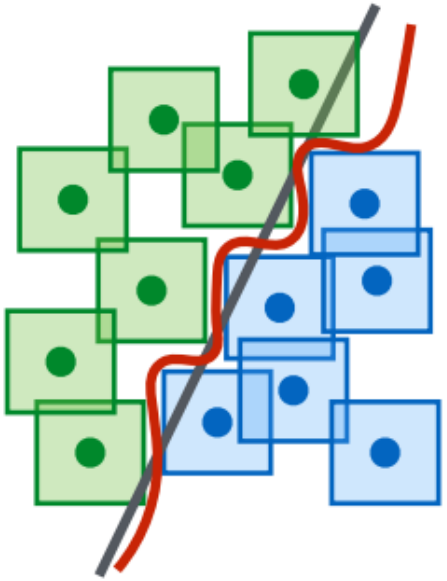


Adversarial Defense

马兴军，复旦大学 计算机学院

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)

Adversarial Attack Competition

RESULTS							
#	User	Entries	Date of Last Entry	Score ▲	Efficiency Score ▲	Error Rate ▲	Detailed Results
1	abcdhhhh	8	10/10/23	0.7042 (3)	0.9901 (1)	0.4183 (5)	View
1	alex_z	2	10/10/23	0.7004 (4)	0.9901 (1)	0.4108 (6)	View
1	siyuandu	2	10/09/23	0.7004 (4)	0.9901 (1)	0.4108 (6)	View
1	xieyong	3	10/06/23	0.7004 (4)	0.9901 (1)	0.4108 (6)	View
1	YiY	1	10/02/23	0.7004 (4)	0.9901 (1)	0.4108 (6)	View
1	jxzhou	7	10/09/23	0.5607 (8)	0.9901 (1)	0.1314 (8)	View
2	wnllixiao	4	10/11/23	0.7077 (1)	0.9802 (2)	0.4353 (4)	View
2	tdlhl	10	10/09/23	0.7077 (1)	0.9802 (2)	0.4353 (4)	View
2	starch	4	10/08/23	0.7077 (1)	0.9802 (2)	0.4353 (4)	View
2	archen	7	10/09/23	0.6516 (6)	0.9802 (2)	0.3231 (7)	View
3	shuyang_jiang	5	10/09/23	0.7069 (2)	0.9703 (3)	0.4436 (3)	View
4	LiGuanyu	2	10/11/23	0.6779 (5)	0.9010 (4)	0.4548 (2)	View
4	hanxunh	1	10/01/23	0.6779 (5)	0.9010 (4)	0.4548 (2)	View
5	X.RW	11	10/08/23	0.6252 (7)	0.7540 (5)	0.4964 (1)	View

Link: https://codalab.lisn.upsaclay.fr/competitions/15669?secret_key=77cb8986-d5bd-4009-82f0-7dde2e819ff8

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
- Later Adversarial Training Methods
- Remaining Challenges and Recent Progress

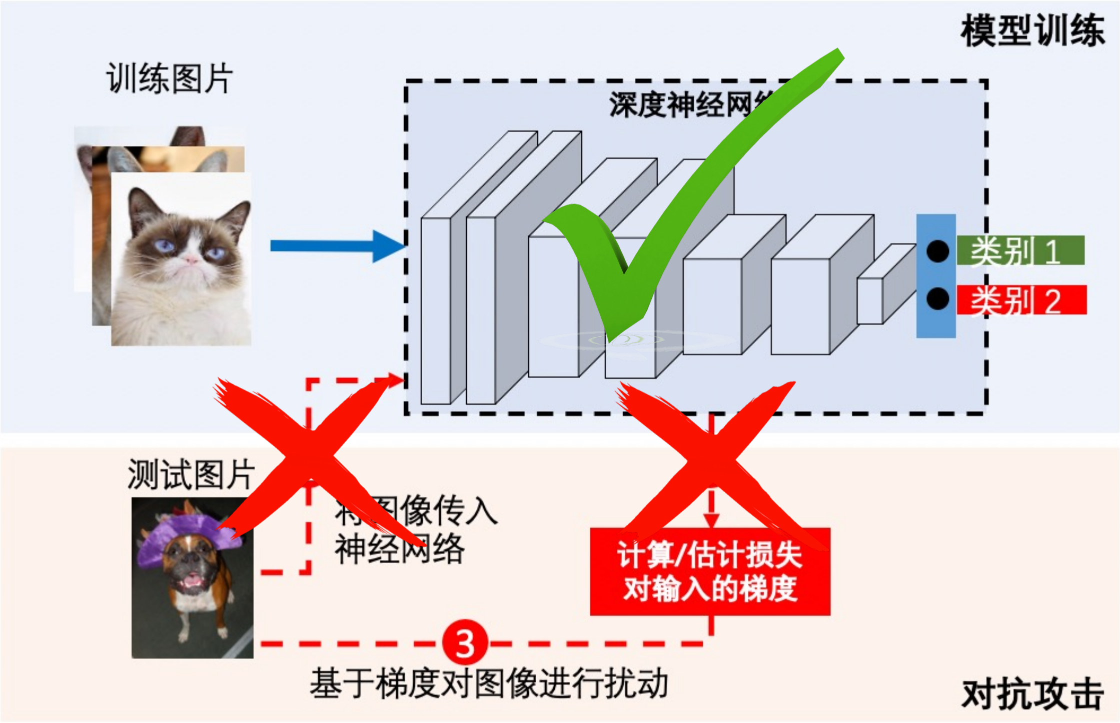
A Recap of the Timeline

- 2013年Biggio等人与Szegedy等人发现对抗样本
- 2014年Goodfellow等人提出快速单步攻击FGSM及对抗训练
- 2015年简单检测方法（PCA）和对抗训练方法
- 2016年提出对抗训练的min-max优化框架
- 2017年大量的对抗样本检测方法和攻击方法（BIM、C&W）、**10种检测方法被攻破**
- 2018年物理世界攻击方法、检测方法升级、PGD攻击与对抗训练、9种防御方法被攻破
- 2019年TRADES及大量其他对抗训练方法、第一篇Science文章
- 2020年AutoAttack攻击、Fast对抗训练
- 2021年增大模型、增加数据的对抗训练、领域延伸
- 2022年尚未解决的问题，攻击越来越多，防御越来越难

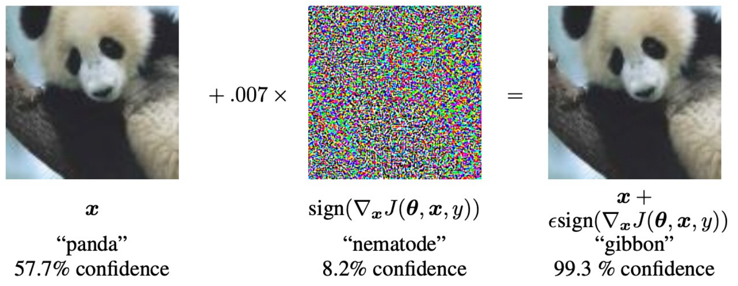
Principles of Defense

- Block the attack (**捨头去尾**)
  - 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: $\underbrace{\max_{x'} L(f_\theta(x'), y)}_{\text{分类错误}}$ $\underbrace{\text{subject to } \|x' - x\|_p \le \epsilon}_{\text{扰动很小}}$ $\underbrace{\text{for } x \in D_{test}}_{\text{测试阶段攻击}}$

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

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:

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

- Measurement of adversarial robustness:

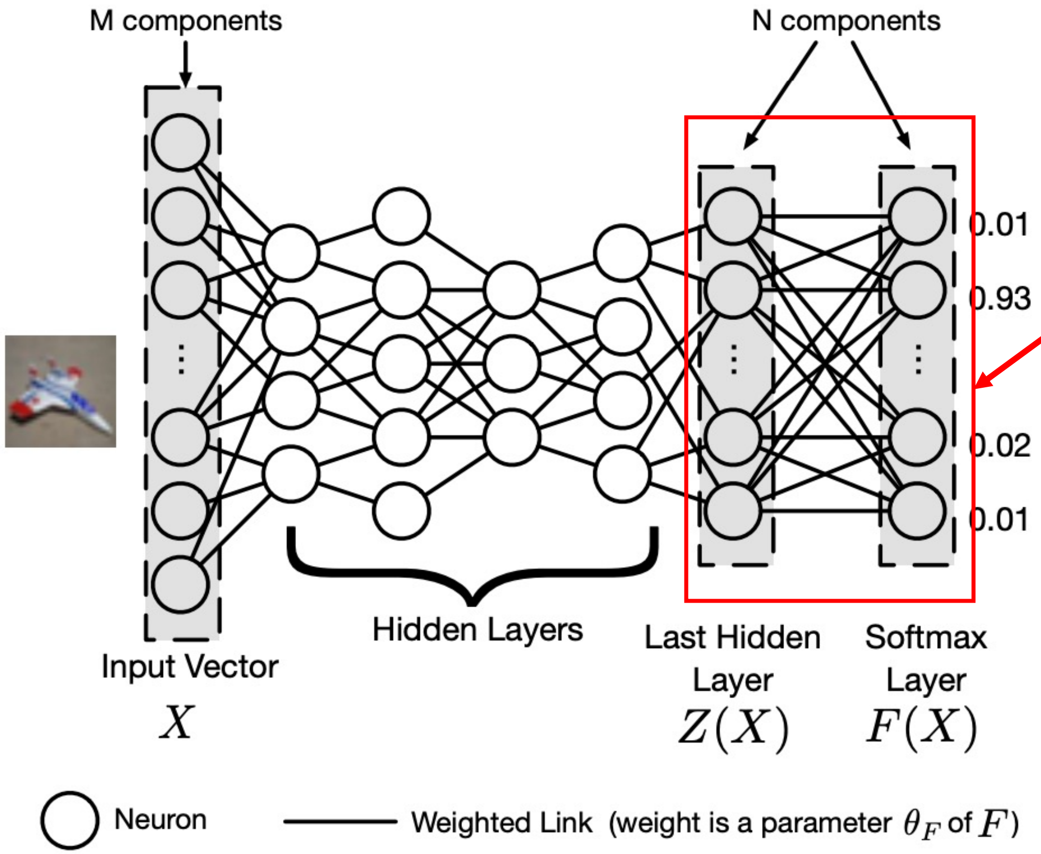
$$\textbf{robustness}\ \text{(robust accuracy)} = \frac{\text{correctly classified } \textbf{advs} \text{ samples}}{\text{total } \textbf{advs} \text{ samples}}$$

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

Defense Methods

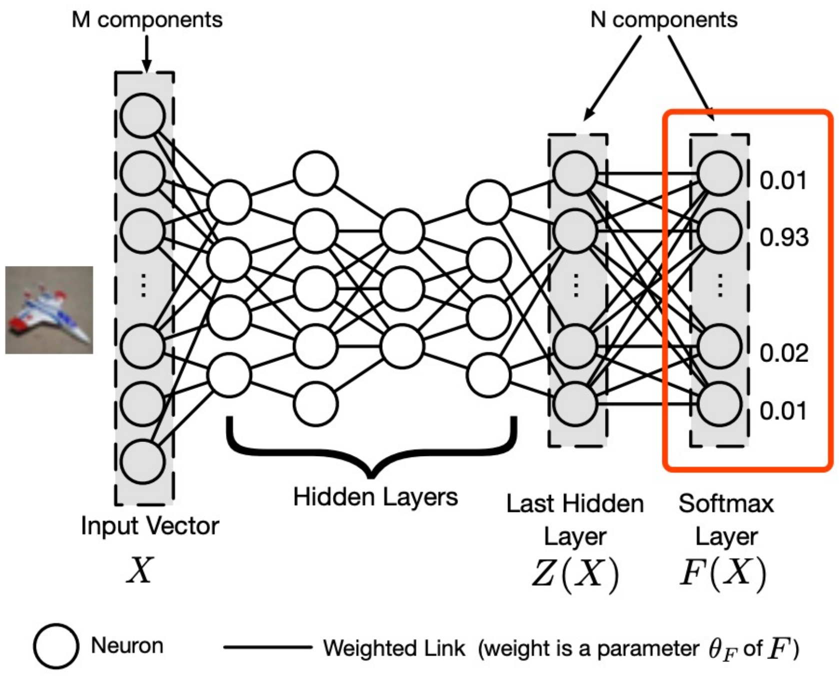
- Early Defense Methods
- Early Adversarial Training Methods
- Advanced Adversarial Training Methods
- Remaining Challenges and Recent Progress

Defensive Distillation



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

Defensive Distillation



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

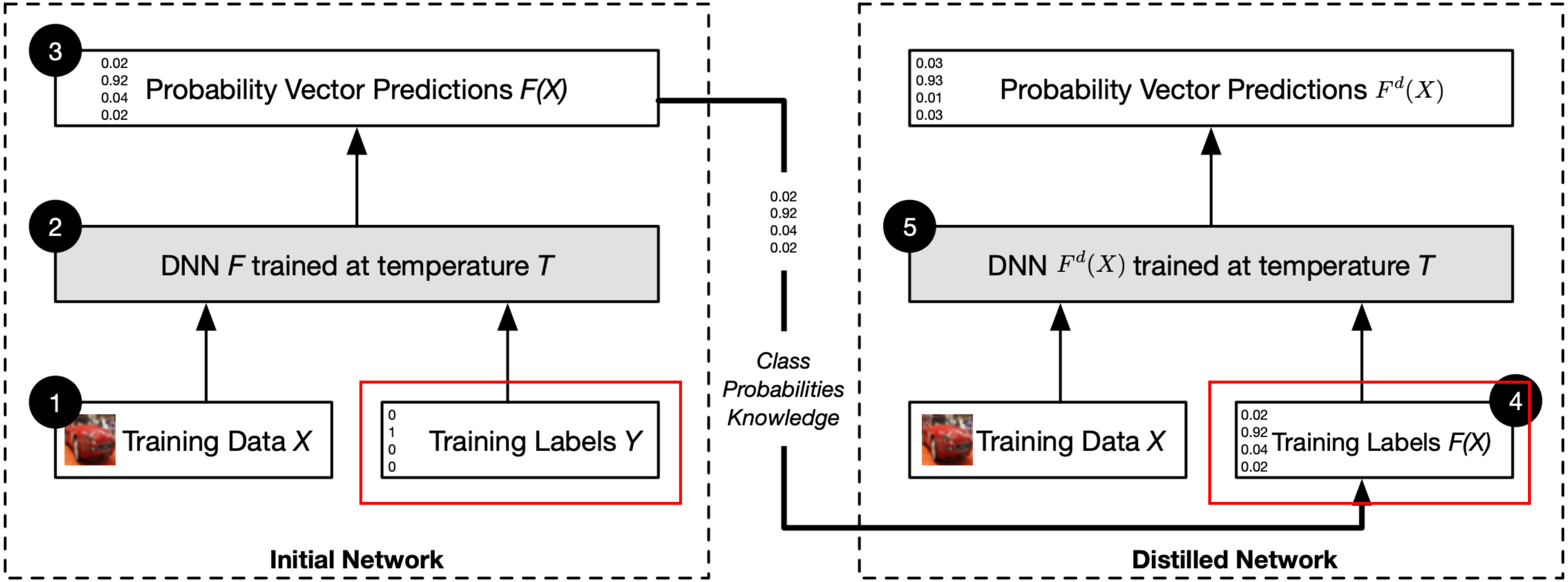
1

Distillation with temperature T

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

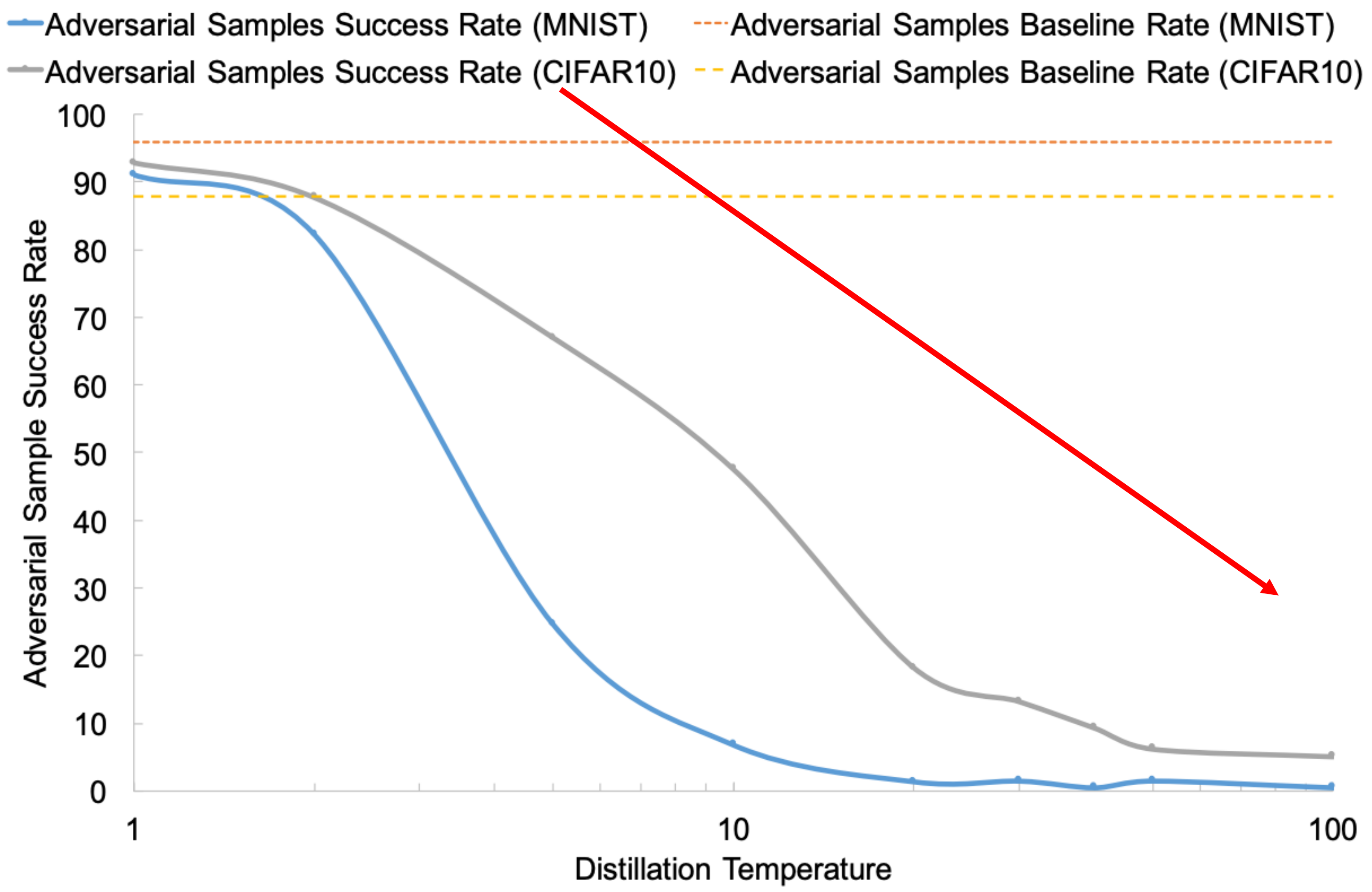
2

Defensive Distillation



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

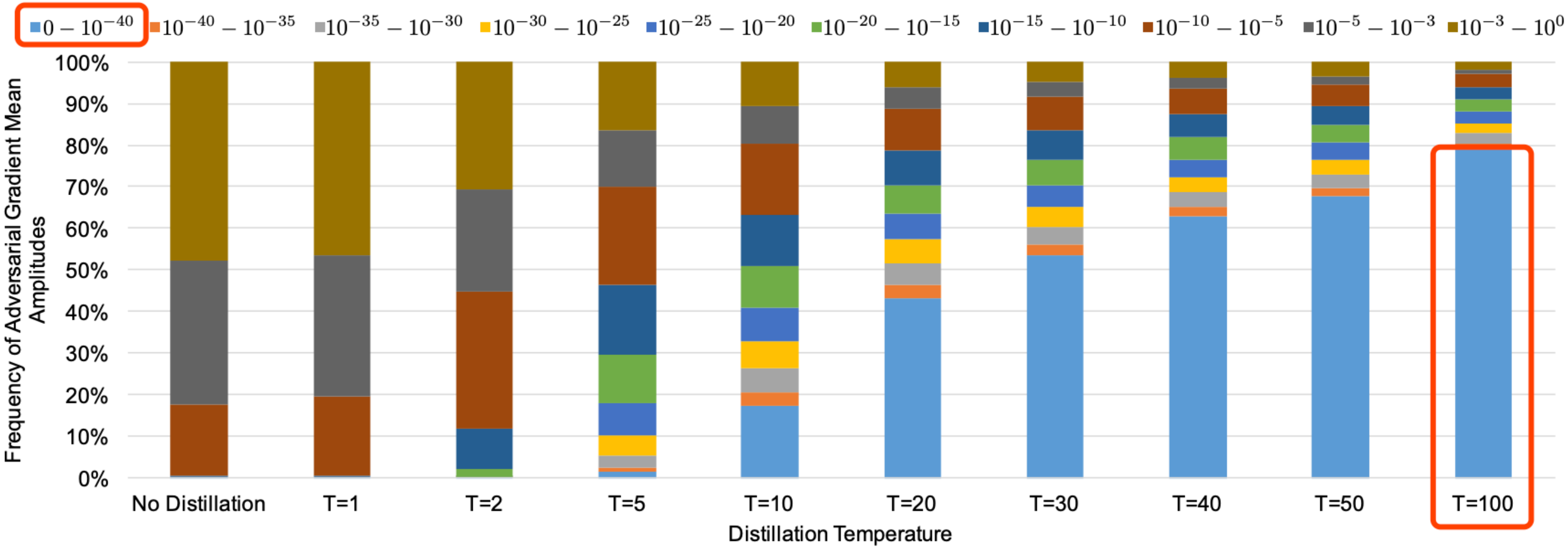
Defensive Distillation



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!

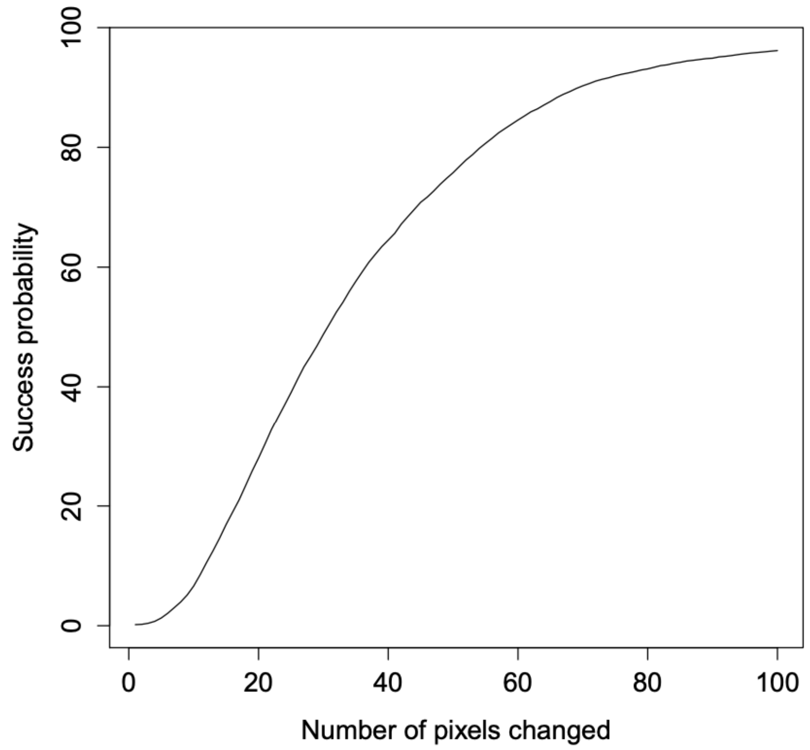


Defensive Distillation Is Not Robust

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

$$x' = x + \varepsilon \cdot \operatorname{sign} \nabla_x L(\widehat{f_\theta}(x), y)$$

$$\widehat{f_\theta}(x) = \mathrm{softmax}(\mathrm{z(x)/T})$$



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

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

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

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

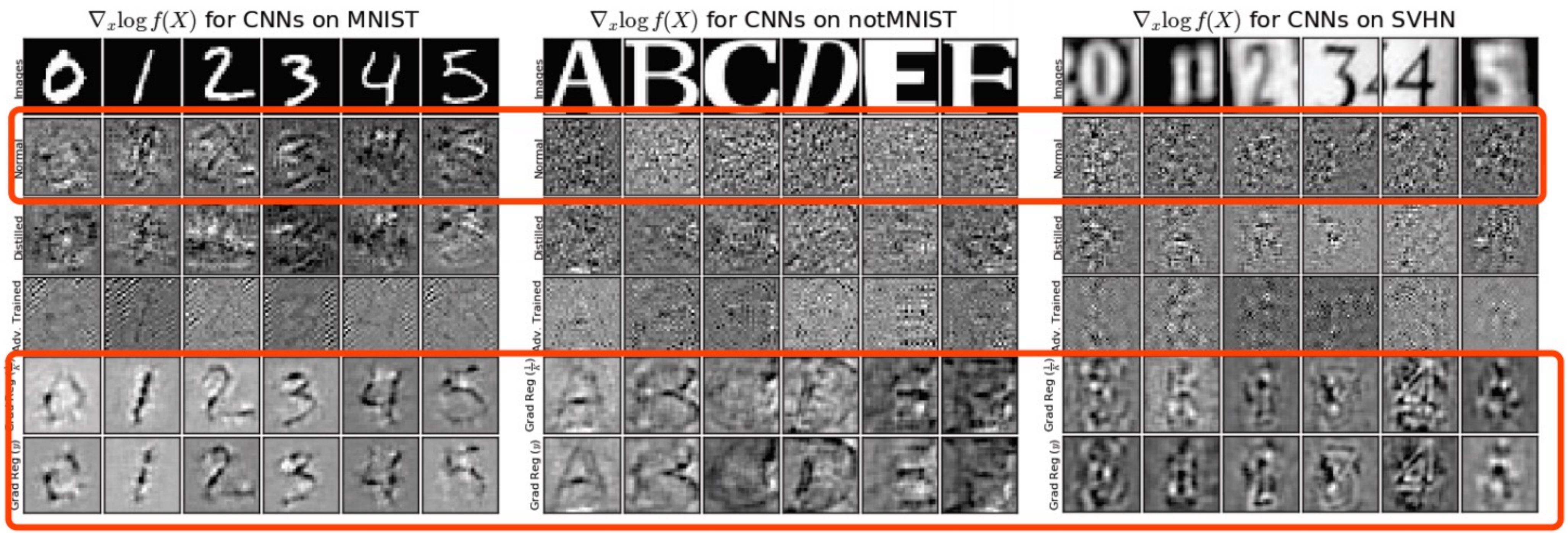
Ross et al. "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients." *AAAI*, 2018.
Drucker, Harris, and Yann Le Cun. "Improving generalization performance using double backpropagation." *TNN,1992*.

Input Gradients Regularization

蒸馏的

对抗训练的

正则化的

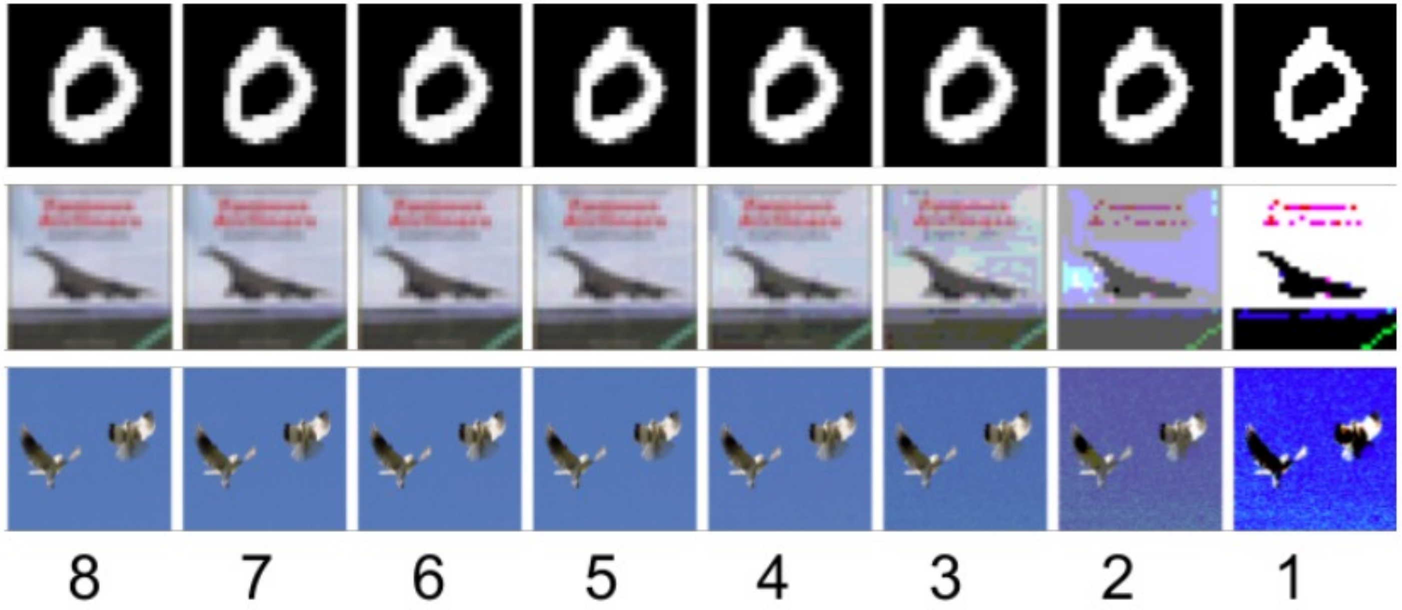


Issues: 1) limited adversarial robustness, 2) hurts learning

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

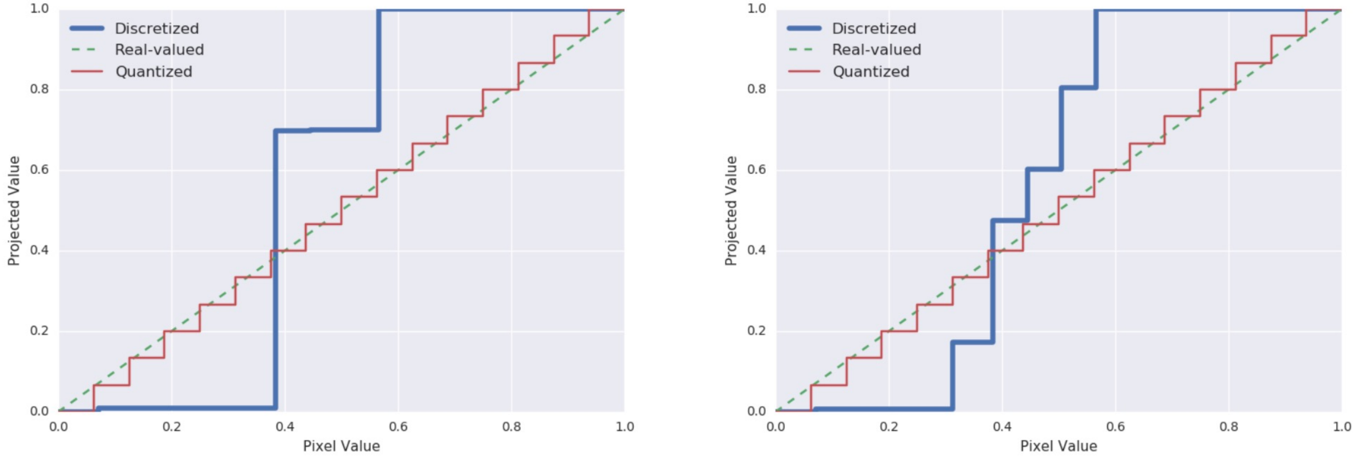


It also hurts performance on large-scale image datasets.

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

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

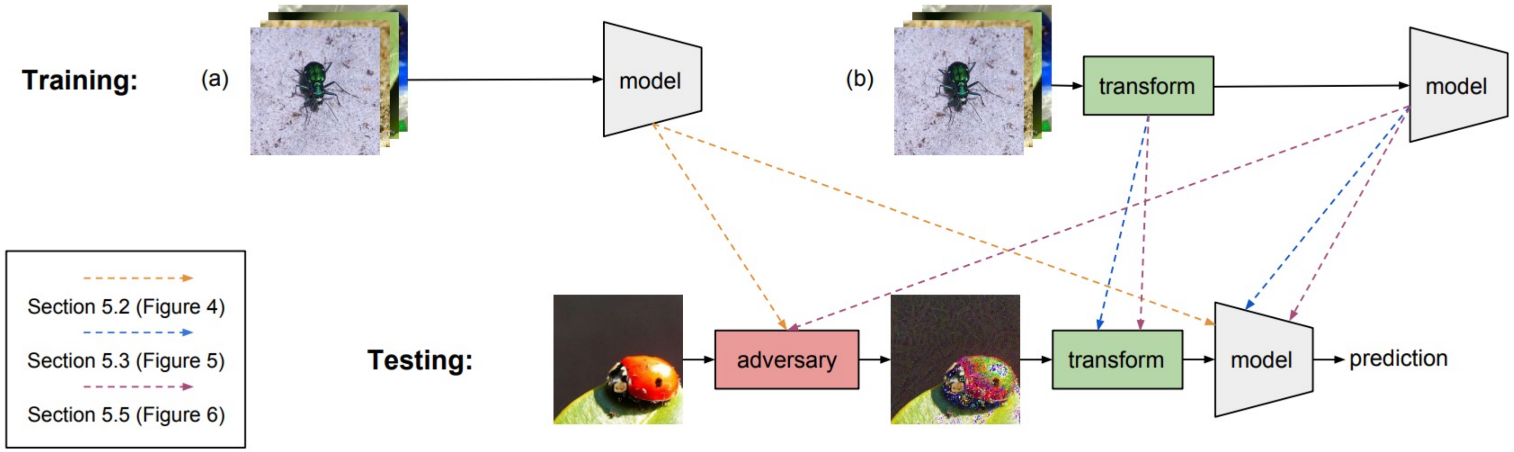
$$\tau(0.66) = 1111110000$$

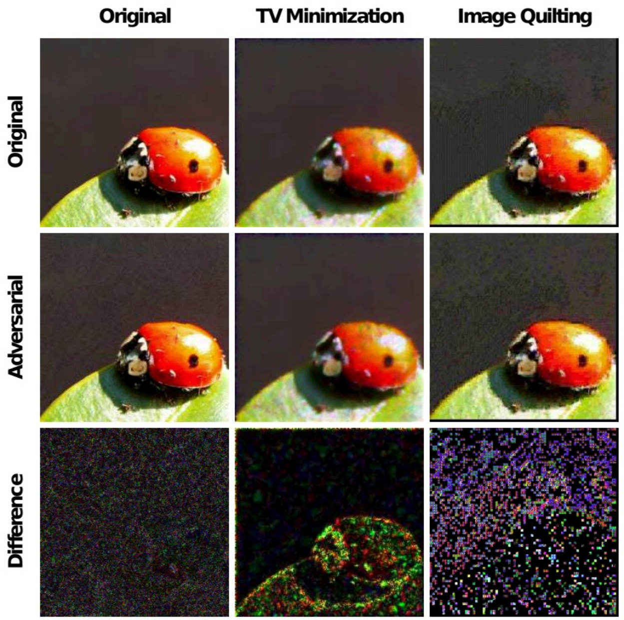
Proposed Thermometer Encoding

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





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

$$\begin{aligned} \underset{x'}{\arg\max} \quad & \mathbb{E}_{t\sim T}[\log P(y_t|t(x'))] \\ \text{subject to} \quad & \mathbb{E}_{t\sim T}[d(t(x'), t(x))] < \epsilon \\ & x \in [0,1]^d \end{aligned}$$

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 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)	0%*
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)	0%*
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)	9%*
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)	55%**
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	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.

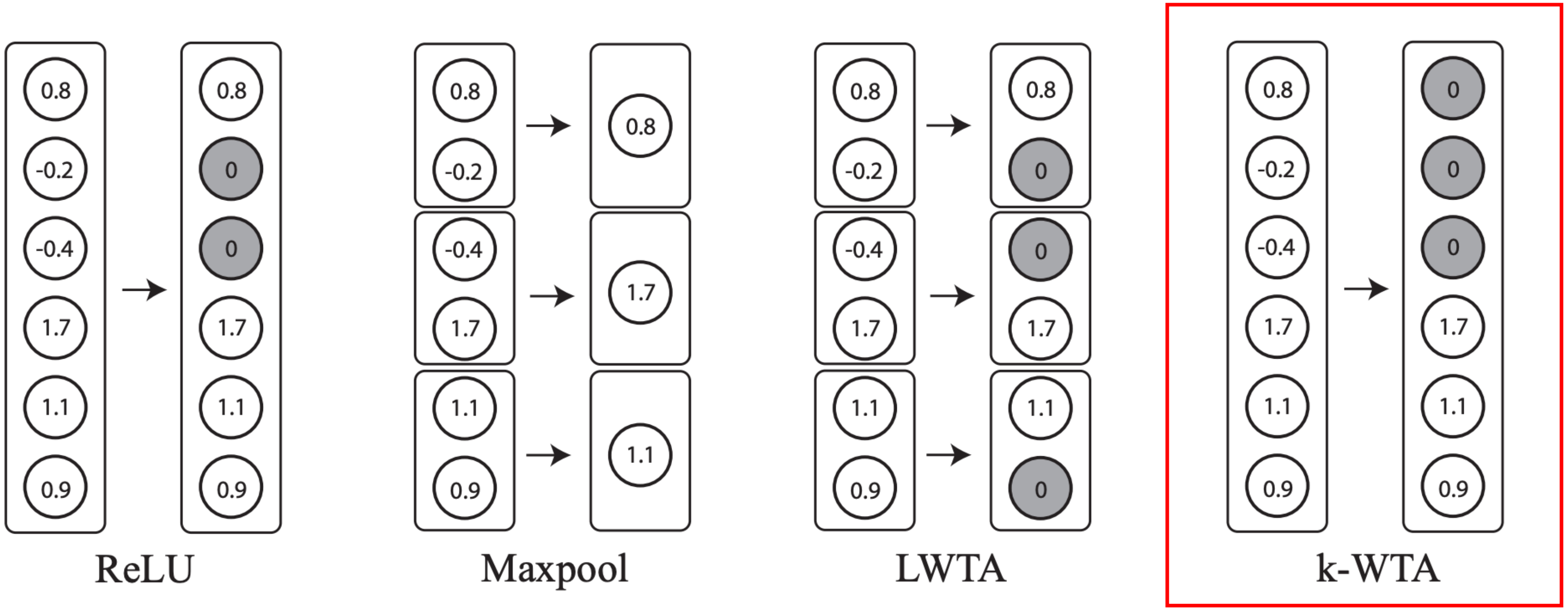
How to Properly Evaluate a Defense?

- ✓ 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

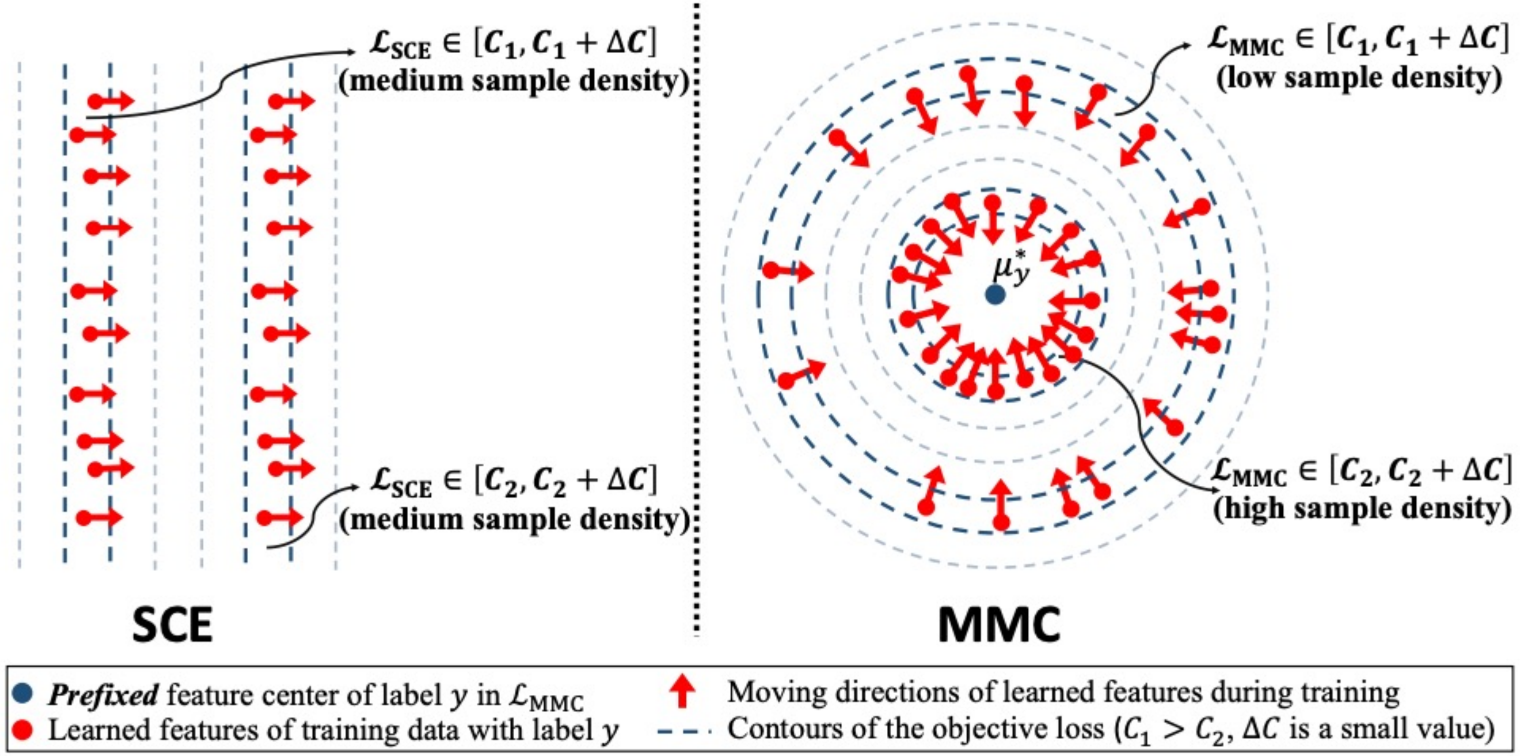


k-Winners-Take-All (k-WTA) activation

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

Pang et al. Rethinking softmax cross-entropy loss for adversarial robustness. ICLR, 2020.

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 λ; 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

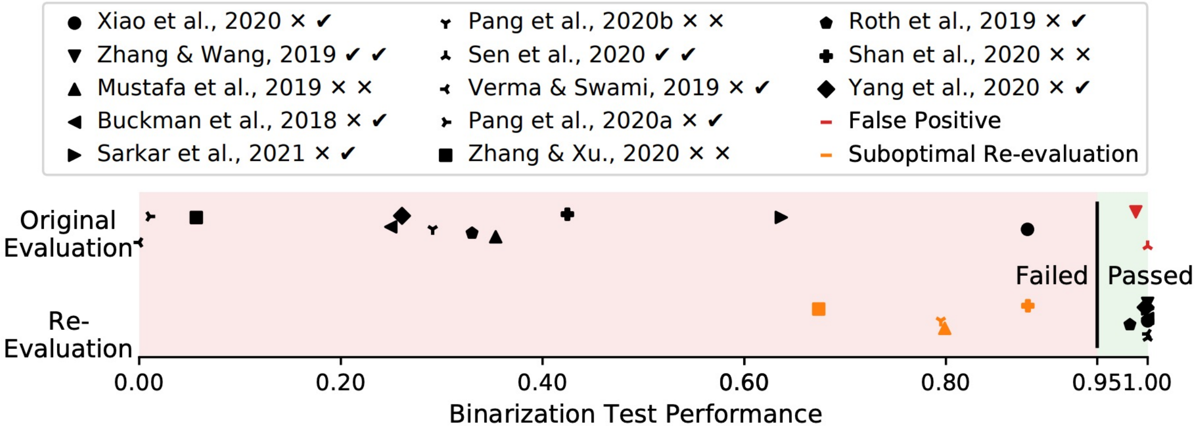
	Defense	T1	T2	T3	T4	T5	T6
		Attack Themes					
Appendix B	*k-Winners Take All* [XZZ20]	●				●	●
Appendix C	*The Odds are Odd* [RKH19]			●	●		
Appendix D	*Generative Classifiers* [LBS19]		●	●			
Appendix E	*Sparse Fourier Transform* [BMV18]	●	●				
Appendix F	*Rethinking Cross Entropy* [PXD+20]			●		●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

How to Evaluate a Defense?

- Strong attacks:
  - ✓ **AutoAttack** (one must-to-test attack)
  - ✓ **Margin Decomposition** (MD) attack (better than AutoAttack on ViT)
  - ✓ **Minimum-Margin** (MM) attack (new SOTA 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:**

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

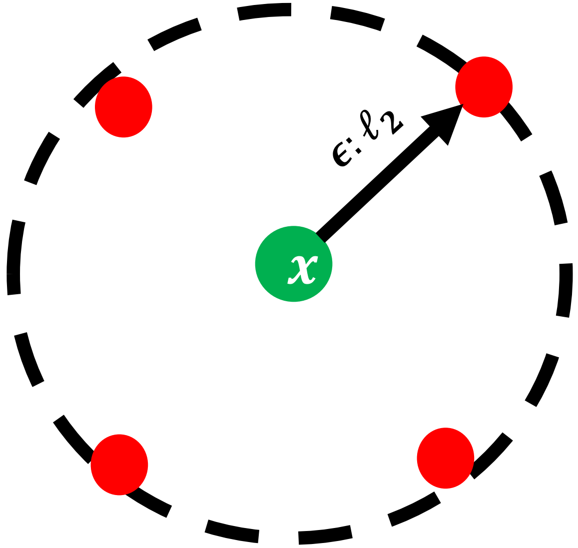
$L(f_\theta(x_i), y_i) = -\mathrm{y_i} \log f_\theta(x_i)$ (交叉熵损失函数)

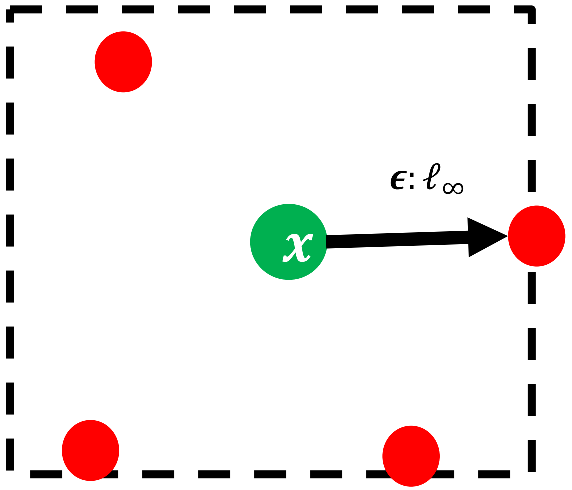
x_i^0 ：原始训练样本, y_i ：x_i^0的正确类别.

Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples[J]. ICLR 2015.

Adversarial Training

在以x为中心的 ϵ-球范围内寻找对抗性最强的样本

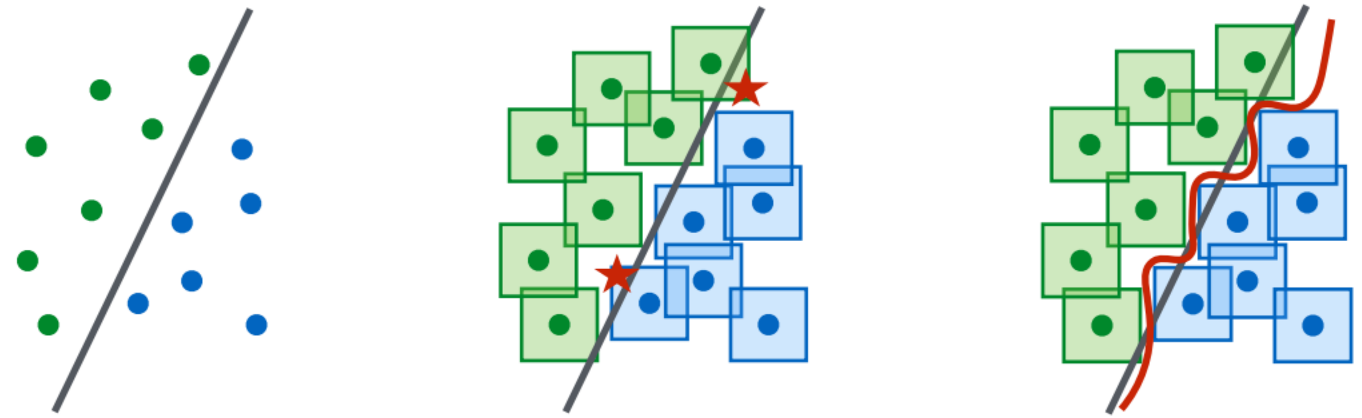




ℓ_∞**范式更常见**

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

$$\min_{\theta} \mathbb{E}_{(\boldsymbol{x},y)\in D}\left[\alpha \mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x}), y) + (1-\alpha)\mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x}_{\mathrm{adv}}), y)\right]$$

$$\boldsymbol{x}_{\mathrm{adv}} = \boldsymbol{x} + \epsilon \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x}), y))$$

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

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

χ_{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

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

- Differences to adversarial training
  - L2 regularized perturbation
  - Use both clean and adv examples for training
  - Use KL divergence to generate adv examples

Weaknesses of Early AT Methods

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

- Use FGSM or BIM to solve the inner maximization problem
- FGSM and BIM were later found to be weak attacks
- Overfit to ϵ-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 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)	0%*
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)	0%*
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)	9%*
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)	55%**
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	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

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

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

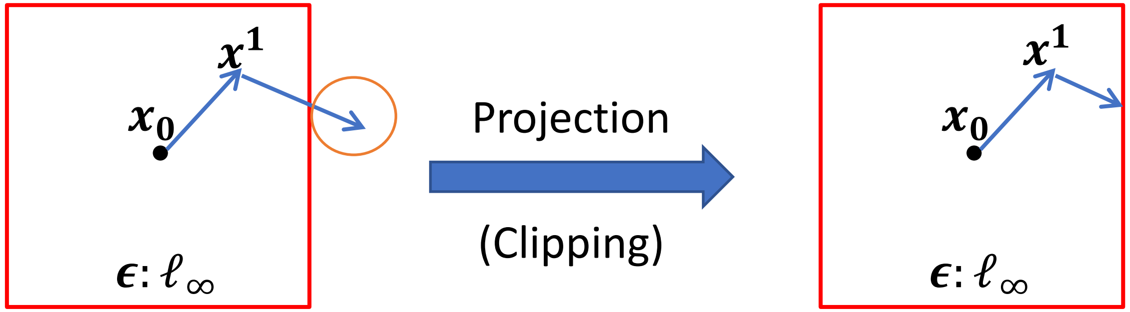
In constrained optimization, **Projected Gradient Descent (PGD)** is the best first-order solver

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

PGD Adversarial Training

Projected Gradient Descent (PGD)

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

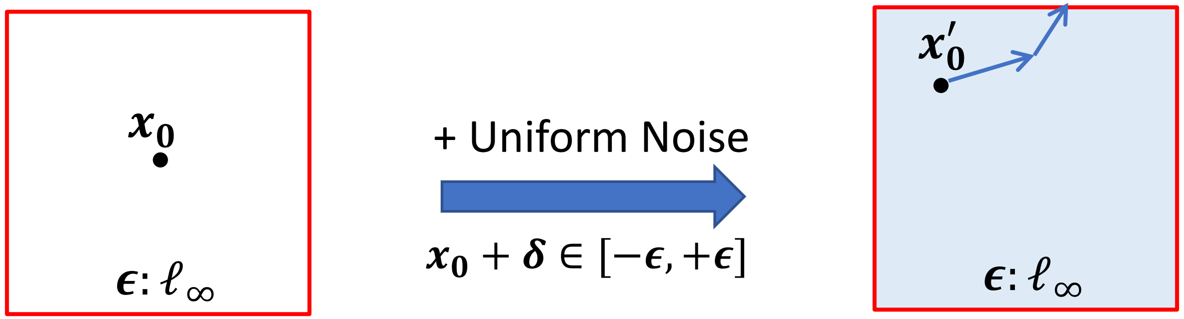


- PGD is an optimizer
- PGD is also known as an adv attack in the field of AML

PGD Adversarial Training

Projected Gradient Descent (PGD)

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



- ☐ Random initialization

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

PGD Adversarial Training

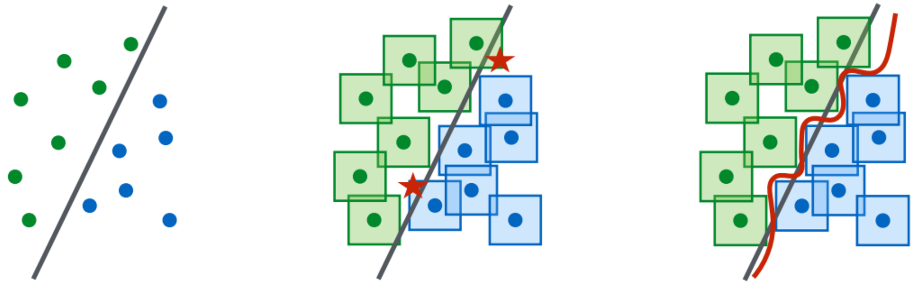
Characteristics of PGD adversarial training

- 只在对抗样本上训练模型
- 通用鲁棒性：< ϵ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.*

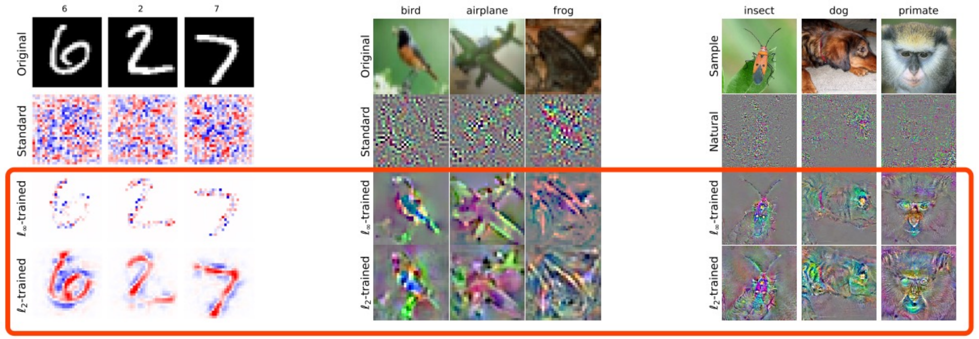
PGD Adversarial Training

决策边界：



普通训练 对抗样本 对抗训练

鲁棒特征：



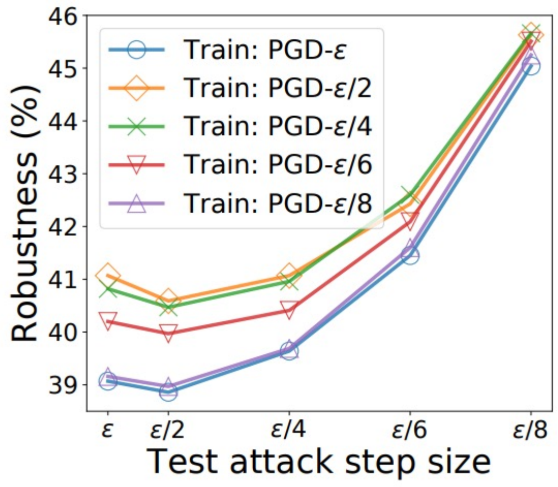
(a) MNIST (b) CIFAR-10 (c) Restricted ImageNet

普通训练

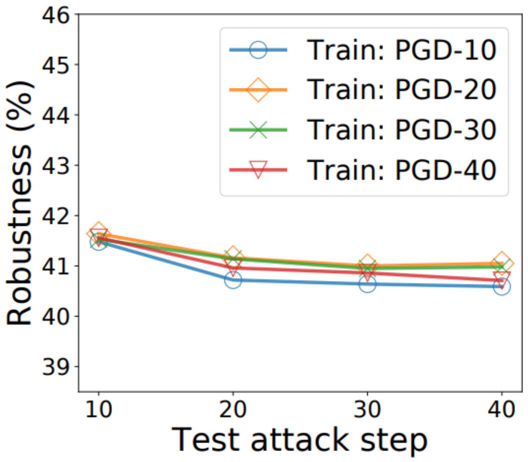
对抗训练

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.*

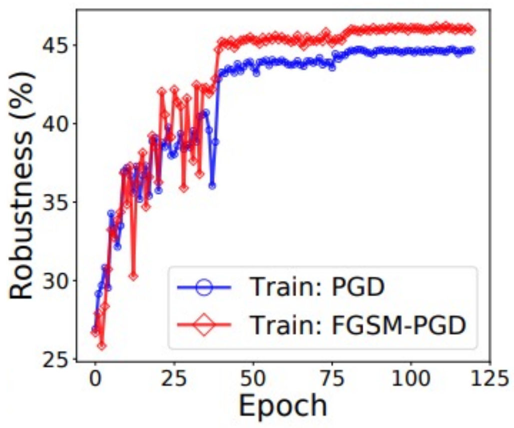
Dynamic Adversarial Training (DART)



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ϵ-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^{th} step of the inner maximization. The First-Order Stationary Condition of x^k is:

$$\mathbf{c}\left(\boldsymbol{x}^{\boldsymbol{k}}\right) = \max_{\mathbf{x} \in \chi} \left\langle \mathbf{x} - \boldsymbol{x}^{\boldsymbol{k}}, \boldsymbol{\nabla}_{\boldsymbol{x}} \mathbf{f}(\boldsymbol{\theta}, \boldsymbol{x}^{\boldsymbol{k}}) \right\rangle,$$

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

FOSC:

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

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

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

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

Robustness on CIFAR-10 with WideResNet

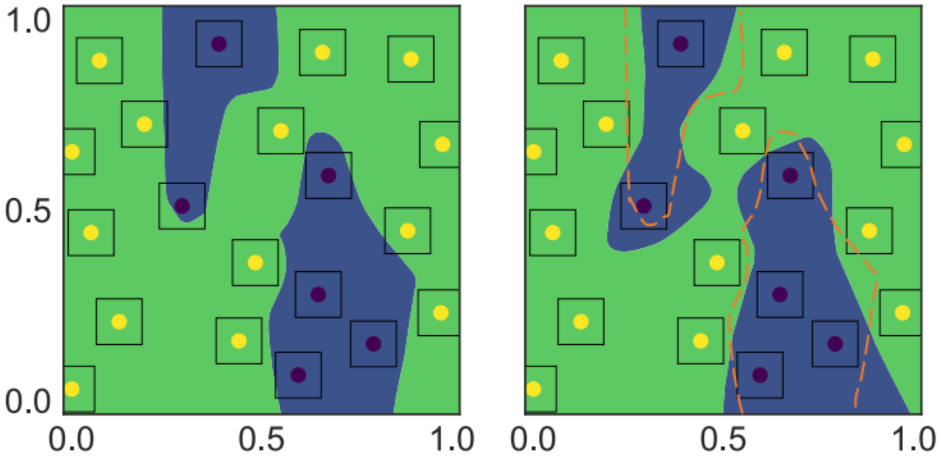
Defense	Clean	FGSM	PGD-20	C&W$_\infty$
Madry's	**87.3**	56.1	45.8	46.8
Curriculum	77.43	57.17	46.06	42.28
Dynamic	85.03	**63.53**	**48.70**	**47.27**

DART improves robustness

TRADES

Use distribution loss (KL) for inner and outer optimizations

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



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

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	$\mathcal{A}_{\mathrm{nat}}(f)$	$\mathcal{A}_{\mathrm{rob}}(f)$
Buckman et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 (ℓ_∞)	-	0%
Ma et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 (ℓ_∞)	-	5%
Dhillon et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 (ℓ_∞)	-	0%
Song et al. (2018)	gradient mask	Athalye et al. (2018)	CIFAR10	0.031 (ℓ_∞)	-	9%
Na et al. (2017)	gradient mask	Athalye et al. (2018)	CIFAR10	0.015 (ℓ_∞)	-	15%
Wong et al. (2018)	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	27.07%	23.54%
Madry et al. (2018)	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	87.30%	**47.04%**
Zheng et al. (2016)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	94.64%	0.15%
Kurakin et al. (2017)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	85.25%	45.89%
Ross & Doshi-Velez (2017)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	95.34%	0%
TRADES ($1/\lambda = 1.0$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	88.64%	49.14%
TRADES ($1/\lambda = 6.0$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	84.92%	**56.61%**
TRADES ($1/\lambda = 1.0$)	regularization	DeepFool (ℓ_∞)	CIFAR10	0.031 (ℓ_∞)	88.64%	59.10%
TRADES ($1/\lambda = 6.0$)	regularization	DeepFool (ℓ_∞)	CIFAR10	0.031 (ℓ_∞)	84.92%	61.38%
TRADES ($1/\lambda = 1.0$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	88.64%	84.41%
TRADES ($1/\lambda = 6.0$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	84.92%	81.58%
TRADES ($1/\lambda = 1.0$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	88.64%	51.26%
TRADES ($1/\lambda = 6.0$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	84.92%	57.95%
TRADES ($1/\lambda = 1.0$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	88.64%	84.03%
TRADES ($1/\lambda = 6.0$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	84.92%	81.24%
Samangouei et al. (2018)	gradient mask	Athalye et al. (2018)	MNIST	0.005 (ℓ_2)	-	55%
Madry et al. (2018)	robust opt.	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.36%	96.01%
TRADES ($1/\lambda = 6.0$)	regularization	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.48%	96.07%
TRADES ($1/\lambda = 6.0$)	regularization	C&W	MNIST	0.005 (ℓ_2)	99.48%	99.46%

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

TRADES vs VAT vs ALP

TRADES: $$\min_{\theta} \mathbb{E}_{(\boldsymbol{x},y)\sim D}\Big[\underbrace{\mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x}),y)}_{\text{提升准确率}} + \underbrace{\lambda \max_{\|\boldsymbol{r}\|\le\epsilon} \mathcal{L}_{\mathrm{KL}}(f(\boldsymbol{x}), f(\boldsymbol{x}+\boldsymbol{r}))}_{\text{提升鲁棒性}}\Big]$$

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

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

$\boxed{\text{相似的优化框架，不同的损失选择，结果差异很大}}$

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:

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

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

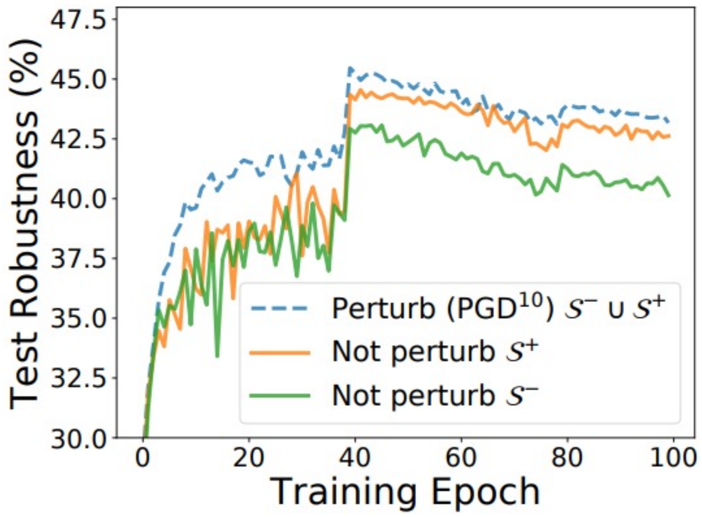
Adversarial examples are only defined on **correctly classified** examples, what about misclassified examples ?

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:

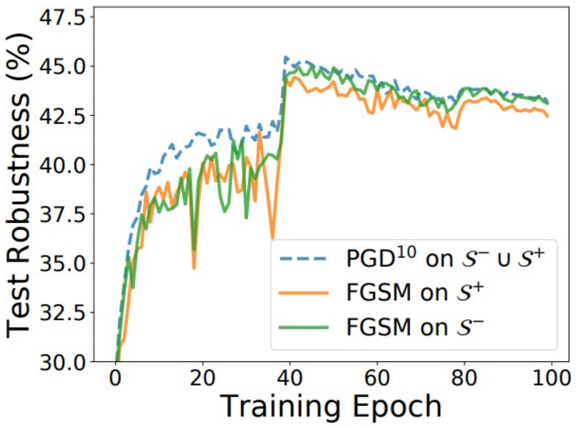
- A pre-trained network to select the **same size (13%)**
  - Subset of misclassified examples S^-
  - Subset of correctly classified examples S^+



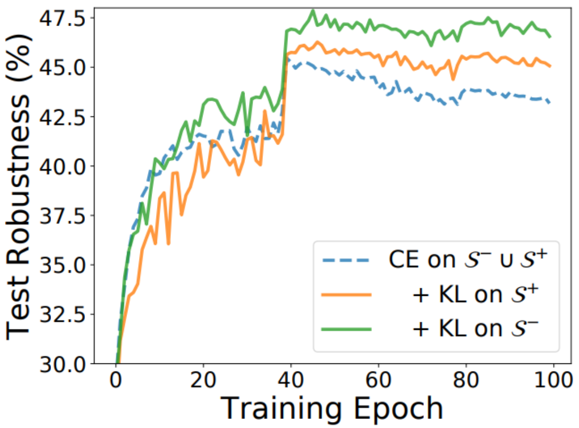
Misclassified examples have a **significant impact** on the final robustness!

MART: Misclassification Aware adveRsarial Training

- For **inner maximization** process:
  - Weak attack on misclassified examples S^-
  - Weak attack on correctly classified examples S^+
- For **outer minimization** process:
  - Regularization on misclassified examples S^-
  - Regularization on correctly classified examples S^+



(b) Inner maximization



(c) Outer minimization

different maximization techniques have negligible effect

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:

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

- Correctly classified and misclassified example:

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

- Misclassification aware adversarial risk:

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

MART: Misclassification Aware adveRsarial Training

- Surrogate loss functions (existing methods and MART):

Defense Method	Loss Function
Standard	$\mathrm{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y)$
ALP	$\mathrm{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) + \lambda \cdot \lVert \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}) - \mathbf{p}(\mathbf{x}, \boldsymbol{\theta}) \rVert_2^2$
CLP	$\mathrm{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) + \lambda \cdot \lVert \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}) - \mathbf{p}(\mathbf{x}, \boldsymbol{\theta}) \rVert_2^2$
TRADES	$\mathrm{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) + \lambda \cdot \mathrm{KL}\big(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}) \Vert \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta})\big)$
MMA	$\mathrm{CE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) \cdot \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}) = y) + \mathrm{CE}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), y) \cdot \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}) \neq y)$
MART	$\mathrm{BCE}(\mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta}), y) + \lambda \cdot \mathrm{KL}(\mathbf{p}(\mathbf{x}, \boldsymbol{\theta}) \Vert \mathbf{p}(\hat{\mathbf{x}}', \boldsymbol{\theta})) \cdot (1 - \mathbf{p}_y(\mathbf{x}, \boldsymbol{\theta}))$

- Semi-supervised extension of MART:

$$\mathrm{BCE}\big(\mathbf{p}(\hat{\mathbf{x}}_i', \boldsymbol{\theta}), y_i\big) = -\log\big(\mathbf{p}_{y_i}(\hat{\mathbf{x}}_i', \boldsymbol{\theta})\big) - \log\big(1 - \max_{k \neq y_i} \mathbf{p}_k(\hat{\mathbf{x}}_i', \boldsymbol{\theta})\big)$$

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

MART: Misclassification Aware adveRsarial Training

Robustness of MART:

- White-box robustness: ResNet-18, CIFAR-10, ϵ=8/255

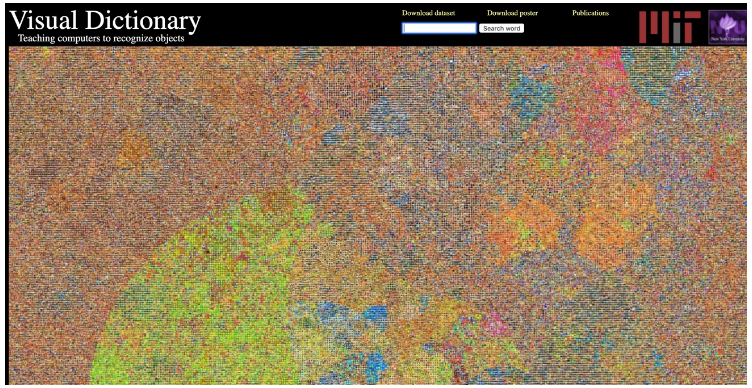
Defense	MNIST				CIFAR-10			
	Natural	FGSM	PGD^{20}	CW_∞	Natural	FGSM	PGD^{20}	CW_∞
Standard	99.11	97.17	94.62	94.25	84.44	61.89	47.55	45.98
MMA	98.92	97.25	95.25	94.77	**84.76**	62.08	48.33	45.77
Dynamic	98.96	97.34	95.27	94.85	83.33	62.47	49.40	46.94
TRADES	**99.25**	96.67	94.58	94.03	82.90	62.82	50.25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

- White-box robustness: WideResNet-34-10, CIFAR-10, ϵ=8/255

Defense	Natural	FGSM		PGD^{20}		PGD^{100}		CW_∞	
		Best	Last	Best	Last	Best	Last	Best	Last
Standard	**87.30**	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
MART	84.17	**67.51**	**67.51**	**58.56**	**57.39**	**57.88**	**55.04**	**54.58**	**54.53**

Using More Data to Improve Robustness

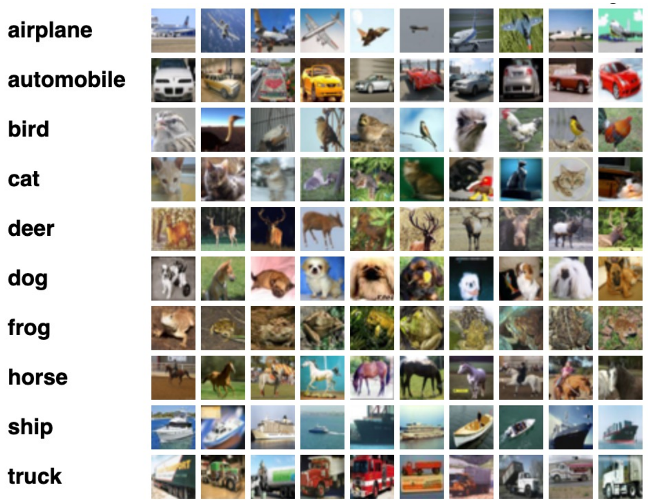
80 Million Tiny Images



500K (10x)

Select (carefully) 100K/500K images into CIAFR-10

CIFAR-10



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):

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

Robust Self-Training (RST):

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

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

RobustBench Leaderboards Paper FAQ Contribute Model Zoo 🚀

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	Robust Principles: Architectural Design Principles for Adversarially Robust CNNs *It uses additional 50M synthetic images in training.*	93.27%	71.07%	71.07%	×	×	RaWideResNet-70-16	BMVC 2023
2	Better Diffusion Models Further Improve Adversarial Training *It uses additional 50M synthetic images in training.*	93.25%	70.69%	70.69%	×	×	WideResNet-70-16	ICML 2023
3	Improving the Accuracy-Robustness Trade-off of Classifiers via Adaptive Smoothing *It uses an ensemble of networks. The robust base classifier uses 50M synthetic images.*	95.23%	68.06%	68.06%	×	☑	ResNet-152 + WideResNet-70-16 + mixing network	arXiv, Jan 2023
4	Decoupled Kullback-Leibler Divergence Loss *It uses additional 20M synthetic images in training.*	92.16%	67.73%	67.73%	×	×	WideResNet-28-10	arXiv, May 2023
5	Better Diffusion Models Further Improve Adversarial Training *It uses additional 20M synthetic images in training.*	92.44%	67.31%	67.31%	×	×	WideResNet-28-10	ICML 2023
6	Fixing Data Augmentation to Improve Adversarial Robustness *66.56% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)*	92.23%	66.58%	66.56%	×	☑	WideResNet-70-16	arXiv, Mar 2021
7	Improving Robustness using Generated Data *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

数据，数据，还是数据！！

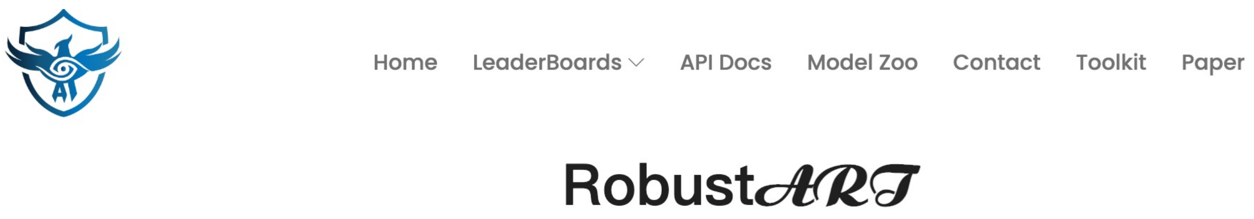
- 数据增广
- 数据生成

State-of-the-art: AT Methods

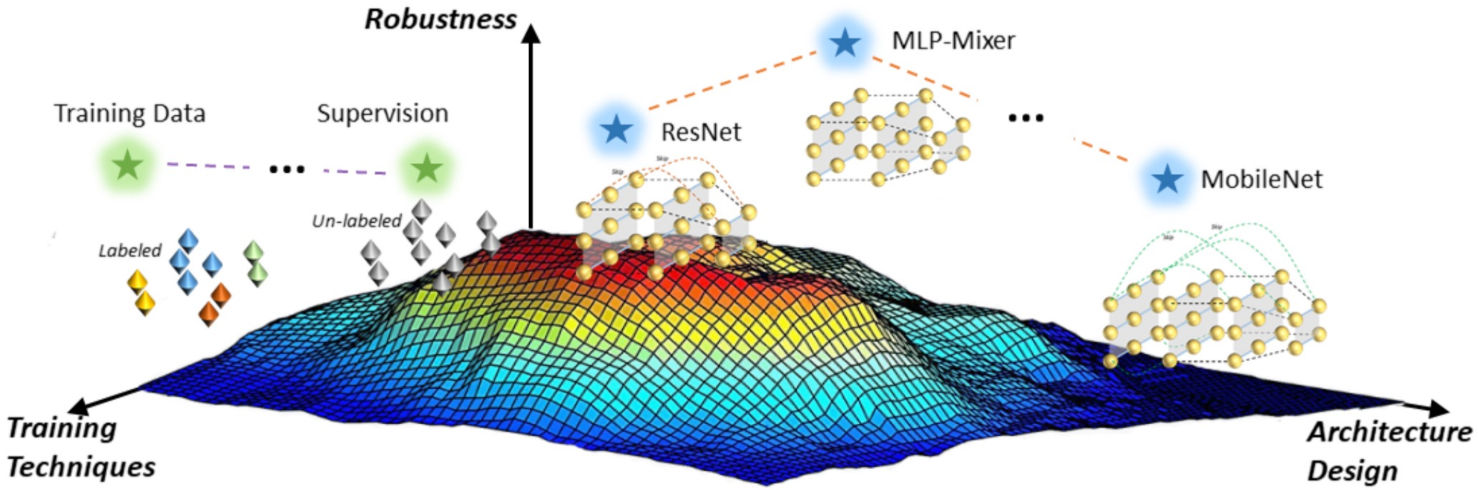
Clean accuracy: 94% vs 66% (robustness)

DNN: WideResNet-70-16; Dataset: CIFAR-10
Perturbation: $\epsilon = 8/255$;
Evaluation attack: AutoAttack

State-of-the-art: DNN Architecture



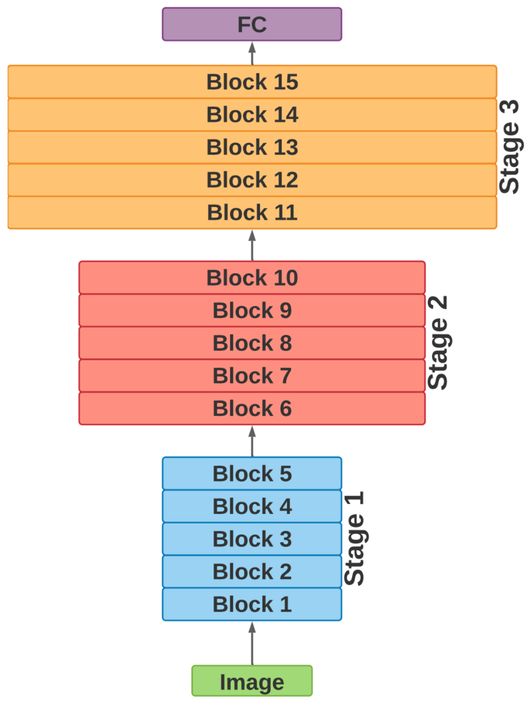
RobustART is the first comprehensive **Robust**ness investigation benchmark on large-scale dataset ImageNet regarding **AR**chitectural design (**49** human-designed off-the-shelf architectures and **1200+** neural architecture searched networks) and **T**raining techniques (**10+** general ones *e.g.*, extra training data, etc) towards diverse noises (adversarial, natural, and system noises). Our benchmark (including open-source toolkit, pre-trained model zoo, datasets, and analyses): **(1)** presents an open-source platform for conducting comprehensive evaluation on diverse robustness types; **(2)** provides a variety of pre-trained models with different training techniques to facilitate robustness evaluation; **(3)** proposes a new view to better understand the mechanism towards designing robust DNN architectures, backed up by the analysis. We will continuously contribute to building this ecosystem for the community.



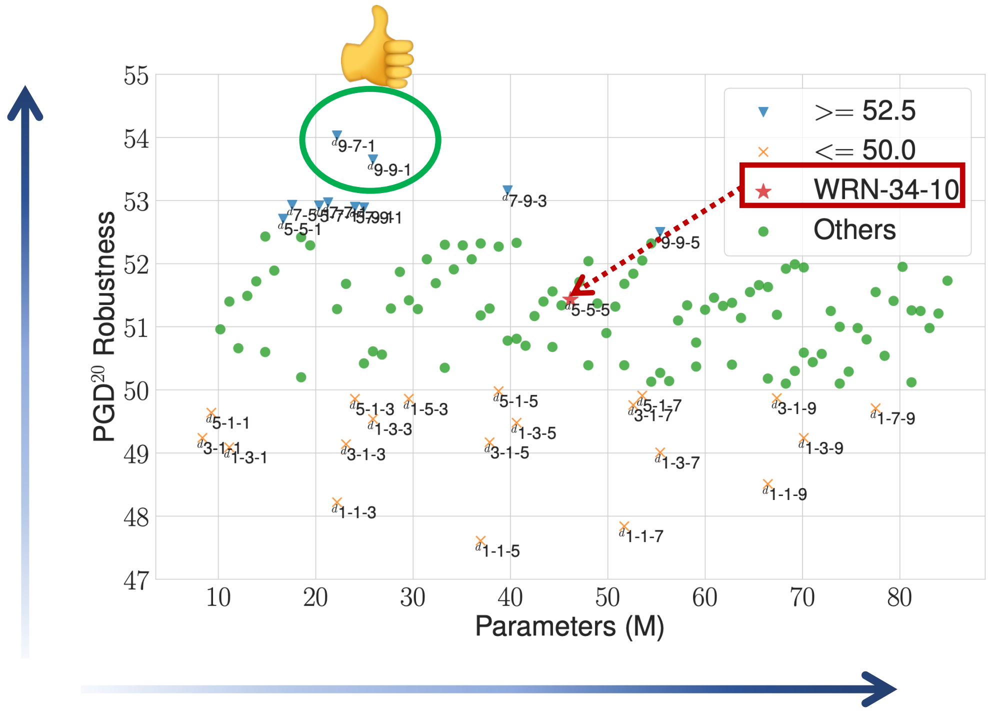
http://robust.art/ Tang et al. "RobustART: Benchmarking robustness on architecture design and training techniques." *arXiv:2109.05211*, 2021.

State-of-the-art: DNN Architecture

Reduce the deep layers can improve robustness



WideResNet-34-10

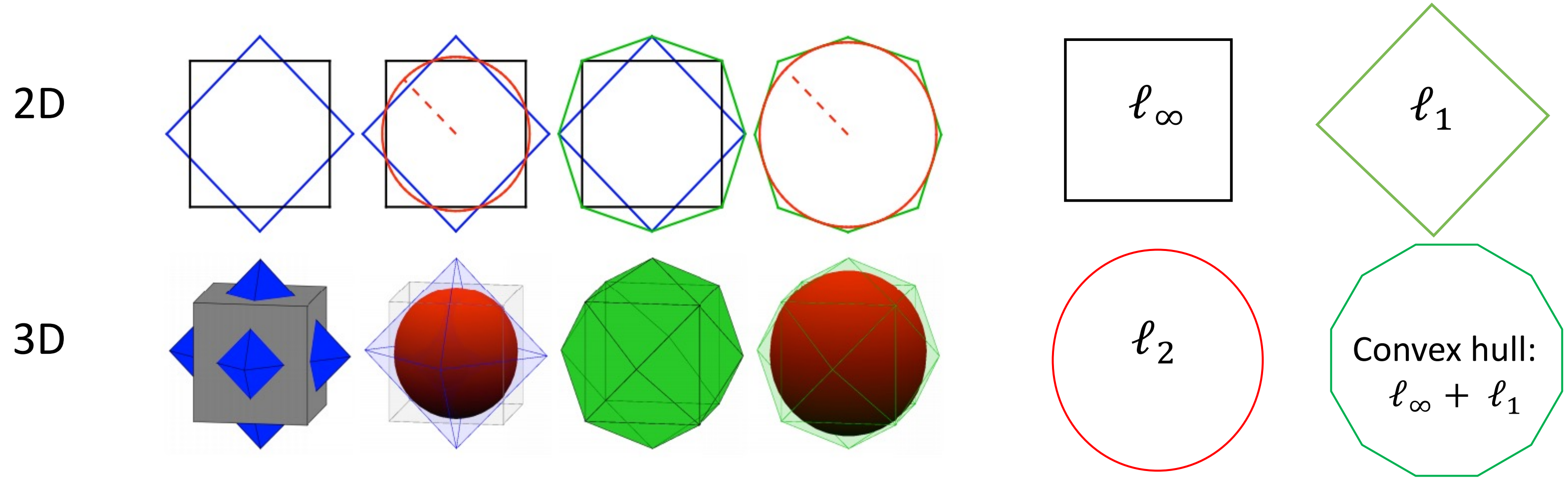


Depth Grid Search

d: depth
^{d}5-5-5: depth=5 at all 3 stages

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 ℓ_2 norm ball
  - Laplacian randomized smoothing -> ℓ_1 robustness
  - Uniform randomized smoothing -> ℓ_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 mode in certain ways



关键点：如何在神经网络里传播边界？

Existing Challenges

- How to attack large language/vision/multi-model models
- How to defend large language/vision/multi-model models
- How to adapt adv training for different applications
- How to reduce the cost of defense: acceleration, loss of clean acc
- How to combine adv detection with adversarial training
- How to include adv training into the pretraining/finetuning pipeline

谢谢！