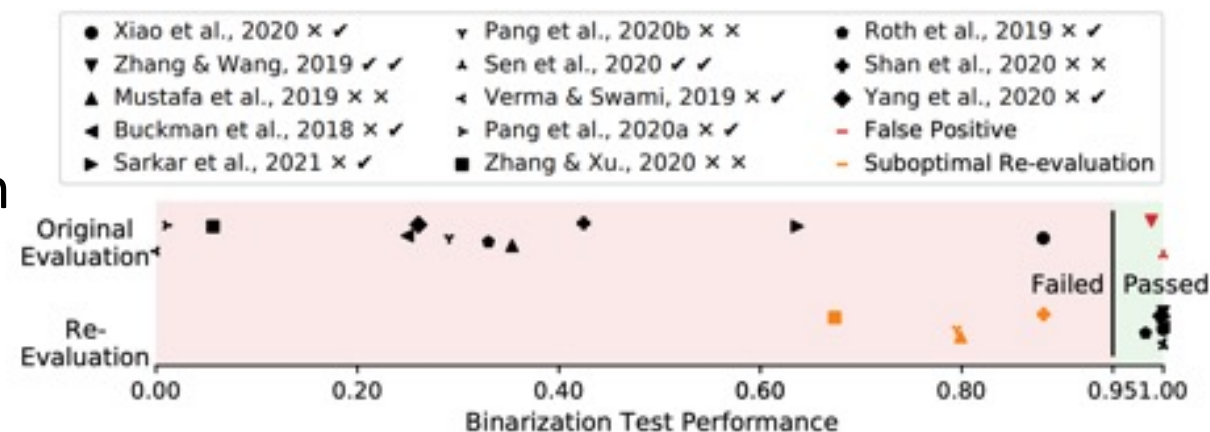




# How to Evaluate a Defense?

- ❑ Strong attacks:
  - ✓ **AutoAttack** (one must-to-test attack)
  - ✓ **Margin Decomposition** (MD) attack (my work, slightly better than AutoAttack)
  - ✓ **Minimum-Margin** (MM) attack (new state-of-the-art attack to test)

- ❑ Extra robustness tests
  - ✓ Attack unit tests (Zimmermann et al, 2022)



Croce and Hein. "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks." *ICML*, 2020.

Gao et al. Fast and Reliable Evaluation of Adversarial Robustness with Minimum-Margin Attack, *ICML* 2022.

Zimmermann et al. "Increasing Confidence in Adversarial Robustness Evaluations." *arXiv preprint arXiv:2206.13991* (2022).





# Adversarial Training

The idea is simple: just train on adversarial examples!

□ 对抗训练是一种数据增广方法

– 原始数据-> 对抗攻击-> 对抗样本->模型训练

□ 对抗训练是一个min-max鲁棒优化框架:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n \max_{\|x_i - x_i^0\| \leq \epsilon} L(f_{\theta}(x_i), y_i)$$

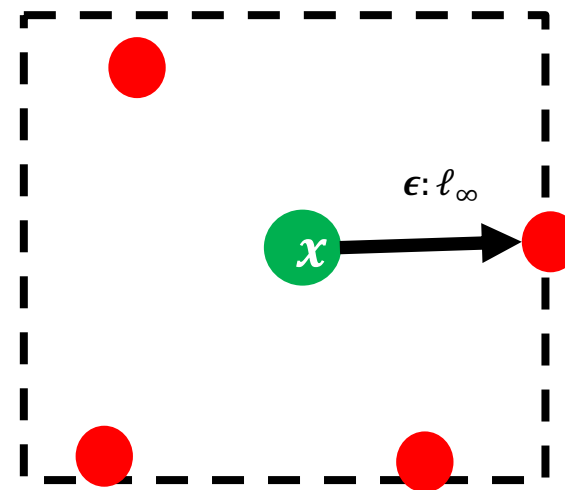
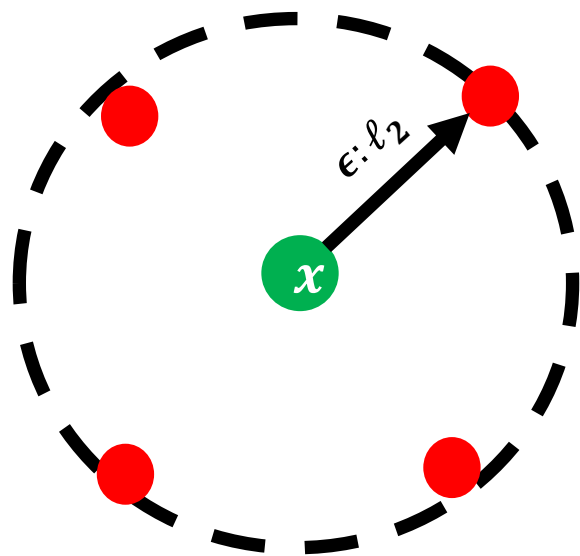
$L(f_{\theta}(x_i), y_i) = -y_i \log f_{\theta}(x_i)$  (交叉熵损失函数)

$x_i^0$  : 原始训练样本,  $y_i$  :  $x_i^0$  的正确类别.



# Adversarial Training

在以 $x$ 为中心的  $\epsilon$ -球范围内寻找对抗性最强的样本



$\ell_\infty$  范式更常见



# Adversarial Training

**Adversarial training produces smooth decision boundary**



正常边界



生成对抗样本



训练后

# Early Adversarial Training Methods

- 2014年, Szegedy et al. 在其解释对抗样本的论文中已探索了对抗训练, 用L-BFGS攻击对神经网络每一层生成对抗样本, 并添加到训练过程中。发现: 深层对抗样本更能提高鲁棒性
- 2015年, Goodfellow et al. 提出使用FGSM (单步) 攻击生成的对抗样本来训练神经网络

$$\min_{\theta} \mathbb{E}_{(x,y) \in D} [\alpha \mathcal{L}_{\text{CE}}(f(\mathbf{x}), y) + (1 - \alpha) \mathcal{L}_{\text{CE}}(f(\mathbf{x}_{\text{adv}}), y)]$$
$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}_{\text{CE}}(f(\mathbf{x}), y))$$

Goodfellow等人并未使用中间层的对抗样本, 因为发现中间层对抗样本没有提升

Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[J]. ICLR 2014.  
Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples[J]. ICLR 2015.



# Min-max Robust Optimization

## The First Proposal of Min-Max Optimization

$$\underbrace{\min_{\theta} \mathbb{E}_{(x,y) \in D}}_{\substack{\text{外部最小化} \\ \text{Outer minimization}}} \underbrace{\max_{\|r\| \leq \epsilon} \mathcal{L}(f(\boxed{x+r}), y)}_{\substack{\text{内部最大化} \\ \text{Inner maximization}}}$$

$\chi_{adv}$

Nokland et al. Improving back-propagation by adding an adversarial gradient. arXiv:1510.04189, 2015.  
Huang et al. Learning with a strong adversary, ICLR 2016. Shaham et al. Understanding adversarial training: Increasing local stability of neural nets through robust optimization, arXiv:1511.05432, 2015



# Virtual Adversarial Training

**VAT: a method to improve generalization**

$$\min_{\theta} \max_{\|\mathbf{x}_{\text{adv}} - \mathbf{x}\|_2 \leq \epsilon} \mathbb{E}_{(\mathbf{x}, y) \in D} [\mathcal{L}_{\text{CE}}(f(\mathbf{x}), y) + \lambda \mathcal{L}_{\text{KL}}(f(\mathbf{x}), f(\mathbf{x}_{\text{adv}}))]$$

- ❑ Differences to adversarial training
  1. L2 regularized perturbation
  2. Use both clean and adv examples for training
  3. Use KL divergence to generate adv examples

Miyato et al. Distributional smoothing with virtual adversarial training. ICLR 2016.



# Weaknesses of Early AT Methods

$$\min_{\theta} \mathbb{E}_{(x,y) \in D} \max_{\|\mathbf{r}\| \leq \epsilon} \mathcal{L}(f(\mathbf{x} + \mathbf{r}), y)$$

- ❑ Use FGSM or BIM to solve the inner maximization problem
- ❑ FGSM and BIM were later found to be **weak attacks**
- ❑ Overfit to  $\epsilon$ -robustness (**not robust to  $<\epsilon$  attacks**)
- ❑ Overfit to single-step attacks (**not robust to multi-step attacks**)

✓ These methods are **fast**! Only takes **x2** time of standard training



# PGD Adversarial Training

Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	0%*
Ma et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	5%
Guo et al. (2018)	ImageNet	0.005 ( $\ell_2$ )	0%*
Dhillon et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	0%
Xie et al. (2018)	ImageNet	0.031 ( $\ell_\infty$ )	0%*
Song et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	9%*
Samangouei et al. (2018)	MNIST	0.005 ( $\ell_2$ )	55%**
Madry et al. (2018)	CIFAR	0.031 ( $\ell_\infty$ )	47%
Na et al. (2018)	CIFAR	0.015 ( $\ell_\infty$ )	15%

We got a survivor!



Athalye et al. “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples.” *ICML*, 2018.  
Athalye et al. Synthesizing robust adversarial examples. *ICML*, 2018.





# PGD Adversarial Training

## A Saddle Point Problem

$$\underbrace{\min_{\theta} \mathbb{E}_{(x,y) \sim D}}_{\text{外部最小化}} \underbrace{\max_{\|r\| \leq \epsilon} \mathcal{L}(f(x+r), y)}_{\text{内部最大化}}$$

A saddle point (**constrained** bi-level optimization) problem

In constrained optimization, **Projected Gradient Descent (PGD)** is the best first-order solver

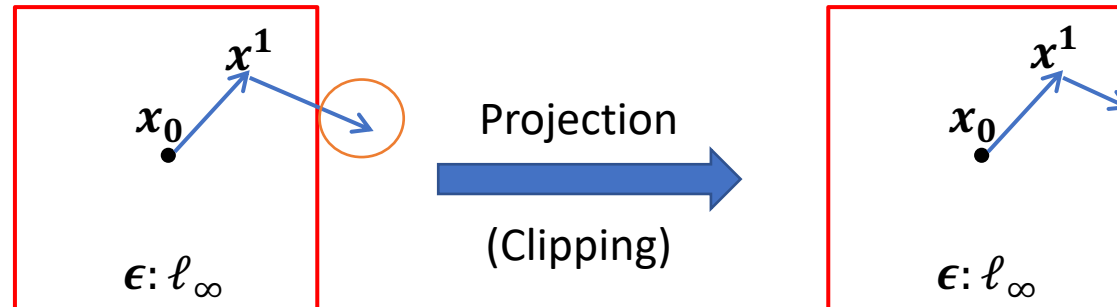
Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks." *ICLR*. 2018.



# PGD Adversarial Training

## Projected Gradient Descent (PGD)

$$\mathbf{x}_{\text{adv}}^{t+1} = \text{Proj}_{\mathbf{x}+\mathcal{S}}(\mathbf{x}^t + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}^t})\mathcal{L}(\theta, \mathbf{x}^t, y))$$



- ❑ PGD is an optimizer
- ❑ PGD is also known as an adv attack in the field of AML

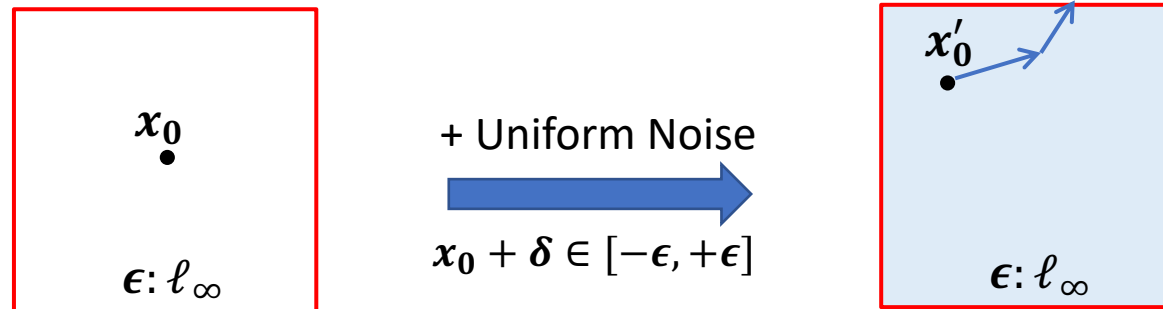
Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks." *ICLR*. 2018.



# PGD Adversarial Training

## Projected Gradient Descent (PGD)

$$\mathbf{x}_{\text{adv}}^{t+1} = \text{Proj}_{\mathcal{X}+\mathcal{S}}(\mathbf{x}^t + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}^t})\mathcal{L}(\theta, \mathbf{x}^t, y))$$



□ Random initialization

Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks." *ICLR*. 2018.



# PGD Adversarial Training

## Characteristics of PGD adversarial training

- 只在对抗样本上训练模型
- 通用鲁棒性： $< \epsilon$  鲁棒性和多步攻击鲁棒性
- 需要大容量模型
- 需要更多训练数据
- 对抗训练会产生平滑的决策边界
- 对抗训练会对内部激活产生一种截断效果
- 对抗训练会强制模型学习鲁棒特征
- 鲁棒性提升的同时干净准确率会下降
- 训练很耗时，相当于5-10倍普通训练

Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks." *ICLR*. 2018.  
Ilyas et al. "Adversarial examples are not bugs, they are features." *NeurIPS*, 2019.

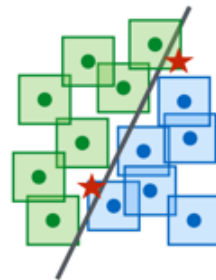


# PGD Adversarial Training

决策边界：



普通训练

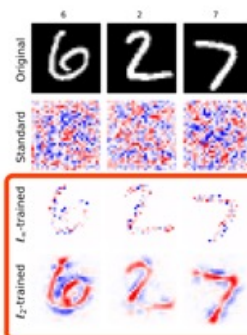


对抗样本

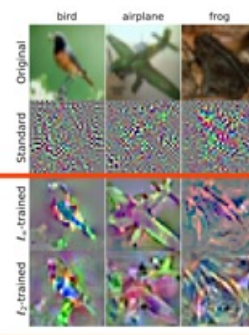


对抗训练

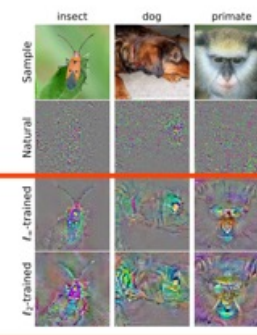
鲁棒特征：



(a) MNIST



(b) CIFAR-10



(c) Restricted ImageNet

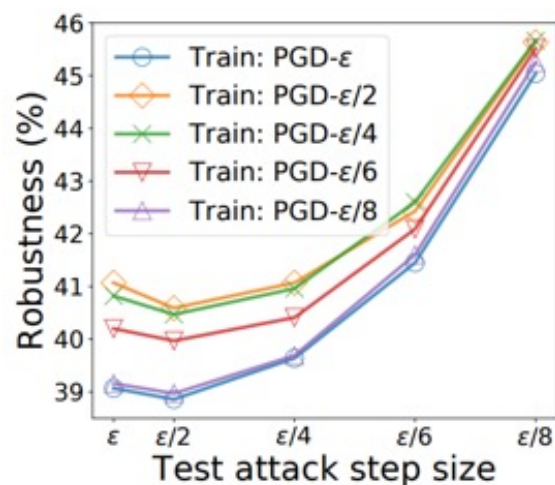
普通训练

对抗训练

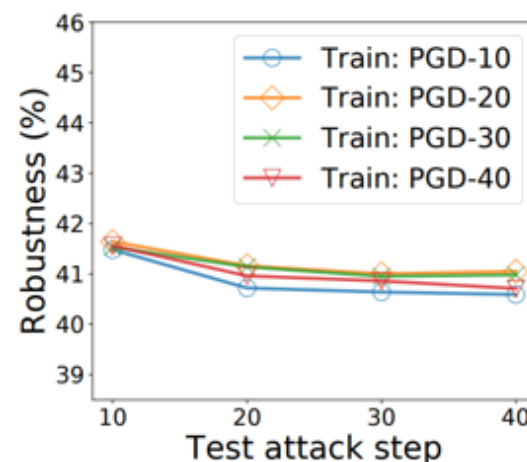
Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks." *ICLR*. 2018.  
Ilyas et al. "Adversarial examples are not bugs, they are features." *NeurIPS*, 2019.



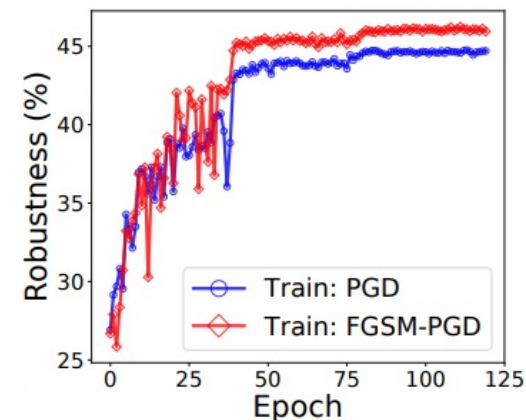
# Dynamic Adversarial Training (DART)



PGD步长对鲁棒性影响



PGD步数对鲁棒性影响



训练初期使用简单攻击

- 对于10步对抗训练（PGD-10 Adversarial Training）来说：
  - 最优步长是 $\epsilon/2$ 和 $\epsilon/4$
  - 步数影响不大，只要足够探索到 $\epsilon$ -ball的边界
- 训练初期使用弱对抗样本反而会提高鲁棒性

Wang et al. "On the Convergence and Robustness of Adversarial Training." *ICML*. 2019.



# Dynamic Adversarial Training (DART)

## How to measure the convergence of the inner maximization?

### Definition ( First-Order Stationary Condition (FOSC))

Given a sample  $x^0 \in X$ , let  $x^k$  be an intermediate example found at the  $k^{\text{th}}$  step of the inner maximization. The First-Order Stationary Condition of  $x^k$  is:

$$c(x^k) = \max_{x \in X} \langle x - x^k, \nabla_x f(\theta, x^k) \rangle,$$

where  $X = \{x | \|x - x^0\|_\infty \leq \epsilon\}$  is the input domain of the  $\epsilon$ -ball around normal example  $x^0$ ,  $f(\theta, x^k) = \ell(h_\theta(x^k), y)$ , and  $\langle \cdot \rangle$  is the inner product.

### FOSC:

- Inspired by the Frank-Wolfe gap for constrained min-max optimization.
- Smaller value of  $c(x^k)$  indicates **better** convergence of the inner maximization.
- It has a close-form solution, affine invariant and norm independent.

Wang et al. "On the Convergence and Robustness of Adversarial Training." *ICML*. 2019.



# Dynamic Adversarial Training (DART)

## Dynamic Adversarial Training:

- Weak attack for early training, strong attack for later training
- **Weak** attack improves **generalization**, **strong** attack improves **final robustness**.

## Convergence analysis:

**Theorem 1.** Suppose Assumptions 1, 2 and 3 hold. Let  $\Delta = L_S(\theta^0) - \min_{\theta} L_S(\theta)$ . If the step size of the outer minimization is set to  $\eta_t = \eta = \min(1/L, \sqrt{\Delta/L\sigma^2T})$ . Then the output of Algorithm 1 satisfies

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla L_S(\theta^t)\|_2^2] \leq 4\sigma\sqrt{\frac{L\Delta}{T}} + \frac{5L_{\theta x}^2\delta}{\mu},$$

where  $L = (L_{\theta x}L_{x\theta}/\mu + L_{\theta\theta})$ .

Robustness on CIFAR-10 with WideResNet

Defense	Clean	FGSM	PGD-20	C&W $_{\infty}$
<i>Madry's</i>	<b>87.3</b>	56.1	45.8	46.8
<i>Curriculum</i>	77.43	57.17	46.06	42.28
<b><i>Dynamic</i></b>	85.03	<b>63.53</b>	<b>48.70</b>	<b>47.27</b>

DART improves robustness

Wang et al. "On the Convergence and Robustness of Adversarial Training." *ICML*. 2019.

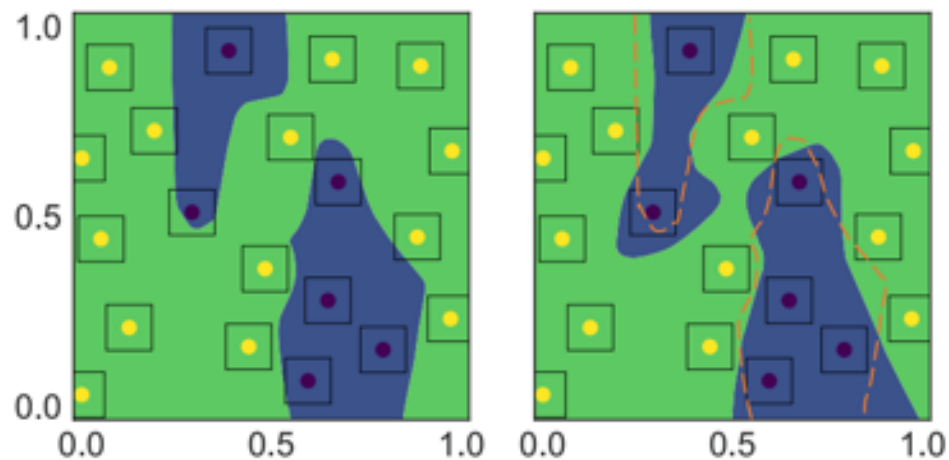




# TRADES

Use distribution loss (KL) for inner and outer optimizations

$$\min_{\theta} \mathbb{E}_{(x,y) \sim D} \left[ \underbrace{\mathcal{L}_{\text{CE}}(f(\mathbf{x}), y)}_{\text{提升准确率}} + \lambda \underbrace{\max_{\|r\| \leq \epsilon} \mathcal{L}_{\text{KL}}(f(\mathbf{x}), f(\mathbf{x} + \mathbf{r}))}_{\text{提升鲁棒性}} \right]$$



Zhang et al. "Theoretically principled trade-off between robustness and accuracy." ICML, 2019.



# TRADES

## Characteristics of TRADES

TRADES既改进了内部最大化又改进了外部最小化

- ❑ 使用KL监督对抗样本的生成，鲁棒性提升显著
- ❑ 干净样本也参与训练，有利于模型收敛和干净准确率
- ❑ 基于KL的对抗样本生成包含自适应的过程
- ❑ 能成训练得到比PGD对抗训练更平滑的决策边界

Winning solutions of  
NeurIPS 2018  
Adversarial Vision  
Challenge

Zhang et al. "Theoretically principled trade-off between robustness and accuracy." ICML, 2019.



# TRADES

## Experimental results of TRADES

Table 5. Comparisons of TRADES with prior defense models under white-box attacks.

Defense	Defense type	Under which attack	Dataset	Distance	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$
Buckman et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 ( $\ell_\infty$ )	-	0%
Ma et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 ( $\ell_\infty$ )	-	5%
Dhillon et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 ( $\ell_\infty$ )	-	0%
Song et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 ( $\ell_\infty$ )	-	9%
Na et al. (2017)	gradient mask	Athalye et al. (2018)	CIFAR10	0.015 ( $\ell_\infty$ )	-	15%
Wong et al. (2018)	robust opt.	FGSM <sup>20</sup> (PGD)	CIFAR10	0.031 ( $\ell_\infty$ )	27.07%	23.54%
Madry et al. (2018)	robust opt.	FGSM <sup>20</sup> (PGD)	CIFAR10	0.031 ( $\ell_\infty$ )	87.30%	<b>47.04%</b>
Zheng et al. (2016)	regularization	FGSM <sup>20</sup> (PGD)	CIFAR10	0.031 ( $\ell_\infty$ )	94.64%	0.15%
Kurakin et al. (2017)	regularization	FGSM <sup>20</sup> (PGD)	CIFAR10	0.031 ( $\ell_\infty$ )	85.25%	45.89%
Ross & Doshi-Velez (2017)	regularization	FGSM <sup>20</sup> (PGD)	CIFAR10	0.031 ( $\ell_\infty$ )	95.34%	0%
TRADES ( $1/\lambda = 1.0$ )	regularization	FGSM <sup>20</sup> (PGD)	CIFAR10	0.031 ( $\ell_\infty$ )	88.64%	49.14%
TRADES ( $1/\lambda = 6.0$ )	regularization	FGSM <sup>20</sup> (PGD)	CIFAR10	0.031 ( $\ell_\infty$ )	84.92%	<b>56.61%</b>
TRADES ( $1/\lambda = 1.0$ )	regularization	DeepFool ( $\ell_\infty$ )	CIFAR10	0.031 ( $\ell_\infty$ )	88.64%	59.10%
TRADES ( $1/\lambda = 6.0$ )	regularization	DeepFool ( $\ell_\infty$ )	CIFAR10	0.031 ( $\ell_\infty$ )	84.92%	61.38%
TRADES ( $1/\lambda = 1.0$ )	regularization	LBFGSAttack	CIFAR10	0.031 ( $\ell_\infty$ )	88.64%	84.41%
TRADES ( $1/\lambda = 6.0$ )	regularization	LBFGSAttack	CIFAR10	0.031 ( $\ell_\infty$ )	84.92%	81.58%
TRADES ( $1/\lambda = 1.0$ )	regularization	MI-FGSM	CIFAR10	0.031 ( $\ell_\infty$ )	88.64%	51.26%
TRADES ( $1/\lambda = 6.0$ )	regularization	MI-FGSM	CIFAR10	0.031 ( $\ell_\infty$ )	84.92%	57.95%
TRADES ( $1/\lambda = 1.0$ )	regularization	C&W	CIFAR10	0.031 ( $\ell_\infty$ )	88.64%	84.03%
TRADES ( $1/\lambda = 6.0$ )	regularization	C&W	CIFAR10	0.031 ( $\ell_\infty$ )	84.92%	81.24%
Samangouei et al. (2018)	gradient mask	Athalye et al. (2018)	MNIST	0.005 ( $\ell_2$ )	-	55%
Madry et al. (2018)	robust opt.	FGSM <sup>40</sup> (PGD)	MNIST	0.3 ( $\ell_\infty$ )	99.36%	96.01%
TRADES ( $1/\lambda = 6.0$ )	regularization	FGSM <sup>40</sup> (PGD)	MNIST	0.3 ( $\ell_\infty$ )	99.48%	96.07%
TRADES ( $1/\lambda = 6.0$ )	regularization	C&W	MNIST	0.005 ( $\ell_2$ )	99.48%	99.46%

Zhang et al. "Theoretically principled trade-off between robustness and accuracy." ICML, 2019.



# TRADES vs VAT vs ALP

Trades: 
$$\min_{\theta} \mathbb{E}_{(x,y) \sim D} \left[ \underbrace{\mathcal{L}_{\text{CE}}(f(\mathbf{x}), y)}_{\text{提升准确率}} + \underbrace{\lambda \max_{\|\mathbf{r}\| \leq \epsilon} \mathcal{L}_{\text{KL}}(f(\mathbf{x}), f(\mathbf{x} + \mathbf{r}))}_{\text{提升鲁棒性}} \right]$$

Virtual Adversarial Training: 
$$\min_{\theta} \max_{\|\mathbf{x}_{\text{adv}} - \mathbf{x}\|_2 \leq \epsilon} \mathbb{E}_{(x,y) \in D} [\mathcal{L}_{\text{CE}}(f(\mathbf{x}), y) + \lambda \mathcal{L}_{\text{KL}}(f(\mathbf{x}), f(\mathbf{x}_{\text{adv}}))]$$

Adversarial Logits Pairing: 
$$\min_{\theta} \mathbb{E}_{(x,y) \in D} \left[ \max_{\|\mathbf{r}\| \leq \epsilon} \mathcal{L}_{\text{CE}}(f(\mathbf{x} + \mathbf{r}), y) + \lambda \|f(\mathbf{x} + \mathbf{r}) - f(\mathbf{x})\|_2^2 \right]$$

相似的优化框架，不同的损失选择，结果差异很大

Zhang et al. "Theoretically principled trade-off between robustness and accuracy." ICML, 2019.

Miyato et al. Distributional smoothing with virtual adversarial training. ICLR 2016.

Kannan, Harini, Alexey Kurakin, and Ian Goodfellow. "Adversarial logit pairing." *arXiv preprint arXiv:1803.06373* (2018).



# MART: Misclassification Aware adversarial Training

## Min-max Adversarial Training:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n \max_{\|x_i - x_i^0\| \leq \epsilon} \mathcal{L}(f_{\theta}(x_i, ) y_i)$$

where,  $x_i^0$  is a natural (clean) training sample,  $y_i$  is the label of  $x_i^0$ .

Adversarial examples are only defined on **correctly classified** examples,  
what about **misclassified examples** ?

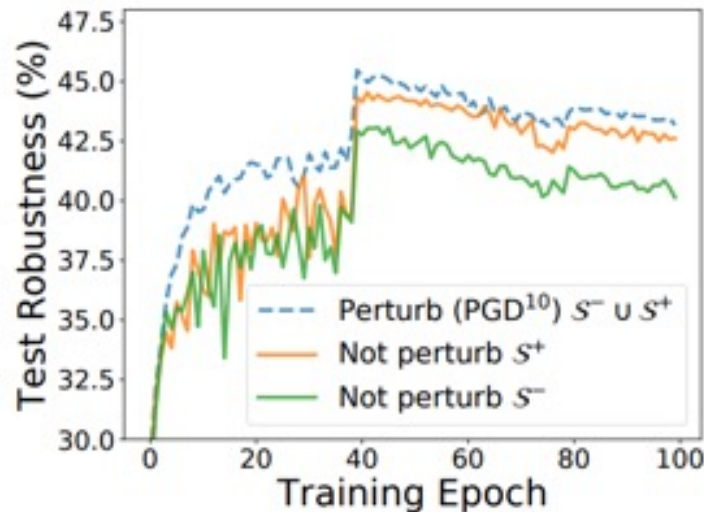
Wang, et al. "Improving adversarial robustness requires revisiting misclassified examples." ICLR, 2019.



# MART: Misclassification Aware adveRsarial Training

## The influence of misclassified and correctly classified examples:

- A pre-trained network to select the **same size (13%)**
  - Subset of misclassified examples  $S^-$
  - Subset of correctly classified examples  $S^+$



**Misclassified examples** have a **significant impact** on the final robustness!

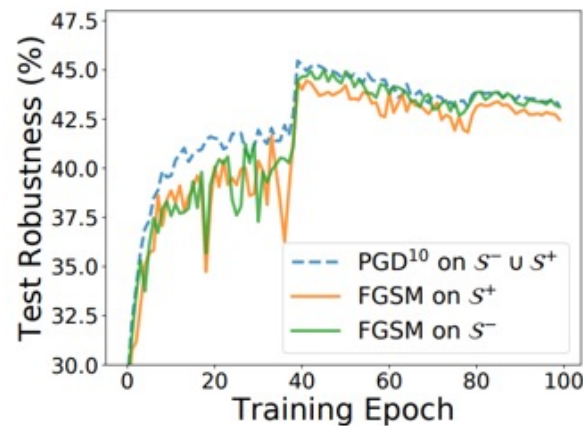
Wang, et al. "Improving adversarial robustness requires revisiting misclassified examples." ICLR, 2019.



# MART: Misclassification Aware adveRsarial Training

□ For **inner maximization** process:

- Weak attack on misclassified examples  $S^-$
- Weak attack on correctly classified examples  $S^+$

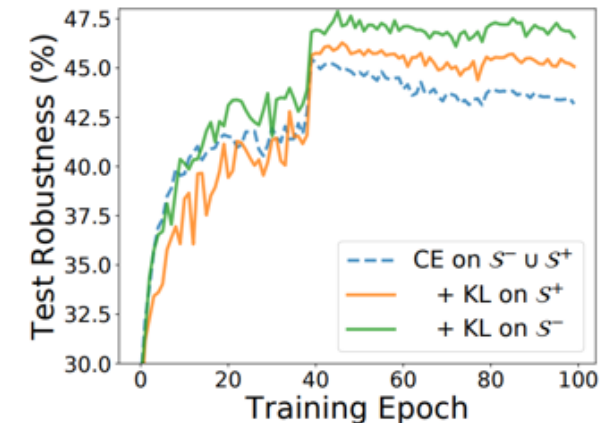


(b) Inner maximization

different maximization techniques have  
**negligible** effect

□ For **outer minimization** process:

- Regularization on misclassified examples  $S^-$
- Regularization on correctly classified examples  $S^+$



(c) Outer minimization

different minimization techniques have  
**significant** effect

Wang, et al. "Improving adversarial robustness requires revisiting misclassified examples." ICLR, 2019.



# MART: Misclassification Aware adversarial Training

## Misclassification aware adversarial risk:

- Adversarial risk:

$$\mathcal{R}(h_{\theta}) = \frac{1}{n} \sum_{i=1}^n \max_{\mathbf{x}'_i \in \mathcal{B}_{\epsilon}(\mathbf{x}_i)} \mathbb{1}(h_{\theta}(\mathbf{x}'_i) \neq y_i),$$

- Correctly classified and misclassified example:

$$\mathcal{S}_{h_{\theta}}^+ = \{i : i \in [n], h_{\theta}(\mathbf{x}_i) = y_i\} \quad \text{and} \quad \mathcal{S}_{h_{\theta}}^- = \{i : i \in [n], h_{\theta}(\mathbf{x}_i) \neq y_i\}$$

- Misclassification aware adversarial risk:

$$\begin{aligned} \min_{\theta} \mathcal{R}_{\text{misc}}(h_{\theta}) &:= \frac{1}{n} \left( \sum_{i \in \mathcal{S}_{h_{\theta}}^+} \mathcal{R}^+(h_{\theta}, \mathbf{x}_i) + \sum_{i \in \mathcal{S}_{h_{\theta}}^-} \mathcal{R}^-(h_{\theta}, \mathbf{x}_i) \right) \\ &= \frac{1}{n} \sum_{i=1}^n \left\{ \mathbb{1}(h_{\theta}(\hat{\mathbf{x}}'_i) \neq y_i) + \mathbb{1}(h_{\theta}(\mathbf{x}_i) \neq h_{\theta}(\hat{\mathbf{x}}'_i)) \cdot \mathbb{1}(h_{\theta}(\mathbf{x}_i) \neq y_i) \right\} \end{aligned}$$





# MART: Misclassification Aware adversarial Training

- Surrogate loss functions (existing methods and MART):

Defense Method	Loss Function
<i>Standard</i>	$\text{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y)$
ALP	$\text{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) + \lambda \cdot \ \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}) - \mathbf{p}(\mathbf{x}, \boldsymbol{\theta})\ _2^2$
CLP	$\text{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) + \lambda \cdot \ \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}) - \mathbf{p}(\mathbf{x}, \boldsymbol{\theta})\ _2^2$
TRADES	$\text{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) + \lambda \cdot \text{KL}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}) \parallel \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}))$
MMA	$\text{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) \cdot \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}) = y) + \text{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) \cdot \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}) \neq y)$
<b>MART</b>	$\text{BCE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) + \lambda \cdot \text{KL}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}) \parallel \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta})) \cdot (1 - \mathbf{p}_y(\mathbf{x}, \boldsymbol{\theta}))$

- Semi-supervised extension of MART:  $\text{BCE}(\mathbf{p}(\hat{\mathbf{x}}'_i, \boldsymbol{\theta}), y_i) = -\log(\mathbf{p}_{y_i}(\hat{\mathbf{x}}'_i, \boldsymbol{\theta})) - \log(1 - \max_{k \neq y_i} \mathbf{p}_k(\hat{\mathbf{x}}'_i, \boldsymbol{\theta}))$

$$\mathcal{L}_{\text{semi}}^{\text{MART}}(\boldsymbol{\theta}) = \sum_{i \in \mathcal{S}_{\text{sup}}} \ell_{\text{sup}}^{\text{MART}}(\mathbf{x}_i, y_i; \boldsymbol{\theta}) + \gamma \cdot \sum_{i \in \mathcal{S}_{\text{unsup}}} \ell_{\text{unsup}}^{\text{MART}}(\mathbf{x}_i, y_i; \boldsymbol{\theta})$$



# MART: Misclassification Aware adversarial Training

## Robustness of MART:

- White-box robustness: **ResNet-18**, CIFAR-10,  $\epsilon=8/255$

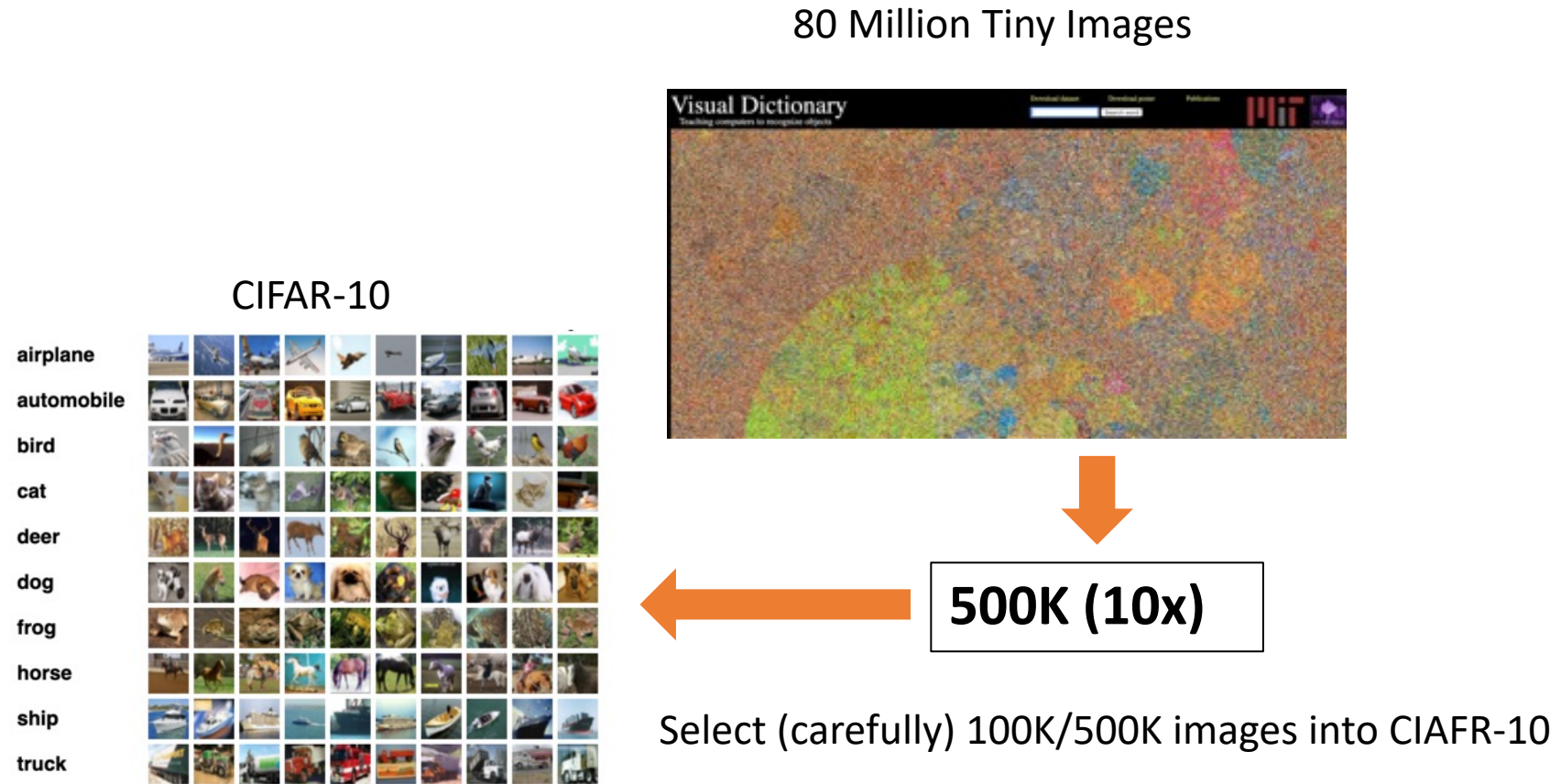
Defense	MNIST				CIFAR-10			
	Natural	FGSM	PGD <sup>20</sup>	CW <sub>∞</sub>	Natural	FGSM	PGD <sup>20</sup>	CW <sub>∞</sub>
<i>Standard</i>	99.11	97.17	94.62	94.25	84.44	61.89	47.55	45.98
MMA	98.92	97.25	95.25	94.77	<b>84.76</b>	62.08	48.33	45.77
Dynamic	98.96	97.34	95.27	94.85	83.33	62.47	49.40	46.94
TRADES	<b>99.25</b>	96.67	94.58	94.03	82.90	62.82	50.25	48.29
<b>MART</b>	98.74	<b>97.87</b>	<b>96.48</b>	<b>96.10</b>	83.07	<b>65.65</b>	<b>55.57</b>	<b>54.87</b>

- White-box robustness: **WideResNet-34-10**, CIFAR-10,  $\epsilon=8/255$

Defense	Natural	FGSM		PGD <sup>20</sup>		PGD <sup>100</sup>		CW <sub>∞</sub>	
		Best	Last	Best	Last	Best	Last	Best	Last
<i>Standard</i>	<b>87.30</b>	56.10	56.10	52.68	49.31	51.55	49.03	50.73	48.47
Dynamic	84.51	63.53	63.53	55.03	51.70	54.12	50.07	51.34	49.27
TRADES	84.22	64.70	64.70	56.40	53.16	55.68	51.27	51.98	51.12
<b>MART</b>	84.17	<b>67.51</b>	<b>67.51</b>	<b>58.56</b>	<b>57.39</b>	<b>57.88</b>	<b>55.04</b>	<b>54.58</b>	<b>54.53</b>



# Using More Data to Improve Robustness



Alayrac et al. “Are labels required for improving adversarial robustness?.” *NeurIPS*, 2019.

Carmon et al. “Unlabeled data improves adversarial robustness.” *NeurIPS* 2019

# UAT & RST

## Unsupervised Adversarial Training (UAT):

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{sup}}(\theta) + \lambda \boxed{\mathcal{L}_{\text{unsup}}(\theta)} \longrightarrow \left\{ \begin{array}{l} \mathcal{L}_{\text{unsup}}^{OT}(\theta) = \mathbb{E}_{x \sim P(X)} \sup_{x' \in \mathcal{N}_\epsilon(x)} \mathcal{D}(p_{\hat{\theta}}(\cdot|x), p_{\theta}(\cdot|x')) \\ \mathcal{L}_{\text{unsup}}^{FT}(\theta) = \mathbb{E}_{x \sim P(X)} \sup_{x' \in \mathcal{N}_\epsilon(x)} \text{xent}(\hat{y}(x), p_{\theta}(\cdot|x')) \end{array} \right.$$

## Robust Self-Training (RST):

$$\text{minimizing } \sum_{i=1}^n L_{\text{robust}}(\theta, x_i, y_i) + w \sum_{i=1}^{\tilde{n}} L_{\text{robust}}(\theta, \boxed{\tilde{x}_i, \tilde{y}_i}) \longrightarrow \left\{ \begin{array}{l} \tilde{x}_i : \text{new samples} \\ \tilde{y}_i : \text{pseudo-labels} \end{array} \right.$$

Alayrac et al. "Are labels required for improving adversarial robustness?." *NeurIPS*, 2019.

Carmon et al. "Unlabeled data improves adversarial robustness." *NeurIPS* 2019



# State-of-the-art: AT Methods

Leaderboard: CIFAR-10,  $\ell_\infty = 8/255$ , untargeted attack

Show 15 entries

Search: Papers, architectures, ve

Rank	Method	Standard accuracy	AutoAttack robust accuracy	Best known robust accuracy	AA eval. potentially unreliable	Extra data	Architecture	Venue
1	<a href="#">Fixing Data Augmentation to Improve Adversarial Robustness</a> 66.56% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)	92.23%	66.58%	66.56%	×	☑	WideResNet-70-16	arXiv, Mar 2021
2	<a href="#">Improving Robustness using Generated Data</a> It uses additional 100M synthetic images in training. 66.10% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)	88.74%	66.11%	66.10%	×	×	WideResNet-70-16	NeurIPS 2021
3	<a href="#">Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples</a> 65.87% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)	91.10%	65.88%	65.87%	×	☑	WideResNet-70-16	arXiv, Oct 2020
4	<a href="#">Fixing Data Augmentation to Improve Adversarial Robustness</a> It uses additional 1M synthetic images in training. 64.58% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)	88.50%	64.64%	64.58%	×	×	WideResNet-106-16	arXiv, Mar 2021
5	<a href="#">Stable Neural ODE with Lyapunov-Stable Equilibrium Points for Defending Against Adversarial Attacks</a> Based on the model Rebuff2021Fixing_70_16_cutmix_extra. 64.20% robust accuracy is due to AutoAttack + transfer APGD from Rebuff2021Fixing_70_16_cutmix_extra	93.73%	71.28%	64.20%	☑	☑	WideResNet-70-16, Neural ODE block	NeurIPS 2021

数据，数据，还是数据

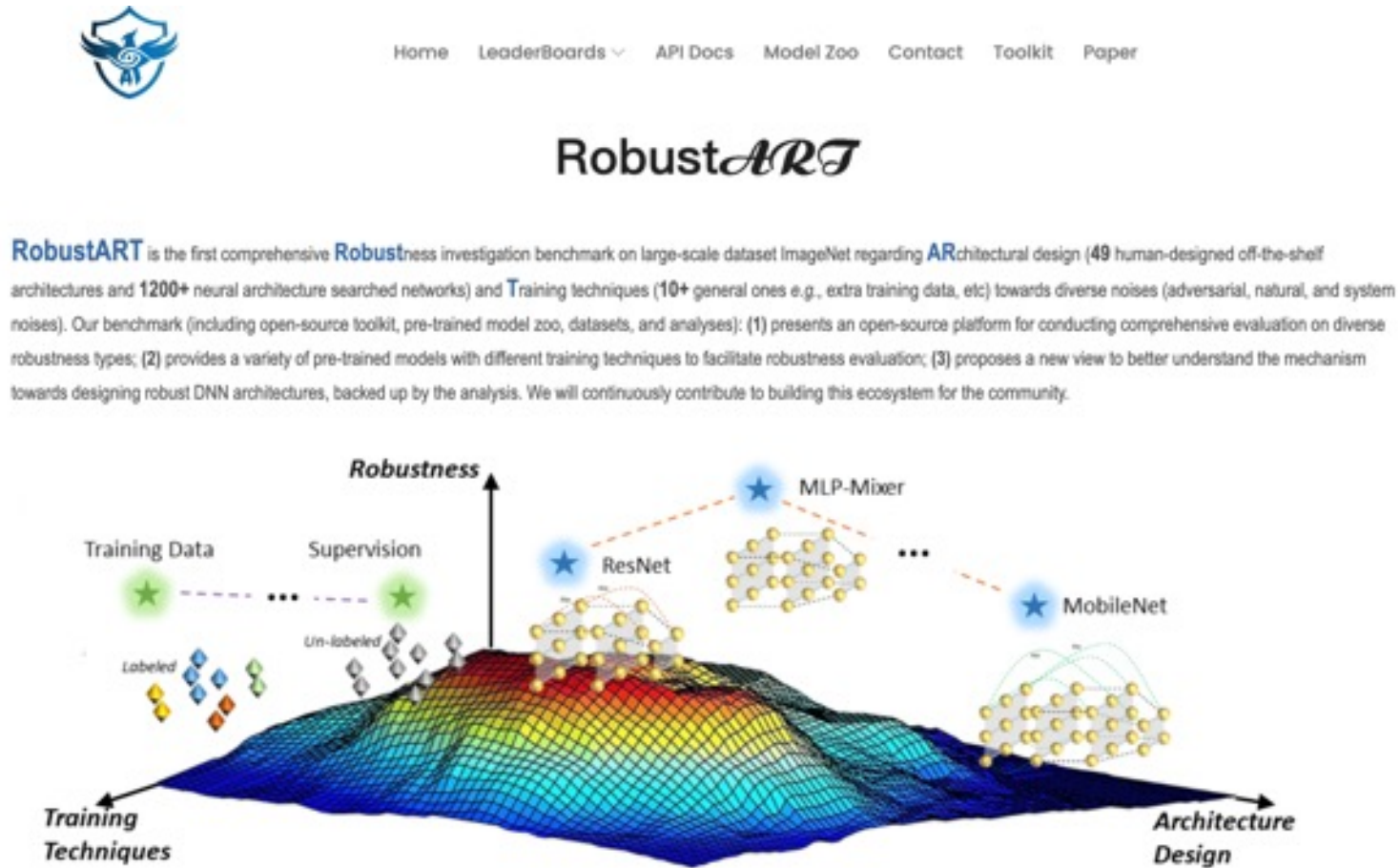
- 数据增广
- 数据生成

<https://robustbench.github.io/>





# State-of-the-art: DNN Architecture

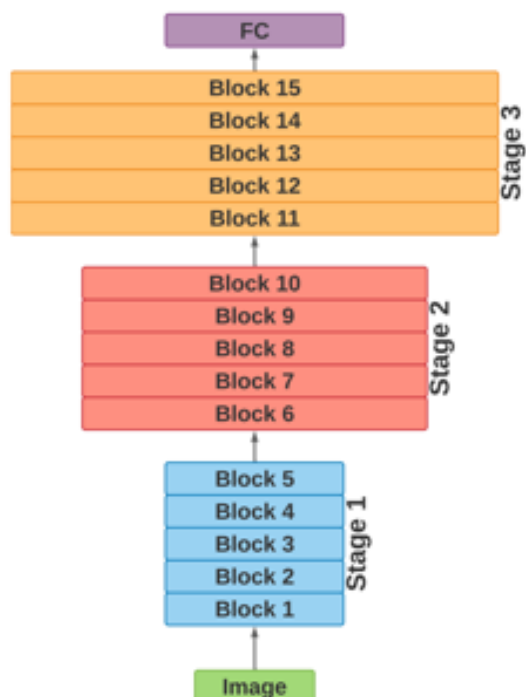


<http://robust.art/> Tang et al. "RobustART: Benchmarking robustness on architecture design and training techniques." *arXiv:2109.05211*, 2021.

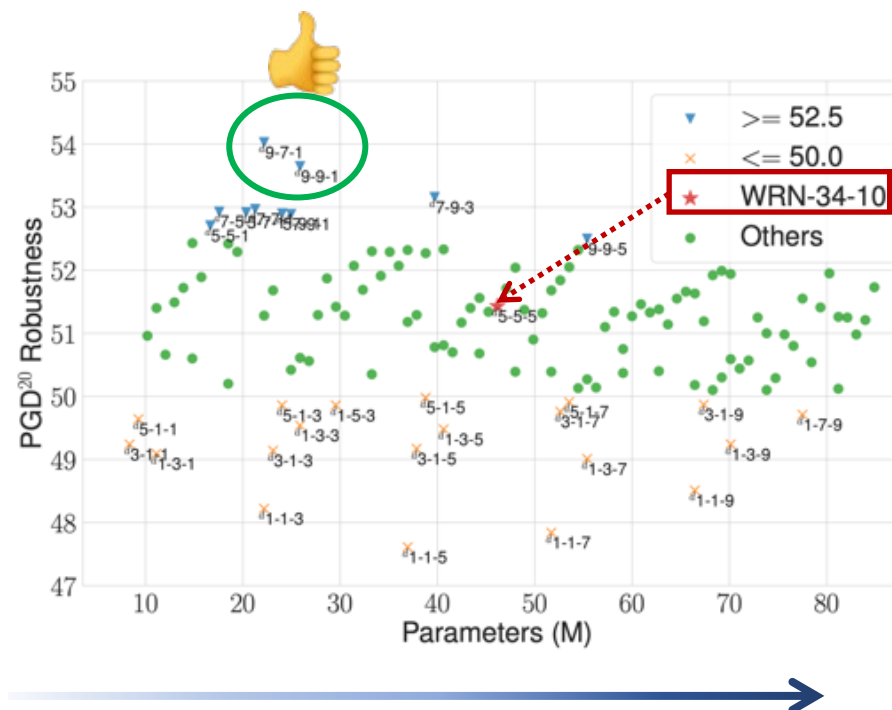


# State-of-the-art: DNN Architecture

Reduce the deep layers can improve robustness



WideResNet-34-10



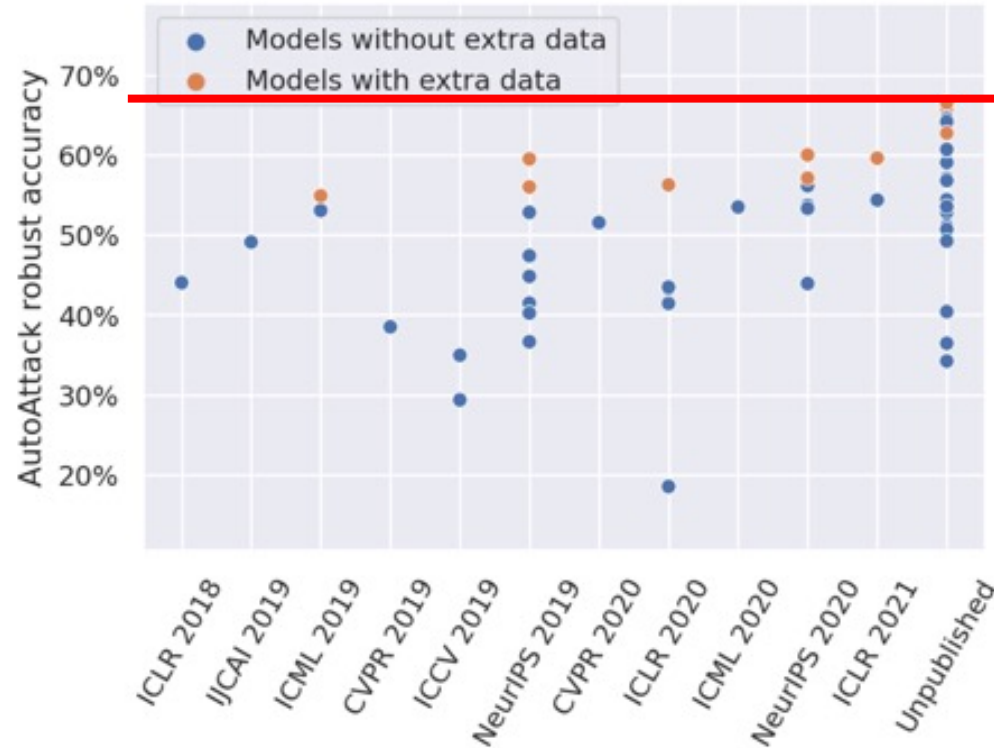
Depth Grid Search

d: depth  
d5-5-5: depth=5  
at all 3 stages

Huang et al. "Exploring architectural ingredients of adversarially robust deep neural networks." NeurIPS 2021.



# State-of-the-art: AT Methods



Clean accuracy: 94% vs 66% (robustness)

DNN: WideResNet-70-16; Dataset: CIFAR-10

Perturbation:  $\epsilon = 8/255$ ;

Evaluation attack: AutoAttack

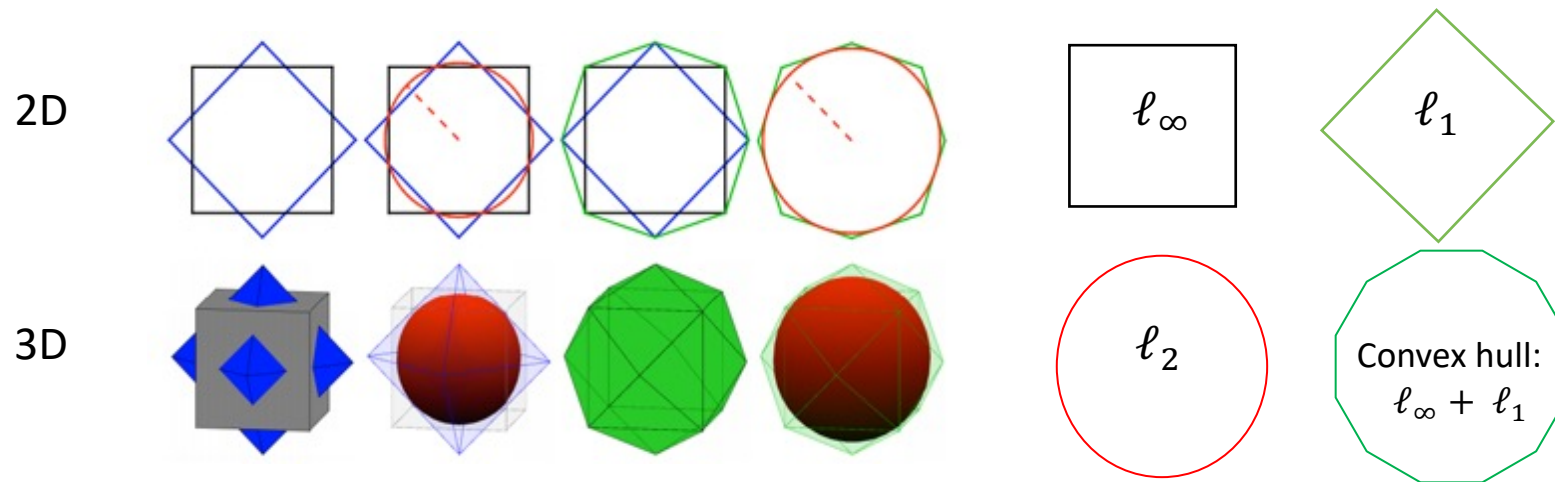
<https://robustbench.github.io/>





# Certified Defense vs Empirical Defense

- **Certified robustness** (Sinha et al. 2018; Cohen et al. 2018; Lee et al. 2019)
  - Gaussian randomized smoothing -> robustness with the  $\ell_2$  norm ball
  - Laplacian randomized smoothing ->  $\ell_1$  robustness
  - Uniform randomized smoothing ->  $\ell_0$  robustness
  - **Pros:** robustness guarantees, 严格的鲁棒性下界证明
  - **Cons:** 1) deep networks are hard to certify, 2) guarantees are loose, 3) need to train the model in certain ways



关键点：如何在神经网络里传播边界？

Li et al. Sok: Certified robustness for deep neural networks, S&P, 2023



# 尚存挑战

- 防御更多样化的攻击，尤其是物理攻击
- 防御大模型、多模态预训练模型
- 对抗训练在不同应用领域的自适应
- 降低防御代价：对抗训练加速、防止自然准确率下降
- 需要组合防御：检测+对抗训练
- 在大规模数据集（如ImageNet）上的对抗训练



# C U Next Week!

**Course page:**

<https://trustworthymachinelearning.github.io/>

**Textbook:**

下载链接: [https://pan.baidu.com/s/1kybxud\\_tz0xshWpMEORAhg?pwd=tauu](https://pan.baidu.com/s/1kybxud_tz0xshWpMEORAhg?pwd=tauu)

Email: [xingjunma@fudan.edu.cn](mailto:xingjunma@fudan.edu.cn)

Personal page: [www.xingjunma.com](http://www.xingjunma.com)

Office: 江湾校区交叉二号楼D5025

