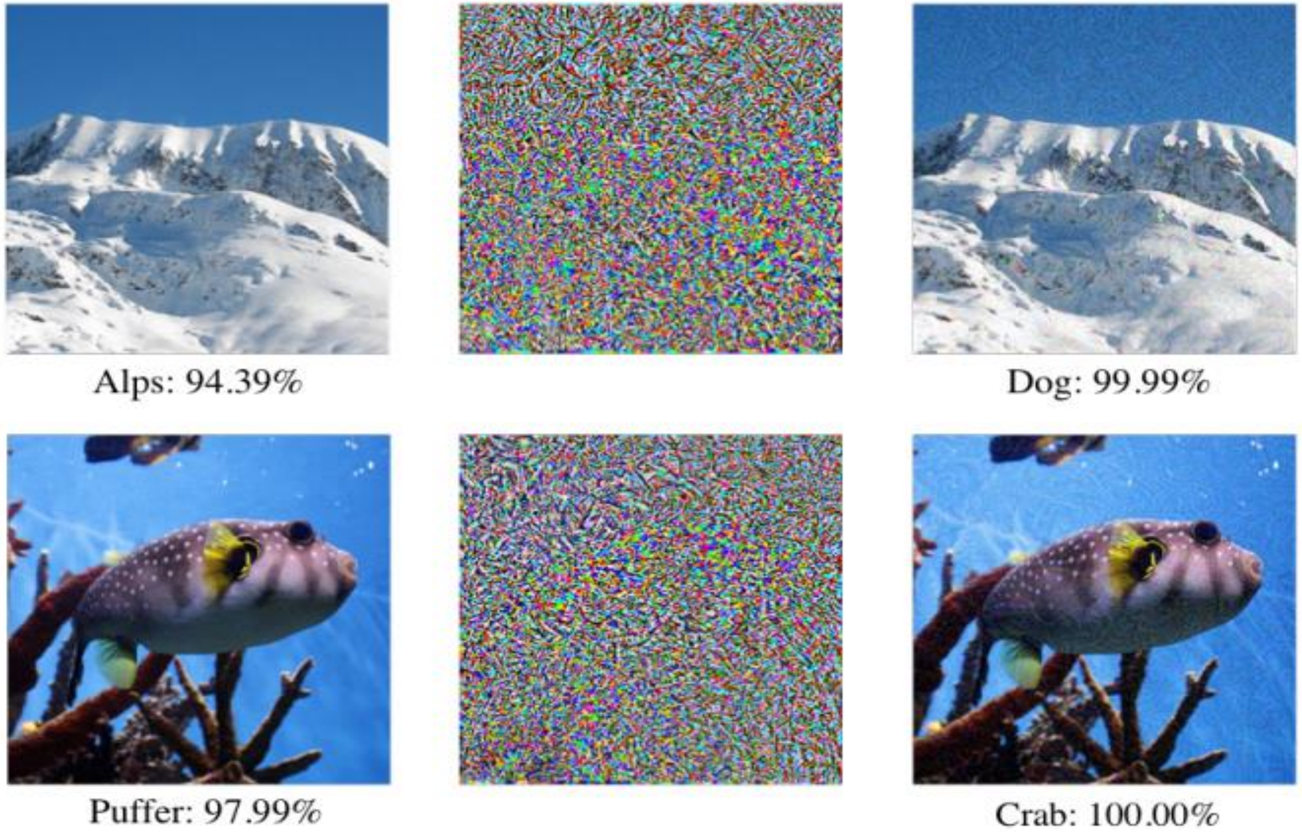
Understand and Benchmark Adversarial Robustness of Deep Learning

Jun Zhu

Tsinghua-Bosch Joint Center for ML
Institute for Artificial Intelligence
Department of Computer Science and Technology
Tsinghua University

Deep networks are vulnerable to adversarial examples

Clean images **Adversarial noise** **Adversarial examples**



(Figure is from Dong et al. CVPR 2018)

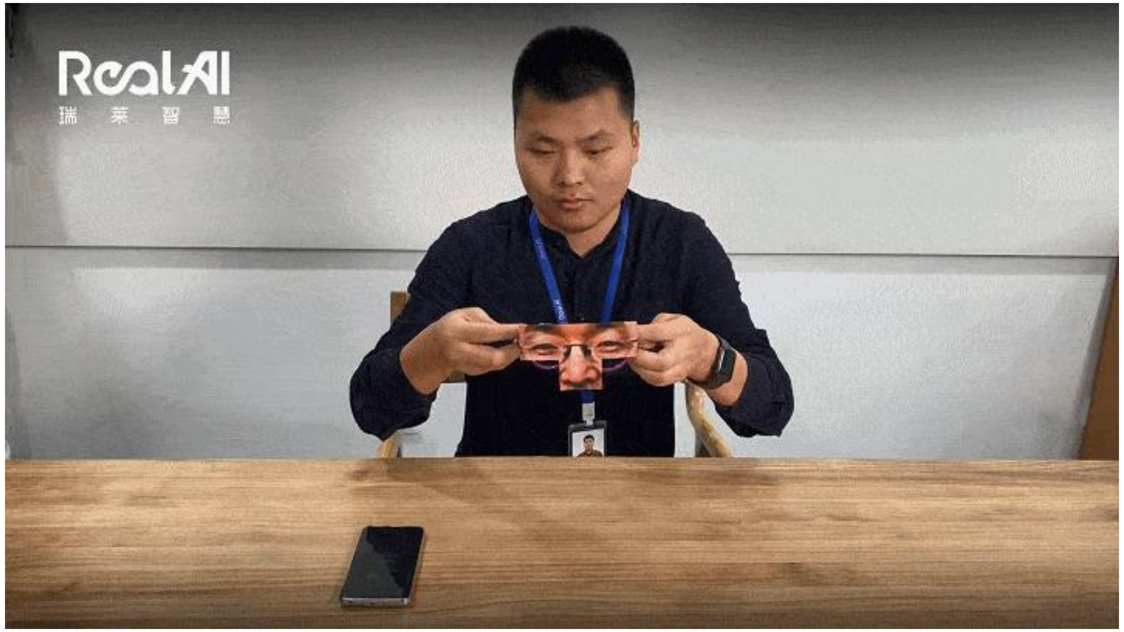
C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In ICLR, 2014

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

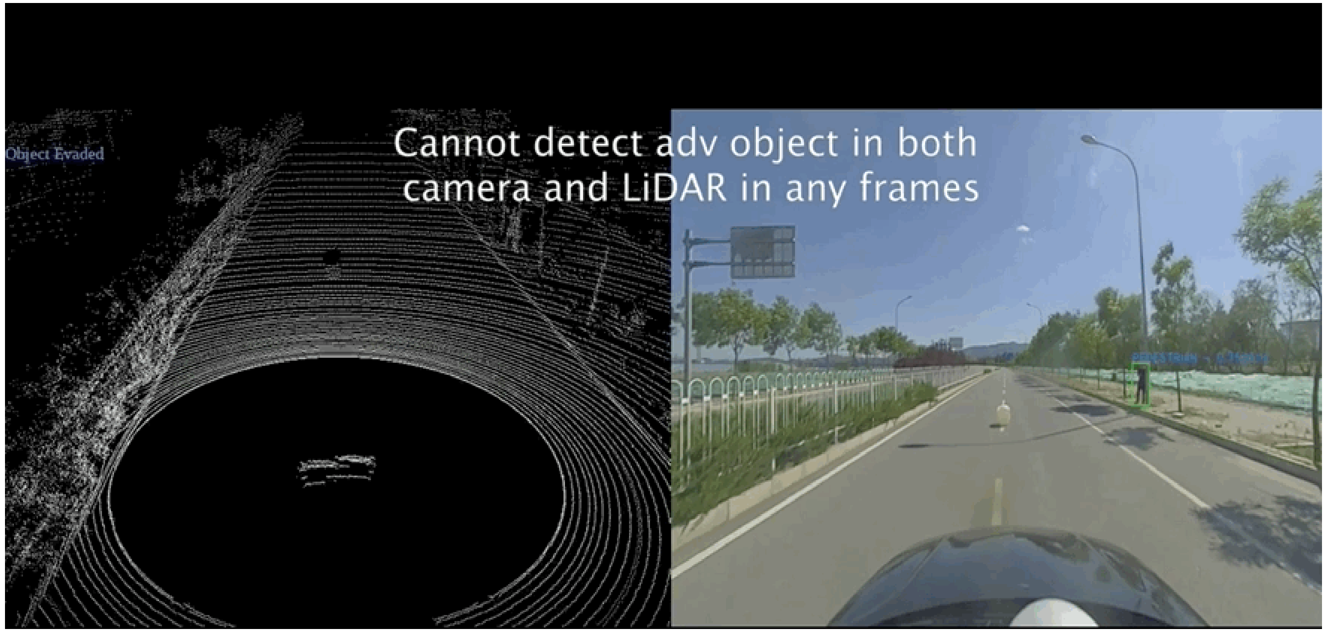
Adversarial attack in practice

◆ Adversarial attack happens in both digital and physical worlds

Unlocked 19 types of the mainstream smart phones within 15 minutes with one "adversarial glass"!



failure chance >= 99.1%

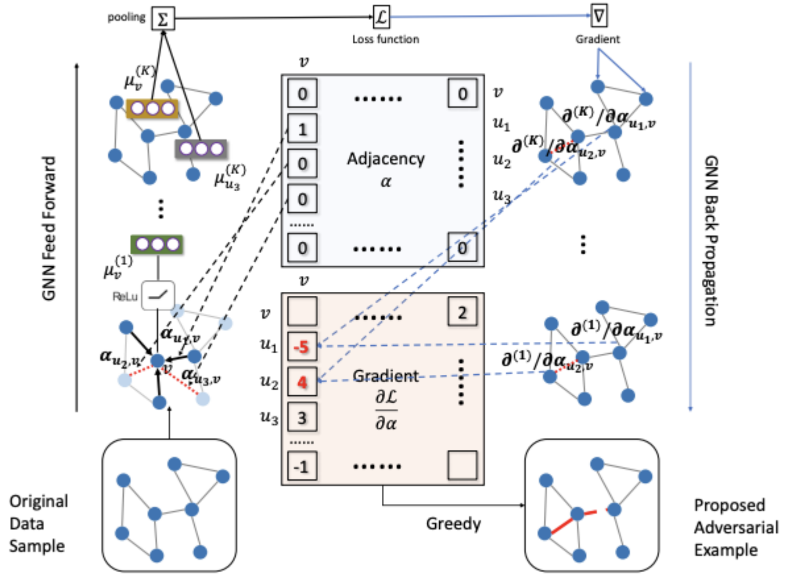


[Cao, Wang, Xiao, et al, IEEE Symposium on Security and Privacy, 2021]

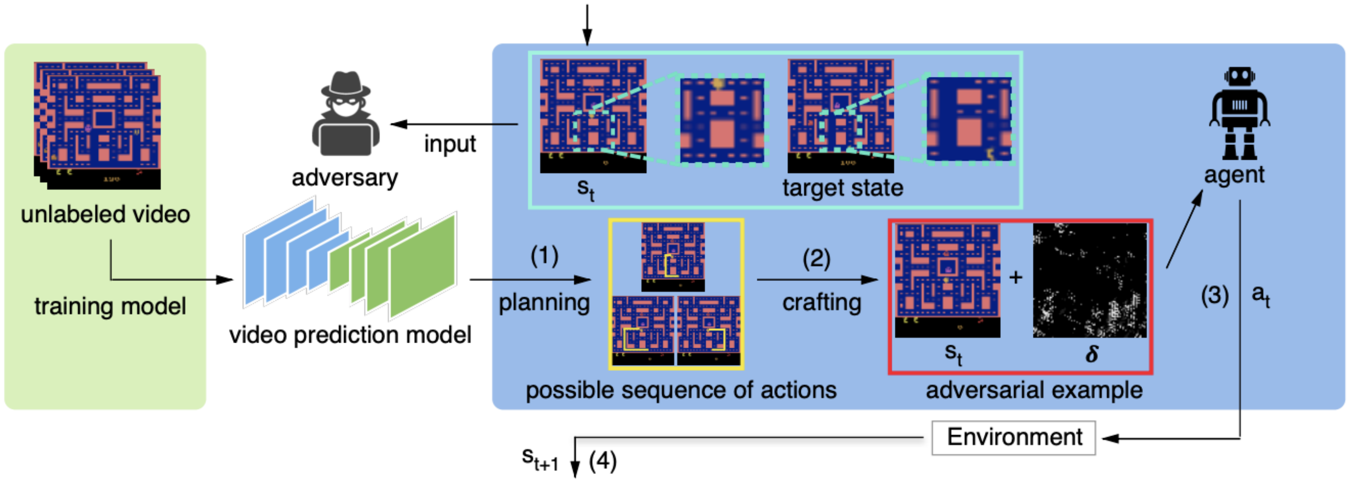
Not only in computer vision

Movie Review (Positive (POS) ↔ Negative (NEG))	
Original (Label: NEG)	The characters, cast in impossibly **contrived situations**, are **totally** estranged from reality.
Attack (Label: POS)	The characters, cast in impossibly **engineered circumstances**, are **fully** estranged from reality.
Original (Label: POS)	It cuts to the **knot** of what it actually means to face your **scares**, and to ride the **overwhelming** **metaphorical wave** that life wherever it takes you.
Attack (Label: NEG)	It cuts to the **core** of what it actually means to face your **fears**, and to ride the **big** **metaphorical wave** that life wherever it takes you.
SNLI (Entailment (ENT), Neutral (NEU), Contradiction (CON))	
Premise	Two small boys in blue soccer uniforms use a wooden set of steps to wash their hands.
Original (Label: CON)	The boys are in band **uniforms**.
Adversary (Label: ENT)	The boys are in band **garment**.
Premise	A child with wet hair is holding a butterfly decorated beach ball.
Original (Label: NEU)	The **child** is at the **beach**.
Adversary (Label: ENT)	The **youngster** is at the **shore**.

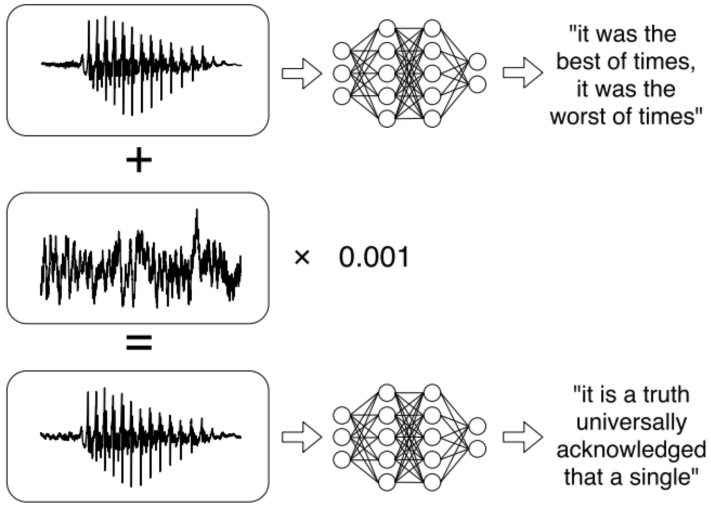
NLP (Jin et al. AAAI 2020)



Graph (Dai et al. ICML 2018)

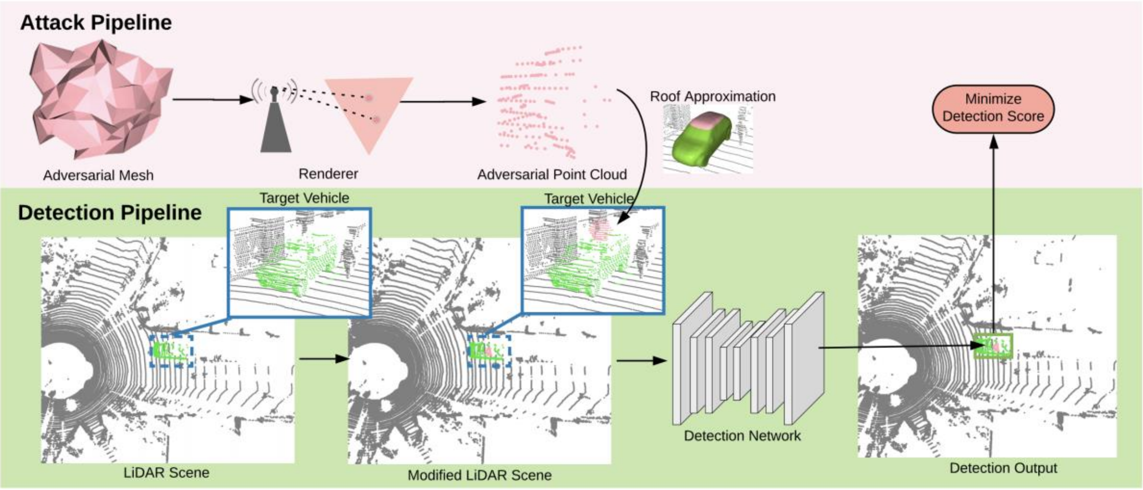


Reinforcement Learning (Lin et al. IJCAI 2017)

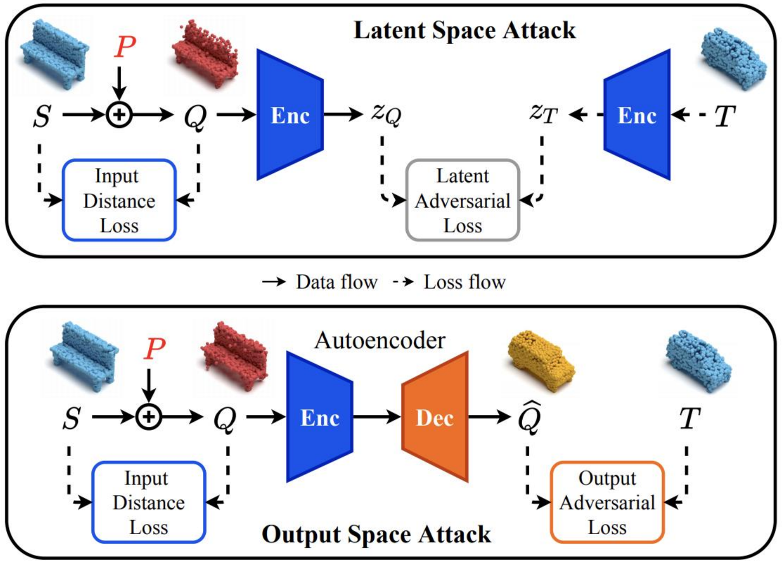


Audio (Carlini and Wagner. S&P 2018)

Not only in computer vision



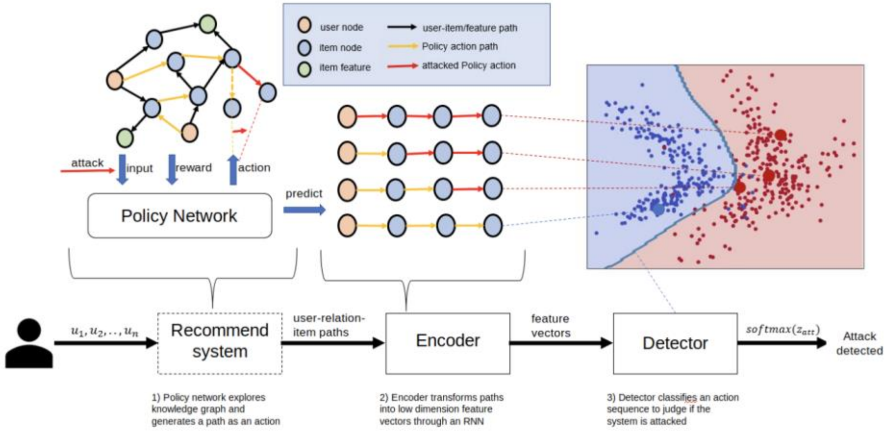
LiDAR (Tu et al. CVPR 2020)



3D Point Cloud (Lang et al. 2020)

Class	Representative Example
VC	**OD:** Given a string a, what is the length of a. **OO:** `(strlen a)` **AD:** Given a string b, what is the length of b. **AO:** `(strlen a)`
RR	**OD:** Given a number a, compute the product of **all** the numbers from 1 to a. **OO:** `(invoke1 (lambda1 (if ( ≤ arg1 1 )1(*( self( -arg1 1 )) arg1 ))) a)` **AD:** Given a number a, compute the product of the numbers from 1 to a. **AO:** `( * a 1 )`
SR	**OD:** consider an array of numbers, what is reverse of elements in the given array that are odd **OO:** `(reverse ( filter a ( lambda1 ( == ( % arg1 2 )1))))` **AD:** consider an array of numbers, what equals reverse of elements in the given array that are odd **AO:** `(reduce ( filter a ( lambda1 ( == ( % arg1 2 )1))))`

Code Generation (Anand et al. 2021)

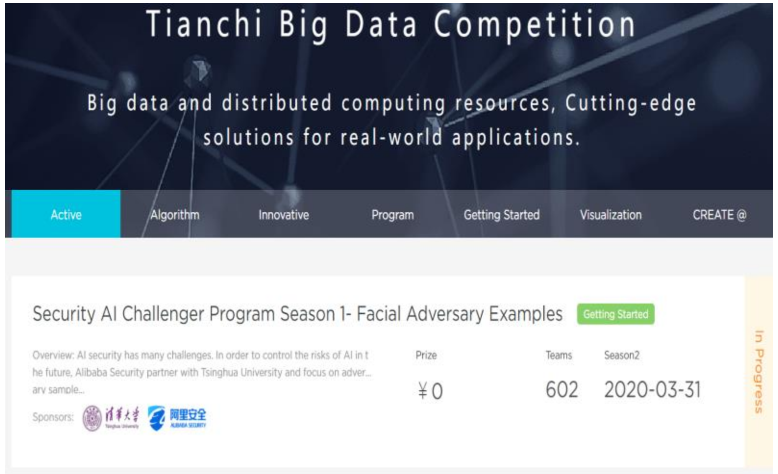


Recommender System (Cao et al. SIGIR 2020)

Competitions on adversarial attack and defense

- Google Brain organized the 1st competition on Adversarial Attack and Defense at NeurIPS 2017
  - Three tasks (black-box)
    - Non-targeted adversarial attack (91 teams)
    - Targeted adversarial attack (65 teams)
    - Defense against adversarial attack (107 teams)
  - We won all three tasks with a large margin (2 papers at CVPR 2018)
    - A summary paper on this competition (Kurakin et al., 2018)

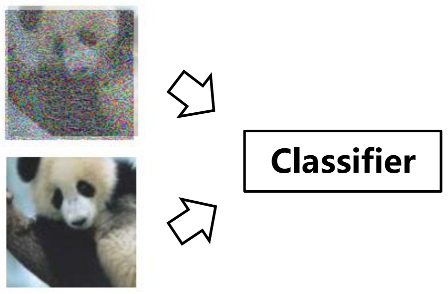
- GeekPwn competitions
  - We won the 1st place at Defcon AI Security competition, 2018
  - CAAD CTF 2019 two 1st places



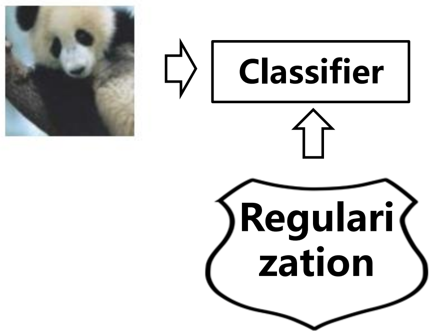
- Security AI Challenger Program, from 2019 (completed seven challenges)
  - Joint with ICDM 2020, CVPR 2021

Categories of existing defense

- The defense techniques can be categorized as (Dong et al., 2020):
  - Robust Training
    - Adversarial training
    - Regularization
  - Input Transformation
  - Randomization
  - Model Ensemble
  - Certified Defenses



Adversarial training



Regularization

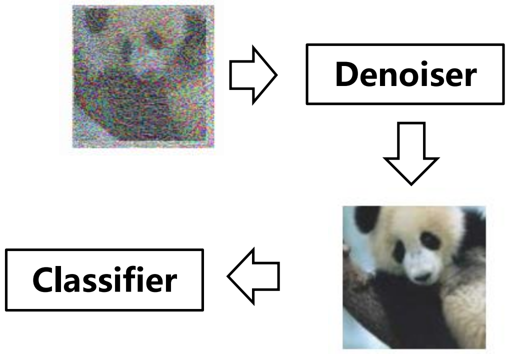
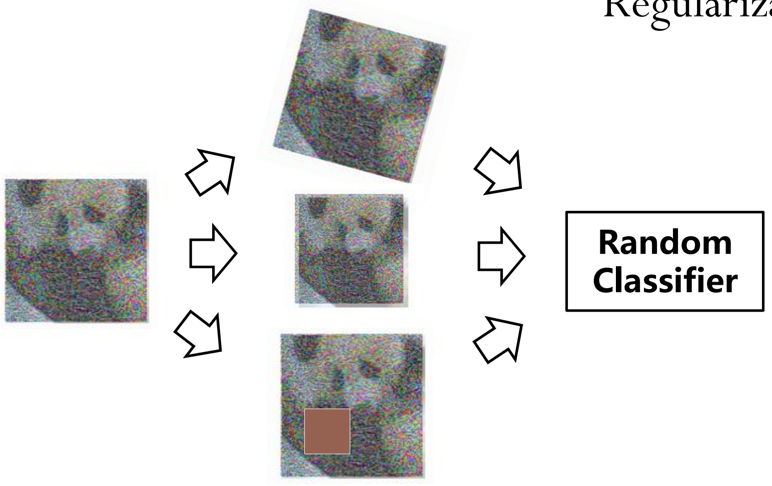


Image denoising

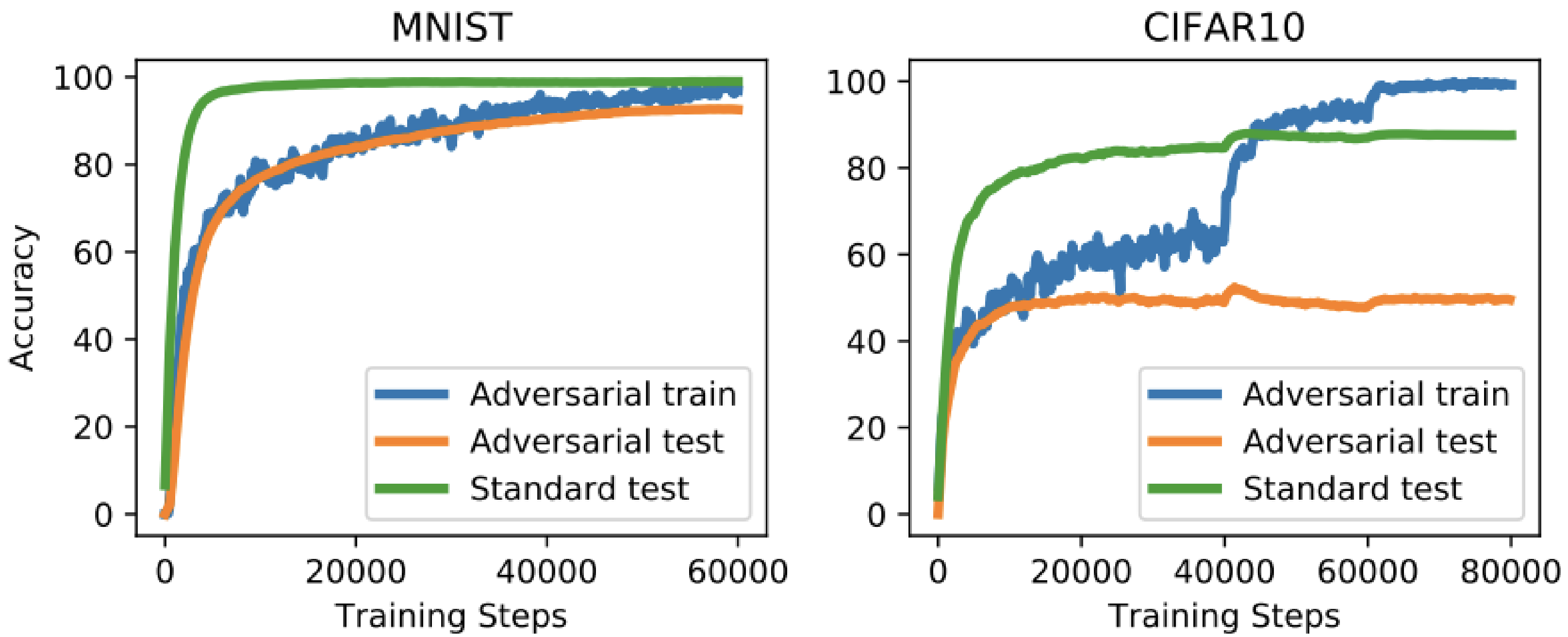


Randomization

Rethinking Softmax Cross-Entropy Loss for Adversarial Robustness

(Pang, Xu, Dong, Du, Chen, Zhu, ICLR 2020)

Observation: Adversarial Robustness requires Higher Sample Complexity



The same dataset, e.g., CIFAR-10, which enables good standard accuracy may not suffice to train robust models.

(Schmidt et al. NeurIPS 2018)

Possible Solutions

- **Introducing extra labeled data**

(Hendrycks et al. ICML 2019)

- **Introducing extra unlabeled data**

(Alayrac et al. NeurIPS 2019; Carmon et al. NeurIPS 2019)

Our solution: Increase sample density to induce locally sufficient training data for robust learning

Q1: What is the definition of sample density?

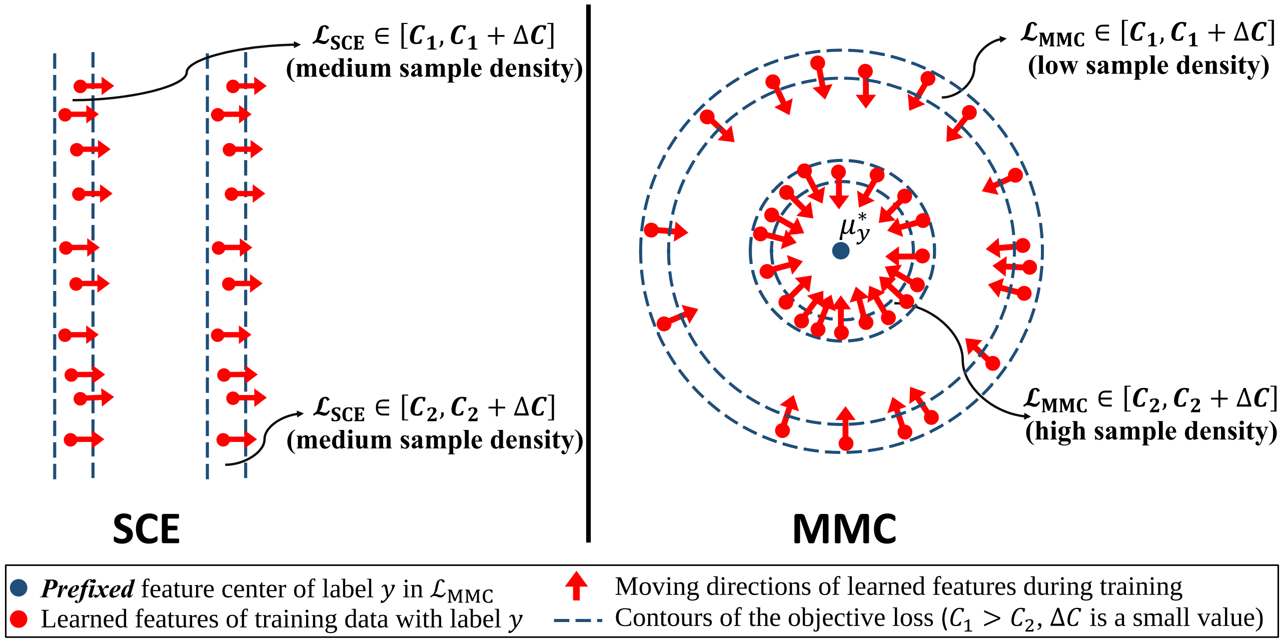
Q2: Can existing training objectives induce high sample density?

Sample Density

Given a training dataset $\mathcal{D}$ with N input-label pairs, and the feature mapping Z trained by the objective $\mathcal{L}(Z(x), y)$ on this dataset, we define the sample density nearby the feature point $z = Z(x)$ following the similar definition in physics (Jackson, 1999) as

$$\mathbb{SD}(z) = \frac{\Delta N}{\mathrm{Vol}(\Delta B)}. \tag{2}$$

Here $\mathrm{Vol}(\cdot)$ denotes the volume of the input set, ΔB is a small neighbourhood containing the feature point z, and $\Delta N = |Z(\mathcal{D}) \cap \Delta B|$ is the number of training points in ΔB, where $Z(\mathcal{D})$ is the set of all mapped features for the inputs in $\mathcal{D}$. Note that the mapped feature z is still of the label y.



Generalized Softmax Cross Entropy Loss (g-SCE loss)

We define g-SCE loss as

$$\mathcal{L}_{\text{g-SCE}}(Z(x), y) = -1_y^\top \log\left[\text{softmax}(h)\right],$$

where $h_i = -(z-\mu_i)^\top \Sigma_i (z-\mu_i) + B_i$ **is the logits in quadratic form.**

We note that the SCE loss is included in the family of g-SCE loss as

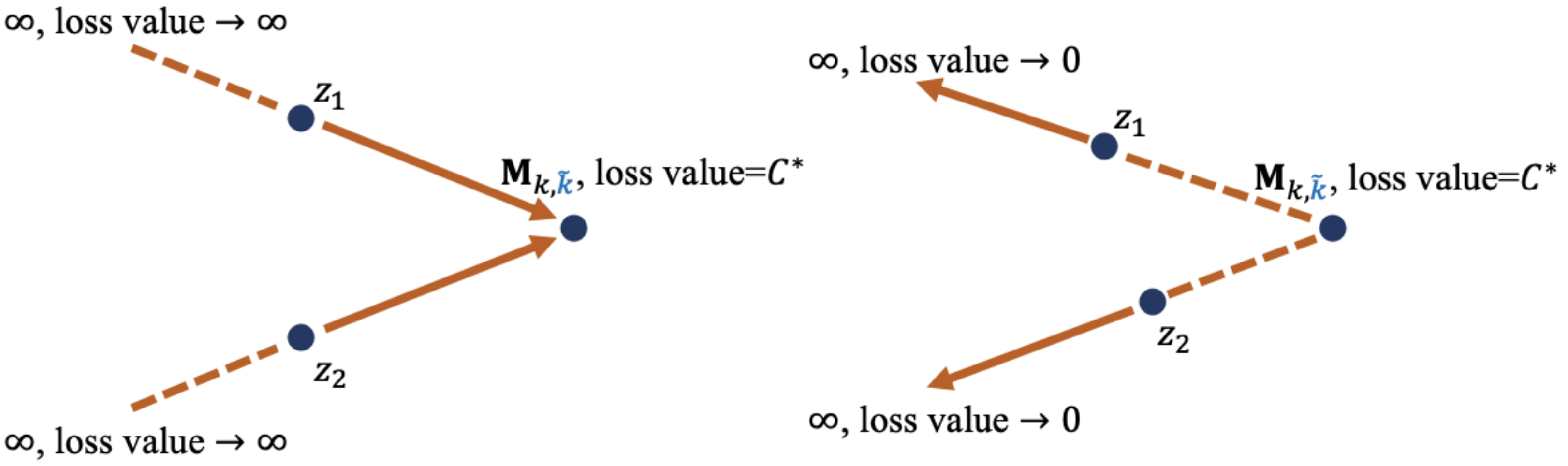
$$\text{softmax}(Wz+b)_i = \frac{\exp(W_i^\top z + b_i)}{\sum_{l\in[L]} \exp(W_l^\top z + b_l)} = \frac{\exp(-\|z - \frac{1}{2}W_i\|_2^2 + b_i + \frac{1}{4}\|W_i\|_2^2)}{\sum_{l\in[L]} \exp(-\|z - \frac{1}{2}W_l\|_2^2 + b_l + \frac{1}{4}\|W_l\|_2^2)}.$$

Key results #1: The widely used g-SCE loss is not sufficient!

Theorem 1. *(Proof in Appendix A.1) Given* $(x, y) \in \mathcal{D}_{k,\tilde{k}}$, $z = Z(x)$ *and* $\mathcal{L}_{g\text{-}SCE}(z, y) = C$, *if there are* $\Sigma_k = \sigma_k I$, $\Sigma_{\tilde{k}} = \sigma_{\tilde{k}} I$, *and* $\sigma_k \neq \sigma_{\tilde{k}}$, *then the sample density nearby the feature point* z *based on the approximation in Eq. (6) is*

$$\mathbb{SD}(z) \propto \frac{N_{k,\tilde{k}} \cdot p_{k,\tilde{k}}(C)}{\left[\mathbf{B}_{k,\tilde{k}} + \frac{\log(C_e - 1)}{\sigma_k - \sigma_{\tilde{k}}}\right]^{\frac{d-1}{2}}}, \text{ and } \mathbf{B}_{k,\tilde{k}} = \frac{\sigma_k \sigma_{\tilde{k}} \|\mu_k - \mu_{\tilde{k}}\|_2^2}{(\sigma_k - \sigma_{\tilde{k}})^2} + \frac{B_k - B_{\tilde{k}}}{\sigma_k - \sigma_{\tilde{k}}}, \tag{7}$$

where for the input-label pair in $\mathcal{D}_{k,\tilde{k}}$, *there is* $\mathcal{L}_{g\text{-}SCE} \sim p_{k,\tilde{k}}(c)$.



The case: $\sigma_k > \sigma_{\tilde{k}}$

The case: $\sigma_k < \sigma_{\tilde{k}}$
(Preferred by models since lower loss values)

The 'Curse' of Softmax Function

$$\mathcal{L}_{\text{g-SCE}}(Z(x), y) = -1_y^\top \log\left[\underline{\text{softmax}}(h)\right],$$

- **The softmax makes the loss value only depend on the relative relation among logits.**
- **This causes indirect and unexpected supervisory signal on the learned features.**

Our Method: Max-Mahalanobis Center (MMC) Loss

$$\mathcal{L}_{\text{MMLDA}}(Z(x), y) = -\log\left[\frac{\exp(-\frac{\|z-\mu_y^*\|_2^2}{2})}{\sum_{l\in[L]}\exp(-\frac{\|z-\mu_l^*\|_2^2}{2})}\right] = -\log\left[\frac{\exp(z^\top\mu_y^*)}{\sum_{l\in[L]}\exp(z^\top\mu_l^*)}\right]$$

$$\mathcal{L}_{\text{MMC}}(Z(x), y) = \frac{1}{2}\|z-\mu_y^*\|_2^2$$

- **No softmax normalization**

[Pang et al., Max-Mahalanobis Linear Discriminant Analysis Networks, ICML 2018]

Key results #2: The MMC loss induces a higher sample density locally

Theorem 2. *(Proof in Appendix A.2) Given* $(x, y) \in \mathcal{D}_k$, $z = Z(x)$ *and* $\mathcal{L}_{MMC}(z, y) = C$, *the sample density nearby the feature point* z *is*

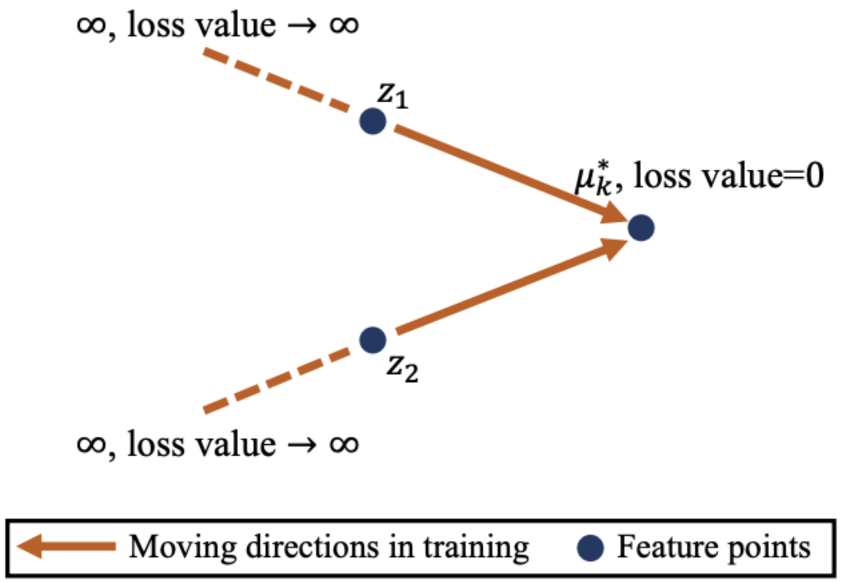
$$\mathbb{SD}(z) \propto \frac{N_k \cdot p_k(C)}{C^{\frac{d-1}{2}}}, \tag{9}$$

where for the input-label pair in $\mathcal{D}_k$, *there is* $\mathcal{L}_{MMC} \sim p_k(c)$.

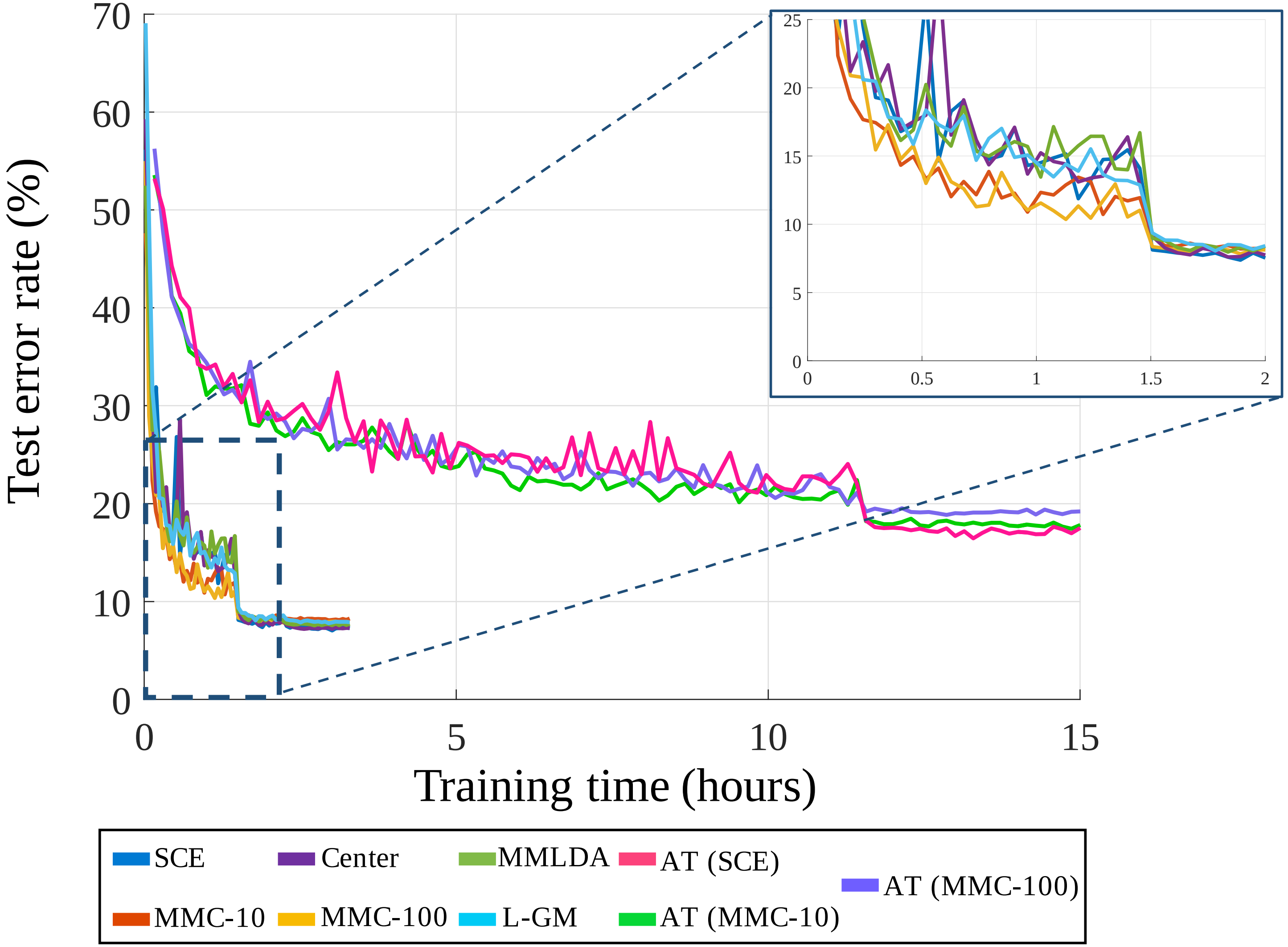
Higher sample density!

$N_k > N_{k,\tilde{k}}$

The sample density will exponentially increase as C gets to 0



Empirical Faster Convergence



MMC loss leads to faster convergence, while keeping comparable performance on the clean images (AT sacrifices clean accuracy)

White-box Robustness (Adaptive Attacks)

Methods	Clean	Perturbation $\epsilon = 8/255$				Perturbation $\epsilon = 16/255$			
		$\text{PGD}_{10}^{\text{tar}}$	$\text{PGD}_{10}^{\text{un}}$	$\text{PGD}_{50}^{\text{tar}}$	$\text{PGD}_{50}^{\text{un}}$	$\text{PGD}_{10}^{\text{tar}}$	$\text{PGD}_{10}^{\text{un}}$	$\text{PGD}_{50}^{\text{tar}}$	$\text{PGD}_{50}^{\text{un}}$
SCE	92.9	≤ 1	3.7	≤ 1	3.6	≤ 1	2.9	≤ 1	2.6
Center loss	92.8	≤ 1	4.4	≤ 1	4.3	≤ 1	3.1	≤ 1	2.9
MMLDA	92.4	≤ 1	16.5	≤ 1	9.7	≤ 1	6.7	≤ 1	5.5
L-GM	92.5	37.6	19.8	8.9	4.9	26.0	11.0	2.5	2.8
MMC-10 (rand)	92.3	43.5	29.2	20.9	18.4	31.3	17.9	8.6	11.6
MMC-10	92.7	**48.7**	**36.0**	**26.6**	**24.8**	**36.1**	**25.2**	**13.4**	**17.5**
$\text{AT}_{10}^{\text{tar}}$ (SCE)	83.7	**70.6**	49.7	**69.8**	47.8	48.4	26.7	31.2	16.0
$\text{AT}_{10}^{\text{tar}}$ (**MMC-10**)	83.0	69.2	**54.8**	67.0	**53.5**	**58.6**	**47.3**	**44.7**	**45.1**
$\text{AT}_{10}^{\text{un}}$ (SCE)	80.9	69.8	55.4	69.4	53.9	53.3	34.1	38.5	21.5
$\text{AT}_{10}^{\text{un}}$ (**MMC-10**)	81.8	**70.8**	**56.3**	**70.1**	**55.0**	**54.7**	**37.4**	**39.9**	**27.7**

CIFAR-10

Adversarial Distributional Training for Robust Deep Learning

(Dong, Deng, Pang, Zhu, Su, NeurIPS 2020)

Adversarial Training

- Adversarial training (AT) is formulated as a minimax optimization problem (Madry et al., 2018)

$$\overbrace{\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \underbrace{\max_{\delta_i \in S} L(f_\theta(x_i+\delta_i), y_i)}_{\text{Inner maximization: generate an adversarial example}}}^{\text{Outer minimization: train a robust classifier}} \qquad S = \{\delta : \|\delta\|_\infty \le \epsilon\}$$

* Adversarial attacks can be used to find an approximate solution, e.g., FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018)

Problem I: Training Speed

- PGD-based adversarial training is much slower than normal training, which cannot be accomplished on **ImageNet** (except Facebook, Google…)
- Free Adversarial Training (Shafahi et al., 2019)
  - Recycling the gradient information computed when updating model parameters
- Fast Adversarial Training (Wong et al., 2020)
  - Use FGSM for training with random initializations, cyclic learning rate, early stopping, etc.

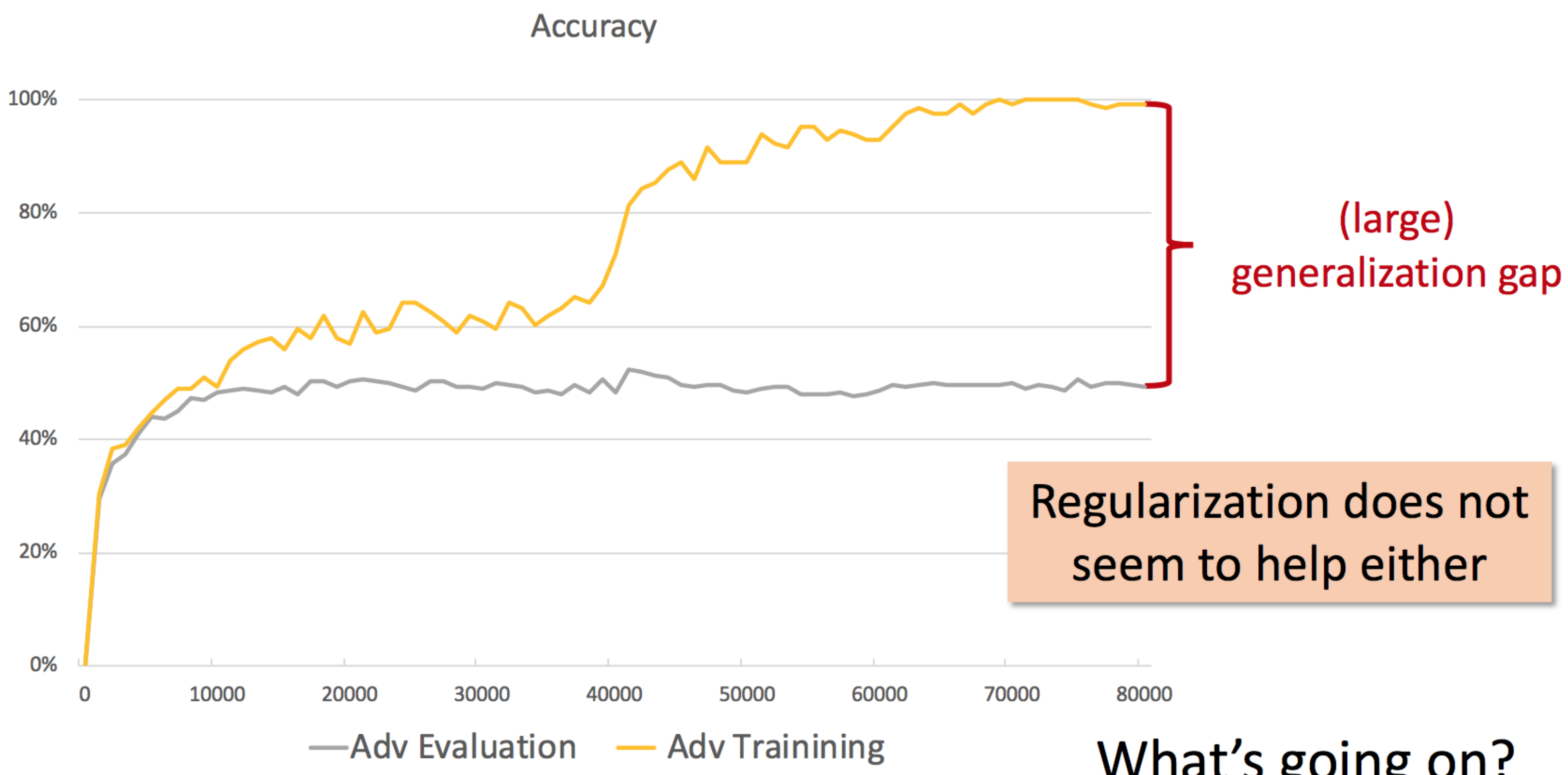
But these methods cannot yield the same level of robustness compared with PGD-based AT on ImageNet.

Problem II: Attack Generalization

- Most AT methods solve the inner maximization using a specific attack, which can result in poor generalization for other attacks under the same threat model.
- Several recent works (Zhang and Wang, 2019) improving AT upon Madry et al. (2018) have this problem.

Model	$\mathcal{A}_{\mathrm{nat}}$	FGSM	PGD-20	PGD-100	MIM	C&W	FeaAttack	$\mathcal{A}_{\mathrm{rob}}$
Standard	**94.81%**	12.05%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
$\mathrm{AT}_{\mathrm{FGSM}}$	93.80%	**79.86%**	0.12%	0.04%	0.06%	0.13%	0.01%	0.01%
$\mathrm{AT}_{\mathrm{PGD}}{}^{\dagger}$	87.25%	56.04%	45.88%	45.33%	47.15%	46.67%	46.01%	44.89%
$\mathrm{AT}_{\mathrm{PGD}}$	86.91%	58.30%	50.03%	49.40%	51.40%	50.23%	50.46%	48.26%
ALP	86.81%	56.83%	48.97%	48.60%	50.13%	49.10%	48.51%	47.90%
FeaScatter	89.98%	**77.40%**	**70.85%**	**68.81%**	**72.74%**	**58.46%**	37.45%	37.40%

Problem III: Large Generalization Gap



Regularization does not seem to help either

What's going on?

(Figure from https://media.neurips.cc/Conferences/NIPS2018/Slides/adversarial_ml_slides_parts_1_4.pdf)

Adversarial Distributional Training

- We formulate adversarial distributional training (ADT) as a different minimax optimization problem

$$P = \{p: \mathrm{supp}(p) \subseteq S\}$$

$$\overbrace{\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \underbrace{\max_{p(\delta_i)\in P} \mathbb{E}_{p(\delta_i)}[L(f_\theta(x_i+\delta_i), y_i)]}_{\text{Inner maximization: learn an adversarial distribution}}}^{\text{Outer minimization: train a robust classifier}}$$

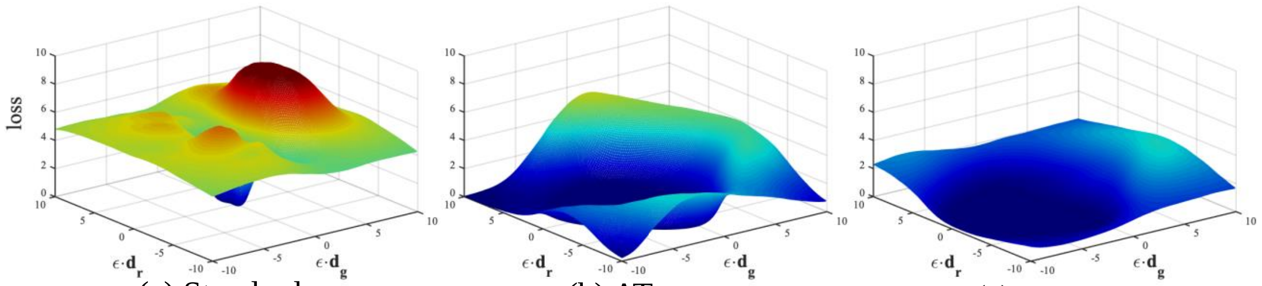
- To prevent ADT from degenerating into AT, we add an entropic regularizer

$$\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \max_{p(\delta_i)\in P} J(p(\delta_i),\theta); \qquad J(p(\delta_i),\theta) = \mathbb{E}_{p(\delta_i)}[L(f_\theta(x_i+\delta_i), y_i)] + \lambda H\big(p(\delta_i)\big)$$

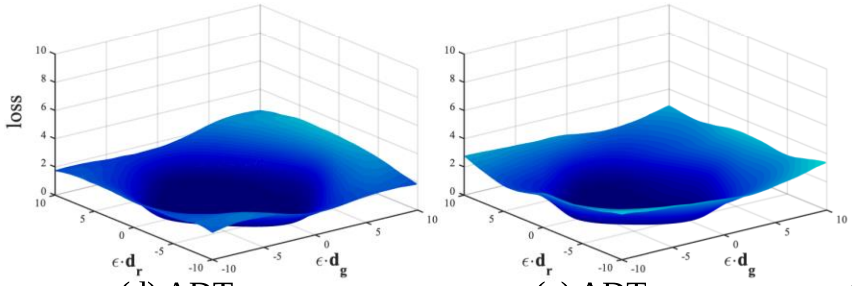
[Dong et al., Adversarial distributional training for robust deep learning, NeurIPS 2020]

Advantages

- Better generalization across attacks
- Better model robustness (more flattened loss surfaces in the vicinity of a nature input)



(a) Standard (b) $\mathrm{AT}_{\mathrm{PGD}}$ (c) $\mathrm{ADT}_{\mathrm{EXP}}$



(d) $\mathrm{ADT}_{\mathrm{EXP\text{-}AM}}$ (e) $\mathrm{ADT}_{\mathrm{IMP\text{-}AM}}$

Model	Dominant eigenvalue
Standard	1.8301±6.3663
$\mathrm{AT}_{\mathrm{PGD}}$	0.0242±0.0478
$\mathrm{ADT}_{\mathrm{EXP}}$	**0.0180**±0.0311
$\mathrm{ADT}_{\mathrm{EXP\text{-}AM}}$	0.0181±**0.0270**
$\mathrm{ADT}_{\mathrm{IMP\text{-}AM}}$	0.0211±0.0353

(f) Comparison on the Hessian

[Dong et al., Adversarial distributional training for robust deep learning, NeurIPS 2020]

Benchmarking Adversarial Robustness of Image Classification

(Dong, Fu, Yang, Pang, Su, Xiao, Zhu, CVPR 2020, Oral) https://github.com/thu-ml/ares

Platform: Ares

Attacks

Adaptive attacks [Athalye et al., 2018]

Optimization-based attacks [Carlini and Wagner, 2017]

Iterative attacks[kurakin et al., 2016]

One-step attacks [Goodfellow et al., 2014]

Defenses

Randomization, denoising [Xie et al., 2018; Liao et al., 2018]

Defensive distillation [Papernot et al., 2016]

Adversarial training with FGSM [Kurakin et al., 2015]

- We developed `Ares`, a platform for adversarial machine learning research focusing on benchmarking adversarial robustness on image classification
- Support all attacks in various threat models;
- Provide ready-to-use pre-trained baseline models (8 on ImageNet & 8 on CIFAR10);
- Provide efficient & easy-to-use tools for benchmarking models.

(Dong, Fu,Yang, Pang, Su, Xiao, Zhu, CVPR 2020, Oral)

Attacks in our Benchmark

https://github.com/thu-ml/ares

Attack Method	Knowledge	Goal	Capability	Distance Metrics
FGSM [Goodfellow et al., 2015]	**white-box** & **transfer-based**	untargeted & targeted	constrained	l_2, l_∞
BIM [Kurakin et al., 2017]	**white-box** & **transfer-based**	untargeted & targeted	constrained	l_2, l_∞
MIM [Dong et al., 2018]	**white-box** & **transfer-based**	untargeted & targeted	constrained	l_2, l_∞
DeepFool [Moosavi-Dezfooli et al., 2016]	**white-box**	untargeted	optimized	l_2, l_∞
C&W [Carlini & Wagner, 2017]	**white-box**	untargeted & targeted	optimized	l_2
DIM [Xie et al., 2019]	**transfer-based**	untargeted & targeted	constrained	l_2, l_∞
ZOO [Chen et al., 2017]	**score-based**	untargeted & targeted	optimized	l_2
NES [Ilyas et al., 2018]	**score-based**	untargeted & targeted	constrained	l_2, l_∞
SPSA [Uesato et al., 2018]	**score-based**	untargeted & targeted	constrained	l_2, l_∞
NATTACK [Li et al., 2019]	**score-based**	untargeted & targeted	constrained	l_2, l_∞
Boundary [Brendel et al., 2018]	**decision-based**	untargeted & targeted	optimized	l_2
Evolutionary [Dong et al., 2019]	**decision-based**	untargeted & targeted	optimized	l_2

Defenses in our Benchmark: CIFAR-10 & ImageNet

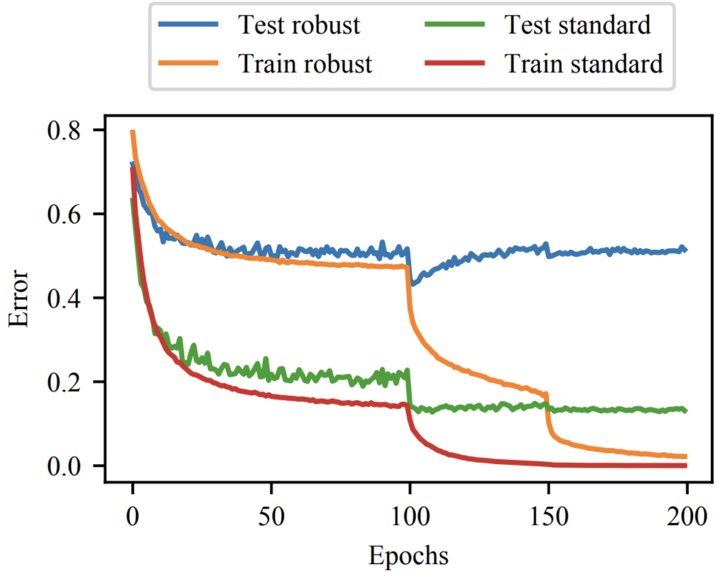
https://github.com/thu-ml/ares

Model	Category	Intended Threat Model	Accuracy (%)
Res-56 [He et al., 2016]	Natural training	-	92.6
PGD-AT [Madry et al., 2018]	Robust training	l_∞ ($\epsilon = 8/255$)	87.3
DeepDefense [Yan et al., 2018]	Robust training	l_2	79.7
TRADES [Zhang et al., 2019]	Robust training	l_∞ ($\epsilon = 0.031$)	84.9
Convex [Wong et al., 2018]	Certified robust training	l_∞ ($\epsilon = 2/255$)	66.3
JPEG [Dziugaite et al., 2016]	Input transformation	General	80.9
RSE [Liu et al., 2018]	Randmization & ensemble	l_2	86.1
ADP [Pang et al., 2019]	Ensemble	General	94.1

Model	Category	Intended Threat Model	Accuracy (%)
Inc-v3 [Szegedy et al., 2016]	Natural training	-	78.0
Ens-AT [Tramer et al., 2018]	Robust training	l_∞ ($\epsilon = 16/255$)	73.5
ALP [Kannan et al., 2018]	Robust training	l_∞ ($\epsilon = 16/255$)	49.0
FD [Xie et al., 2019]	Robust training	l_∞ ($\epsilon = 16/255$)	64.3
JPEG [Dziugaite et al., 2016]	Input transformation	General	77.3
Bit-Red [Xu et al., 2018]	Input transformation	General	61.8
R&P [Xie et al., 2018]	Randomization	General	77.0
RandMix [Zhang & Liang., 2019]	Certified randomization	General	52.4

A Case Study: AT has inconsistent results …

Rice et al. (ICML 2020) find that simply early stopping the training process of PGD-AT can attain the gains from almost all the previously proposed improvements, including state-of-the-art TRADES.



(From Rice et al. 2020)

- *TRADES also applied early stopping by decaying learning rate at 75th epoch and used the checkpoint of 76th epoch.*

Who is wrong?

Gowal et al. (2020) find that TRADES actually performs better than PGD-AT

Training settings in previous work are highly inconsistent

Method	l.r.	Total epoch (l.r. decay)	Batch size	Weight decay	Early stop (train / attack)	Warm-up (l.r. / pertub.)
Madry et al. (2018)	0.1	200 (100, 150)	128	2×10^{-4}	No / No	No / No
Cai et al. (2018)	0.1	300 (150, 250)	200	5×10^{-4}	No / No	No / Yes
Zhang et al. (2019b)	0.1	76 (75)	128	2×10^{-4}	Yes / No	No / No
Wang et al. (2019)	0.01	120 (60, 100)	128	1×10^{-4}	No / Yes	No / No
Qin et al. (2019)	0.1	110 (100, 105)	256	2×10^{-4}	No / No	No / Yes
Mao et al. (2019)	0.1	80 (50, 60)	50	2×10^{-4}	No / No	No / No
Carmon et al. (2019)	0.1	100 (cosine anneal)	256	5×10^{-4}	No / No	No / No
Alayrac et al. (2019)	0.2	64 (38, 46, 51)	128	5×10^{-4}	No / No	No / No
Shafahi et al. (2019b)	0.1	200 (100, 150)	128	2×10^{-4}	No / No	No / No
Zhang et al. (2019a)	0.05	105 (79, 90, 100)	256	5×10^{-4}	No / No	No / No
Zhang & Wang (2019)	0.1	200 (60, 90)	60	2×10^{-4}	No / No	No / No
Atzmon et al. (2019)	0.01	100 (50)	32	1×10^{-4}	No / No	No / No
Wong et al. (2020)	0∼0.2	30 (one cycle)	128	5×10^{-4}	No / No	Yes / No
Rice et al. (2020)	0.1	200 (100, 150)	128	5×10^{-4}	Yes / No	No / No
Ding et al. (2020)	0.3	128 (51, 77, 102)	128	2×10^{-4}	No / No	No / No
Pang et al. (2020a)	0.01	200 (100, 150)	50	1×10^{-4}	No / No	No / No
Zhang et al. (2020)	0.1	120 (60, 90, 110)	128	2×10^{-4}	No / Yes	No / No
Huang et al. (2020)	0.1	200 (cosine anneal)	256	5×10^{-4}	No / No	Yes / No
Cheng et al. (2020)	0.1	200 (80, 140, 180)	128	5×10^{-4}	No / No	No / No
Lee et al. (2020)	0.1	200 (100, 150)	128	2×10^{-4}	No / No	No / No
Xu et al. (2020)	0.1	120 (60, 90)	256	1×10^{-4}	No / No	No / No

[Pang et al., Bag of tricks for Adversarial Training, ICLR 2021] **Code:** https://github.com/P2333/Bag-of-Tricks-for-AT

Takeaways through Extensive Benchmarking

Takeaways:
(i) Slightly different values of weight decay could largely affect the robustness of trained models;
(ii) Moderate label smoothing and linear scaling rule on l.r. for different batch sizes are beneficial;
(iii) Applying eval BN mode to craft training adversarial examples can avoid blurring the distribution;
(iv) Early stopping the adversarial steps or perturbation may degenerate worst-case robustness;
(v) Smooth activation benefits more when the model capacity is not enough for adversarial training.

- **Adversarial training is more sensitive to these usually overlooked hyperparameters, compared to standard training.**
- **Standardize the basic training setting enables fairer benchmarks.**

[Pang et al., Bag of tricks for Adversarial Training, ICLR 2021] **Code:** https://github.com/P2333/Bag-of-Tricks-for-AT

ADVERSARIAL ROBUSTNESS BENCHMARK

http://ml.cs.tsinghua.edu.cn/adv-bench/

The goal of the adversarial robustness benchmark is to provide a comprehensive comparison of adversarial defense models. These models are evaluated against various attacks developed by research and during the CVPR 2021 competition of white-box adversarial attacks on ML defense models. We welcome contributions to both robust models and effective attacks.

This is the temporary benchmark result. We will incorporate the top attack solutions in this competition in this benchmark (at about April 2021).

Defense Leaderboard | Attack Leaderboard

Defense Leaderboard: CIFAR-10, Untargeted (epsilon=8/255)

Method	Clean	PGD-100	CW-100	MIM-100	Overall Robust Accuracy
Towards Deep Learning Models Resistant to Adversarial Attacks	87.25%	45.33%	46.61%	46.21%	45.18%
Unlabeled Data Improves Adversarial Robustness	89.69%	62.30%	60.97%	62.90%	60.23%
Theoretically Principled Trade-off between Robustness and Accuracy	84.92%	55.09%	53.73%	55.53%	53.10%

Summary

- Adversarial robustness is a crucial issue of deep learning for safety-critical applications

- Much progress has been done on adversarial attack, including program synthesis for automated attack
  - E.g., AutoDA (Fu et al., USENIX Security Symposium 2022)

- Defending over adversarial attack requires a deep investigatio learning objectives, uncertainty, theory, evaluation, etc.
  - E.g., certified defense against semantic transformations (Hao effect of adversarial training (Dong et al., NeurIPS 2022)

- Robustness is closely related to interpretability, privacy, OoD

- **Upcoming book on AI Safety, stay tuned …**

Thank you!