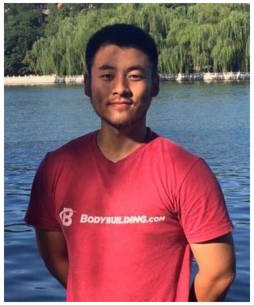
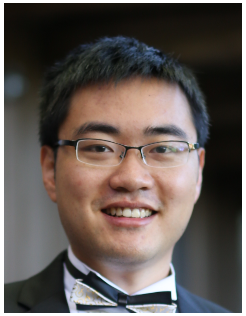
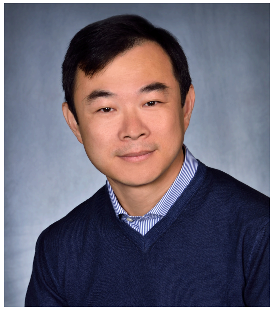
Theoretically Principled Trade-off between Robustness and Accuracy

Hongyang Zhang, CMU → TTIC

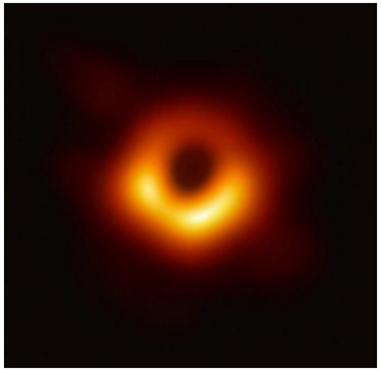
Yaodong Yu (UVa) Jiantao Jiao (UCB) Eric Xing (CMU) Laurent Ghaoui (UCB) Mike Jordan (UCB)

Deep Geometric Learning of Big Data and Applications

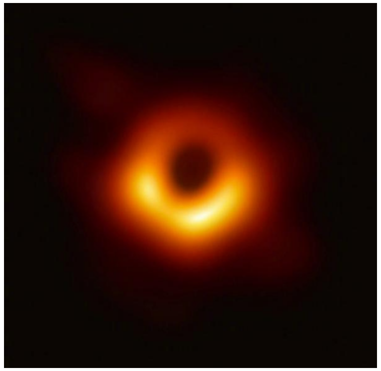
May 21st, 2019

Deep networks are unsafe



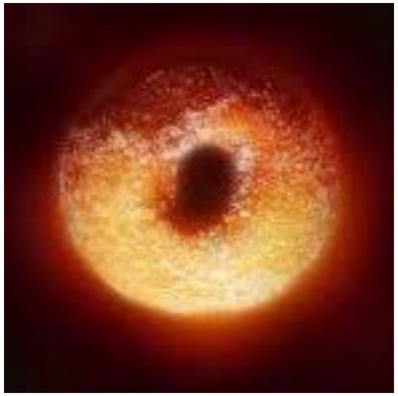
$+ .007 \times$

$=$

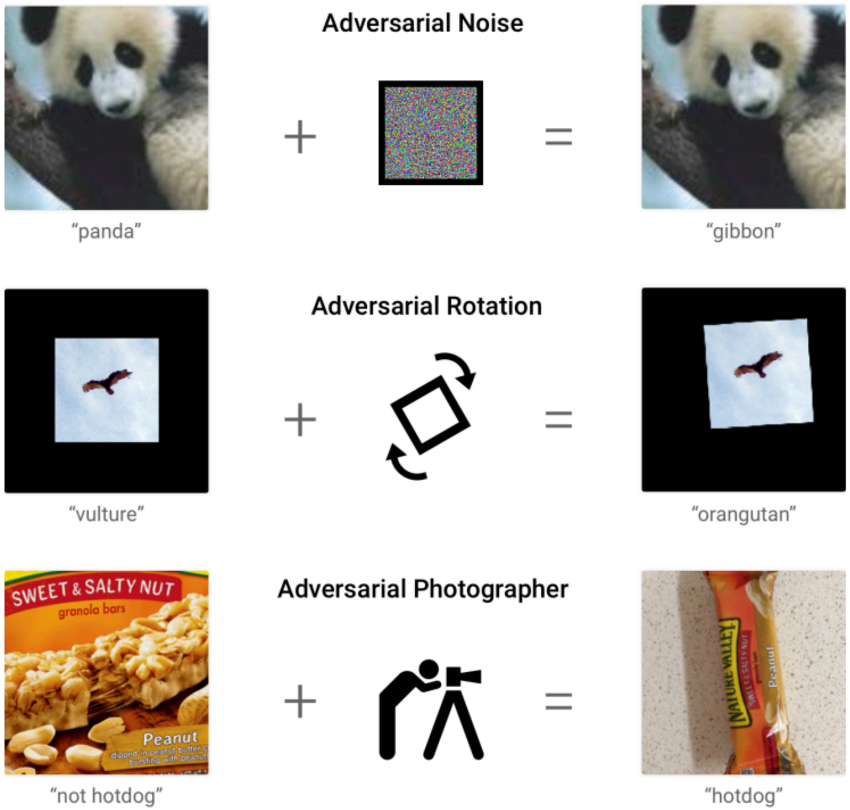


"black hole"
87.7% confidence

"donut"
99.3% confidence



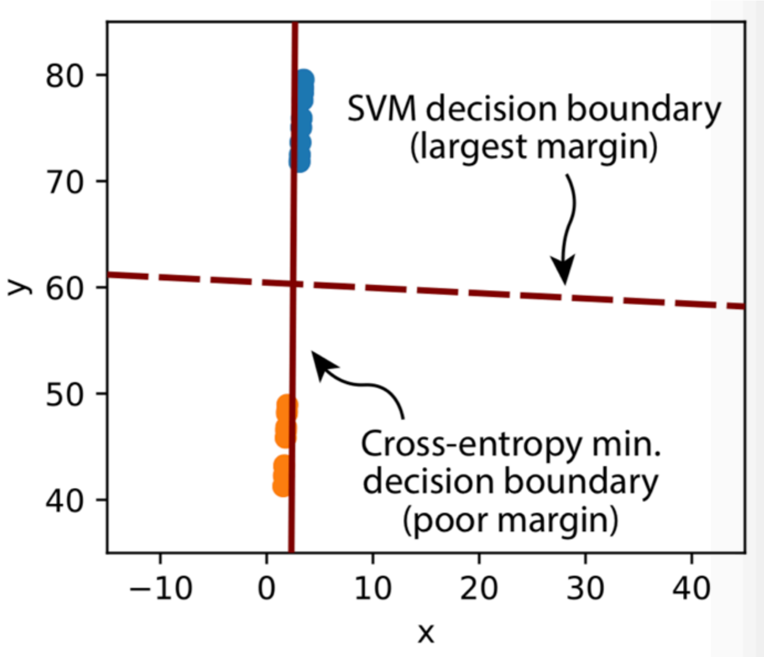
Deep networks are unsafe



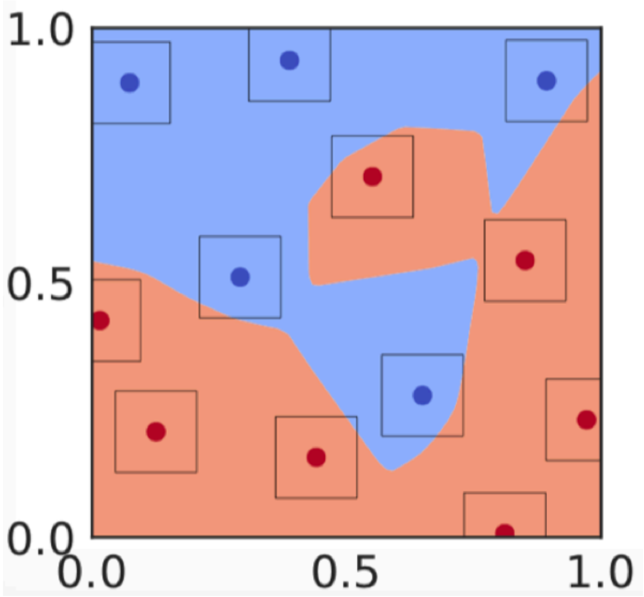
[BCZOCG'18] Unrestricted Adversarial Example, 2018

Why are there adversarial examples?

- We use a wrong loss function



Linear Case

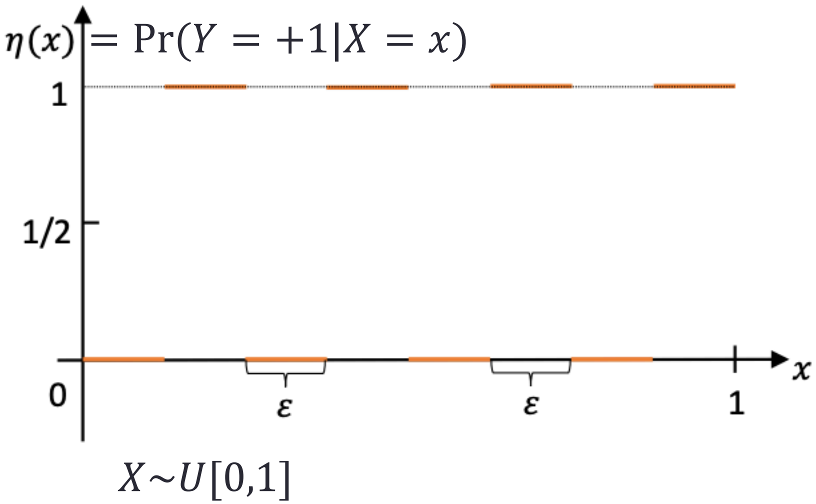


Non-Linear Case

Trade-off between Robustness and Accuracy

$$R_{rob}(f) := \mathbb{E}_{(X,Y)\sim D} 1\{\exists X' \in \mathbb{B}(X,\varepsilon)\ s.t.\ f(X')Y \leq 0\}$$
$$R_{nat}(f) := \mathbb{E}_{(X,Y)\sim D} 1\{f(X)Y \leq 0\}$$

- An example of trade-off:



	Bayes Optimal Classifier	All-One Classifier
$\mathcal{R}_{\text{nat}}$	0 (optimal)	1/2
$\mathcal{R}_{\text{rob}}$	1	1/2 (optimal)

Trade-off between Robustness and Accuracy

- Our goal: Find a classifier $\hat{f}$ such that $R_{rob}(\hat{f}) \leq \mathrm{OPT} + \delta$

$$\mathrm{OPT} := \min_{f} R_{rob}(f), \qquad \text{s.t.} \qquad R_{nat}(f) \leq R_{nat}^{*} + \delta$$

$$\Downarrow$$

suffice to show $\boxed{R_{rob}(f) - R_{nat}^{*}} \leq \delta$

Computationally, both $R_{nat}(f)$ and $R_{rob}(f)$ are non-differentiable.

Surrogate Loss

- Classification-calibrated loss ϕ:

$$H(\eta) := \min_{\alpha \in \mathbb{R}} (\eta\phi(\alpha) + (1-\eta)\phi(-\alpha))$$

$$H^{-}(\eta) := \min_{\alpha:\alpha(2\eta-1)\leq 0} (\eta\phi(\alpha) + (1-\eta)\phi(-\alpha))$$

Definition (classification-calibrated loss):

ϕ is classification-calibrated loss, if for any $\eta \neq 1/2$, $H^{-}(\eta) > H(\eta)$.

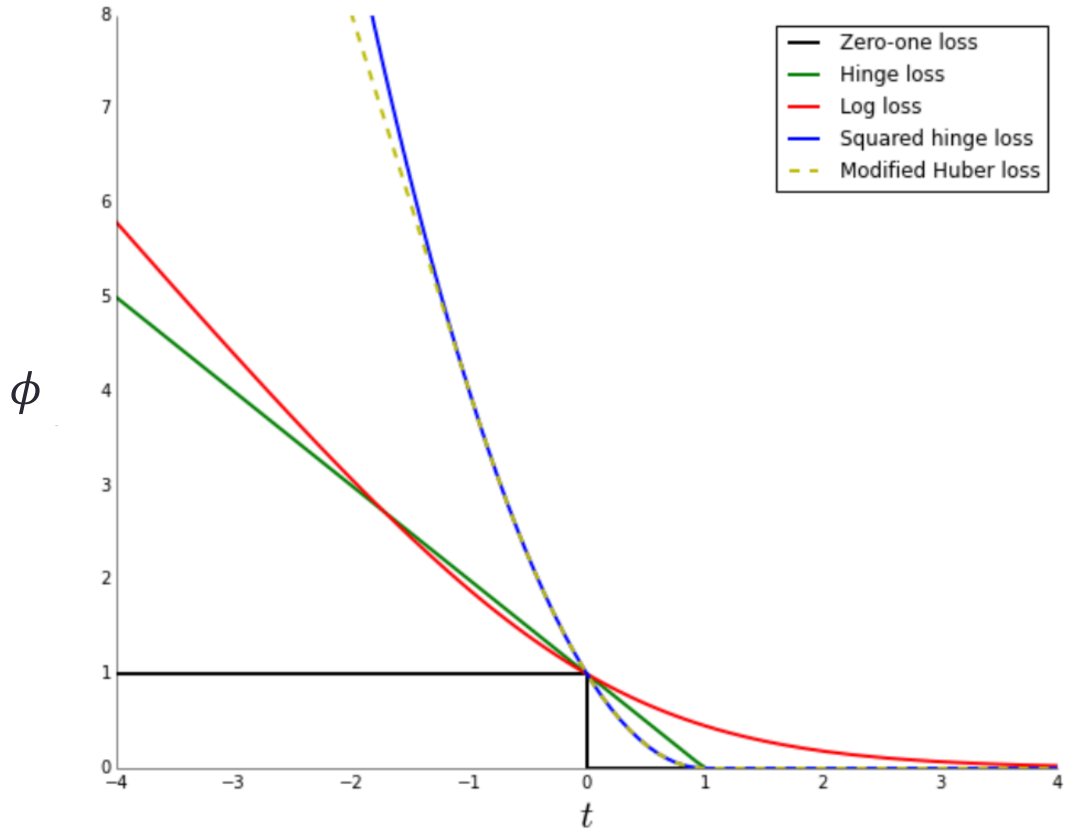
Intuitive explanation:

- Think about η as $\eta(x) = \Pr[Y = +1 | X = x]$, and α as score of positive class by f
- Then $H(\eta) = \min_f R_{nat}(f)$

  $H^{-}(\eta) = \min_f R_{nat}(f)$ s.t. f is inconsistent with Bayes optimal classifier
- Classification-calibrated loss: wrong classifier leads to larger loss for all $\eta(x)$

[BJM'06] Convexity, Classification, and Risk Bounds, 2006

Surrogate Loss



[BJM’06] Convexity, Classification, and Risk Bounds, 2006

Main Results

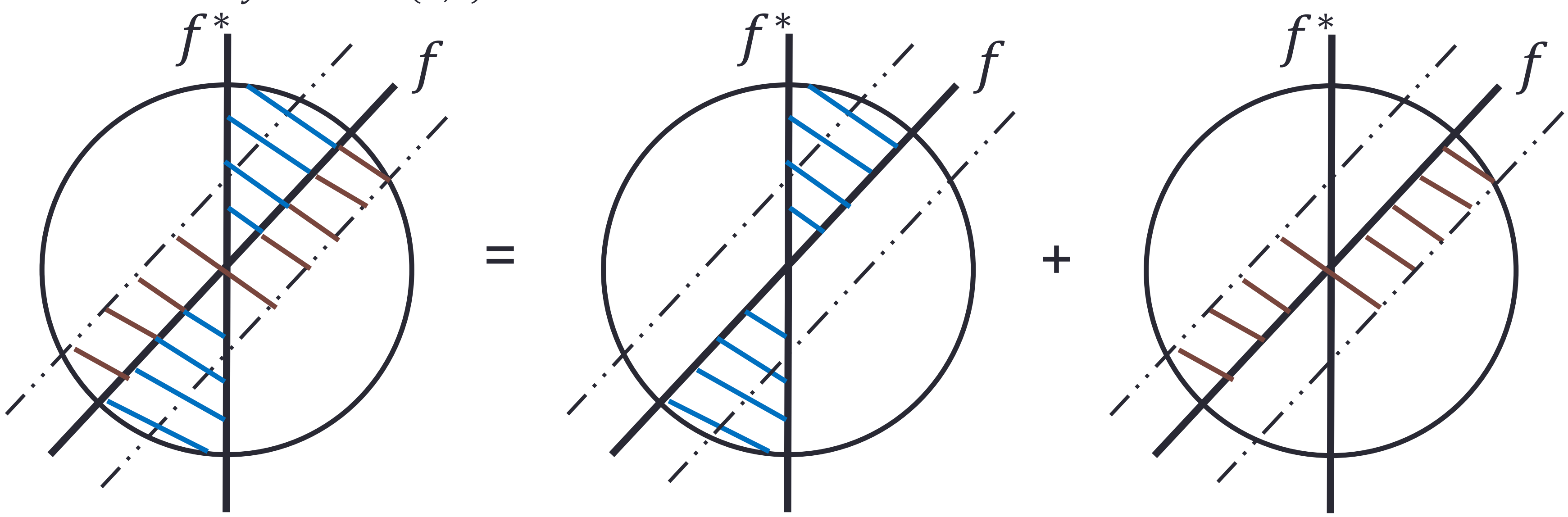
Theorem 1 (Informal, upper bound, ZYJXGJ'19):

We have $R_{rob}(f) - R_{nat}^* \leq R_\phi(f) - R_\phi^* + \mathbb{E} \max_{X' \in \mathbb{B}(X,\varepsilon)} \phi(f(X')f(X)/\lambda)$.

Proof Sketch:

- An important decomposition: $R_{rob}(f) = R_{nat}(f) + R_{bdy}(f)$

  where $R_{bdy}(f) = \mathbb{E}_{(X,Y)\sim D} 1\{\exists X \in \varepsilon \text{ neighbour of } f \text{ s.t. } f(X)Y > 0\}$



[ZYJXGJ'19] Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2019

Main Results

Theorem 1 (Informal, upper bound, ZYJXGJ'19):

We have $R_{rob}(f) - R_{nat}^* \leq R_\phi(f) - R_\phi^* + \mathbb{E} \max_{X' \in \mathbb{B}(X,\varepsilon)} \phi(f(X')f(X)/\lambda)$.

Proof Sketch:

- An important decomposition: $R_{rob}(f) = R_{nat}(f) + R_{bdy}(f)$
  where $R_{bdy}(f) = \mathbb{E}_{(X,Y)\sim D} 1\{\exists X \in \varepsilon \text{ neighbour of } f \text{ s.t. } f(X)Y > 0\}$
- $R_{rob}(f) - R_{nat}^* = R_{nat}(f) - R_{nat}^* + R_{bdy}(f)$
- $R_{nat}(f) - R_{nat}^* \leq R_\phi(f) - R_\phi^*$ by [BJM'06]
- $R_{bdy}(f) = \mathbb{E} \max_{X' \in \mathbb{B}(X,\varepsilon)} 1(f(X')f(X) < 0) \leq \mathbb{E} \max_{X' \in \mathbb{B}(X,\varepsilon)} \phi(f(X')f(X)/\lambda)$

[BJM'06] Convexity, Classification, and Risk Bounds, 2006

[ZYJXGJ'19] Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2019

Main Results

Theorem 1 (Informal, upper bound, ZYJXGJ'19):

We have $R_{rob}(f) - R^*_{nat} \le R_\phi(f) - R^*_\phi + \mathbb{E} \max_{X' \in \mathbb{B}(X,\varepsilon)} \phi(f(X')f(X)/\lambda)$.

Theorem 2 (Informal, lower bound, ZYJXGJ'19):

There exist a data distribution, a classifier f, and an $\lambda > 0$ such that

$$R_{rob}(f) - R^*_{nat} \ge R_\phi(f) - R^*_\phi + \mathbb{E} \max_{X' \in \mathbb{B}(X,\varepsilon)} \phi(f(X')f(X)/\lambda).$$

[ZYJXGJ'19] Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2019

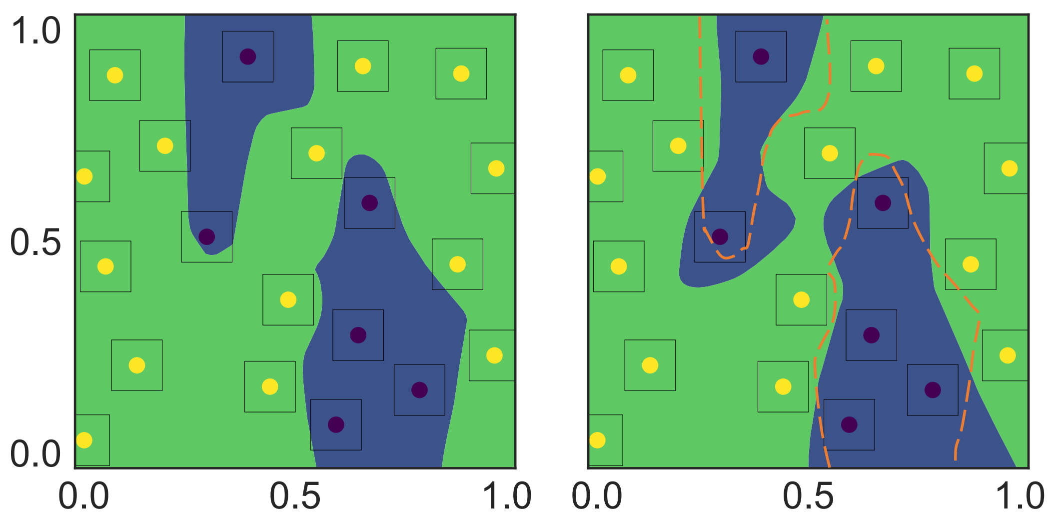
Main Results

Theorem 1 (Informal, upper bound, ZYJXGJ'19):

We have $R_{rob}(f) - R^*_{nat} \le R_\phi(f) - R^*_\phi + \mathbb{E} \max_{X' \in \mathbb{B}(X,\varepsilon)} \phi(f(X')f(X)/\lambda)$.

- New Surrogate Loss:

$$\min_f [\mathbb{E}\, \phi\big(Yf(X)\big) + \mathbb{E} \max_{X' \in B_\varepsilon(X)} \phi(f(X)f(X')/\lambda)]$$



$\min_f \mathbb{E}\, \phi\big(Yf(X)\big)$ $\qquad$ $\min_f [\mathbb{E}\, \phi\big(Yf(X)\big) + \mathbb{E} \max_{X' \in B_\varepsilon(X)} \phi(f(X)f(X')/\lambda)]$

[ZYJXGJ'19] Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2019

PyTorch Package

- New Surrogate Loss:

$$\min_f [\mathbb{E}\,\phi(Yf(X)) + \mathbb{E} \max_{X' \in B_\varepsilon(X)} \phi(f(X)f(X')/\lambda)]$$

Natural training:

```
def train(args, model, device, train_loader, optimizer, epoch):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        loss = F.cross_entropy(model(data), target)
        loss.backward()
        optimizer.step()
```

replace

Adversarial training by TRADES:

To apply TRADES, cd into the directory, put 'trades.py' to the directory.

```
from trades import trades_loss

def train(args, model, device, train_loader, optimizer, epoch):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        # calculate robust loss - TRADES loss
        loss = trades_loss(model=model,
                           x_natural=data,
                           y=target,
                           optimizer=optimizer,
                           step_size=args.step_size,
                           epsilon=args.epsilon,
                           perturb_steps=args.num_steps,
                           batch_size=args.batch_size,
                           beta=args.beta,
                           distance='l_inf')
        loss.backward()
        optimizer.step()
```

- Link: https://github.com/yaodongyu/TRADES

Significant Experimental Results

Experiments --- CIFAR10

Defense	Defense type	Under which attack	Dataset	Distance	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$
[BRRG18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	0%
[MLW$^+$18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	5%
[DAL$^+$18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	0%
[SKN$^+$18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	9%
[NKM17]	gradient mask	[ACW18]	CIFAR10	0.015 (ℓ_∞)	-	15%
[WSMK18]	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	27.07%	23.54%
[MMS$^+$18]	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	87.30%	**47.04%**

$$\min_f \max_{X' \in B_\varepsilon(X)} \phi(Yf(X'))$$ (by Madry et al.)

TRADES ($1/\lambda = 1$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	88.64%	49.14%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	84.92%	**56.61%**

$$\min_f [\mathbb{E}\,\phi(Yf(X)) + \mathbb{E} \max_{X' \in B_\varepsilon(X)} \phi(f(X)f(X'))/\lambda]$$ (ours)

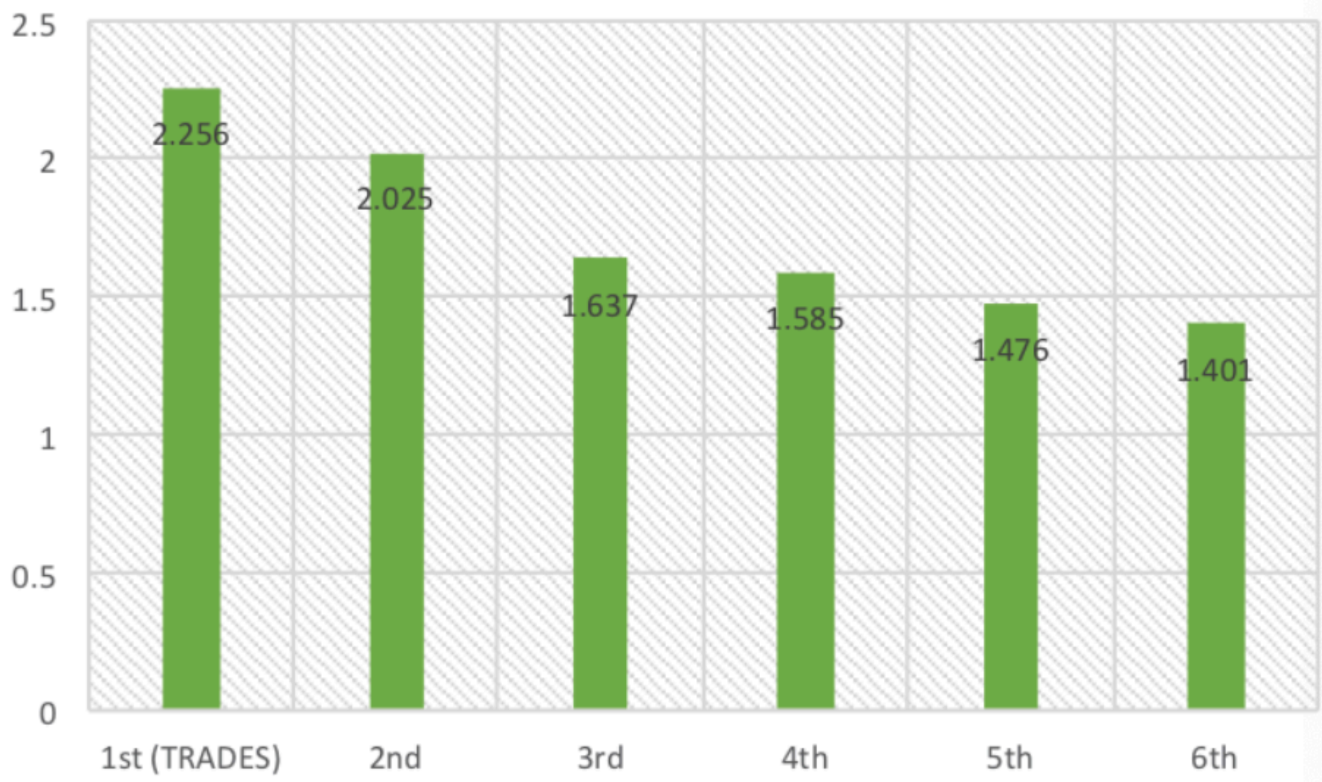
TRADES ($1/\lambda = 6$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	84.92%	81.58%
TRADES ($1/\lambda = 1$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	88.64%	51.26%
TRADES ($1/\lambda = 6$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	84.92%	57.95%
TRADES ($1/\lambda = 1$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	88.64%	84.03%
TRADES ($1/\lambda = 6$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	84.92%	81.24%
[SKC18]	gradient mask	[ACW18]	MNIST	0.005 (ℓ_2)	-	55%
[MMS$^+$18]	robust opt.	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.36%	96.01%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.48%	96.07%
TRADES ($1/\lambda = 6$)	regularization	C&W	MNIST	0.005 (ℓ_2)	99.48%	99.46%

Competition I: NeurIPS 2018 Adversarial Vision Challenge

- Evaluation criterion
  - 400+ teams, ~2,000 submissions
  - Tiny ImageNet dataset
  - Model Track and Attack Track
  - Participants in the two tracks play against each other

Competition I: NeurIPS 2018 Adversarial Vision Challenge

- Final Result



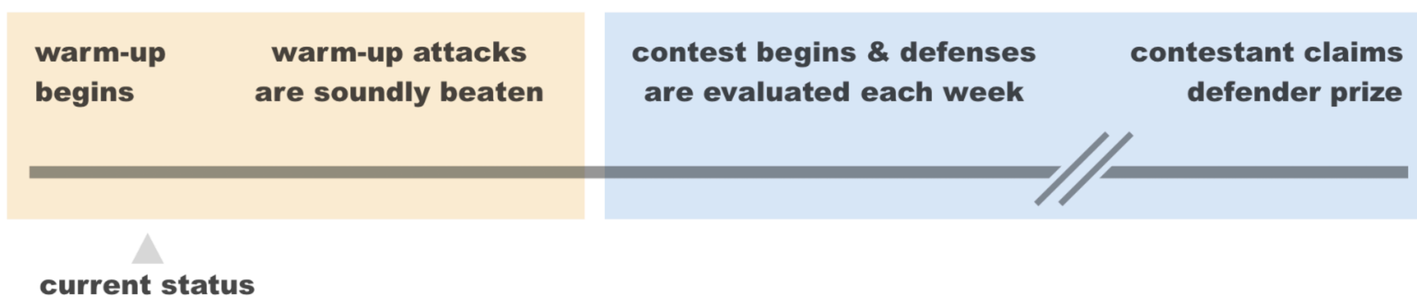
Competition II: Unrestricted Adversarial Example

Unrestricted Adversarial Examples Challenge build passing

In the Unrestricted Adversarial Examples Challenge, attackers submit arbitrary adversarial inputs, and defenders are expected to assign low confidence to difficult inputs while retaining high confidence and accuracy on a clean, unambiguous test set. You can learn more about the motivation and structure of the contest in our recent paper

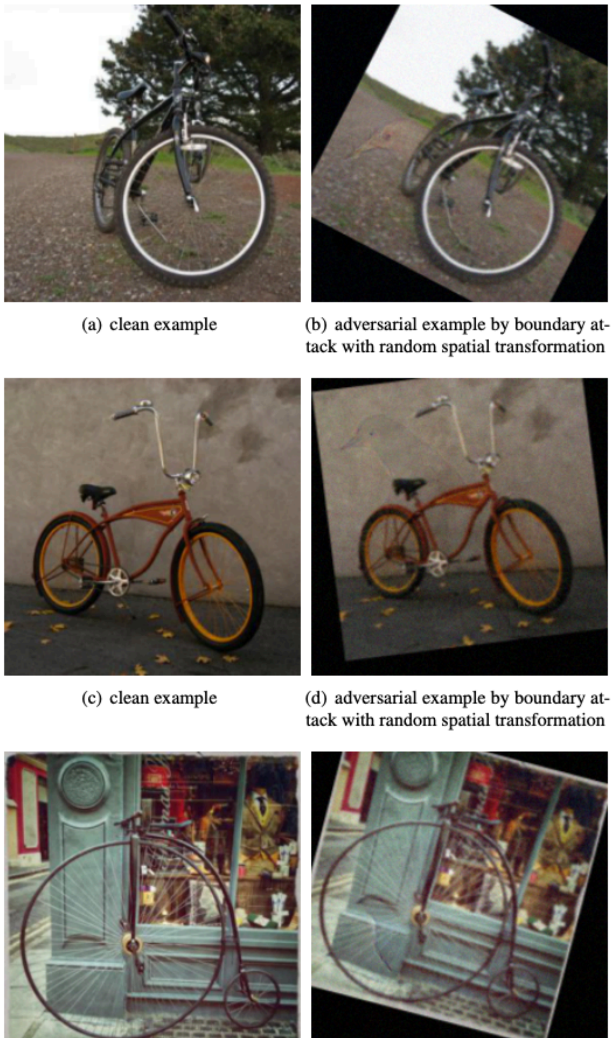
This repository contains code for the warm-up to the challenge, as well as the public proposal for the contest. We are currently accepting defenses for the warm-up.

Warm-up & Contest Timeline



Interpretability

the class of bicycle



(a) clean example

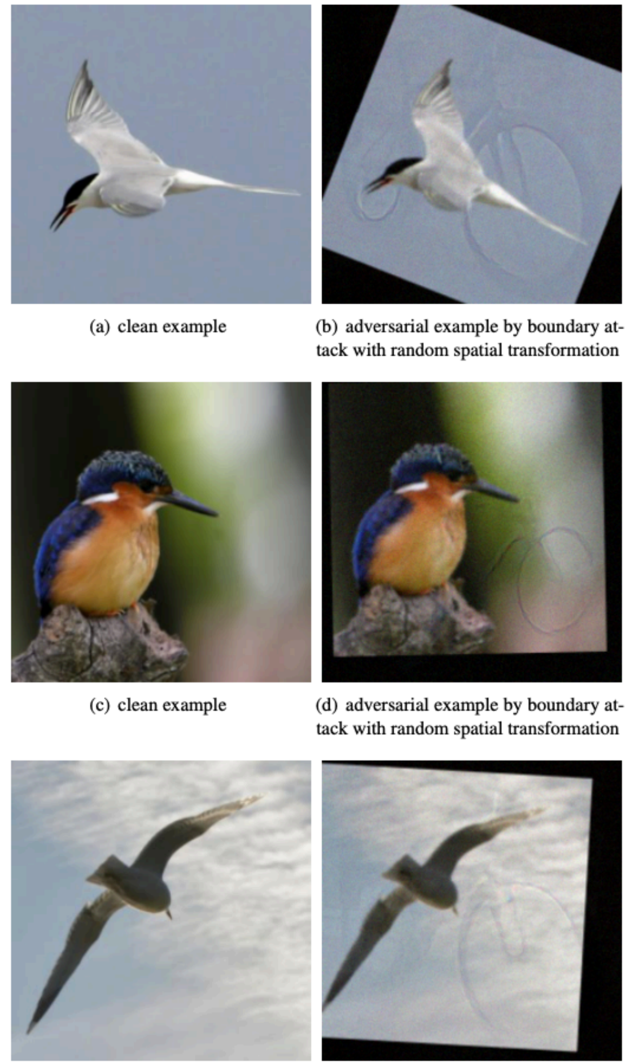
(b) adversarial example by boundary attack with random spatial transformation

(c) clean example

(d) adversarial example by boundary attack with random spatial transformation

(e) clean example

(f) adversarial example by boundary attack with random spatial transformation



(a) clean example

(b) adversarial example by boundary attack with random spatial transformation

(c) clean example

(d) adversarial example by boundary attack with random spatial transformation

(e) clean example

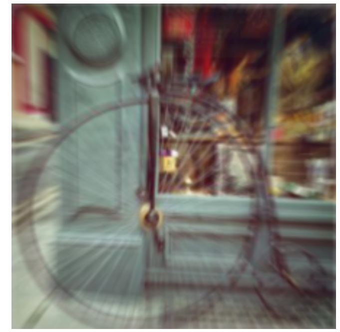
(f) adversarial example by boundary attack with random spatial transformation

the class of bird

Competition II: Unrestricted Adversarial Example

Defense	Submitted by	Clean data	Common corruptions	Spatial grid attack	SPSA attack	Boundary attack	Submission Date
Pytorch ResNet50 (trained on bird-or-bicycle extras)	TRADESv2	100.0%	100.0%	99.5%	100.0%	95.0%	Jan 17th, 2019 (EST)
Keras ResNet (trained on ImageNet)	Google Brain	100.0%	99.2%	92.2%	1.6%	4.0%	Sept 29th, 2018
Pytorch ResNet (trained on bird-or-bicycle extras)	Google Brain	98.8%	74.6%	49.5%	2.5%	8.0%	Oct 1st, 2018

Competition II: Unrestricted Adversarial Example



Defense	Submitted by	Clean data	Common corruptions	Spatial grid attack	SPSA attack	Boundary attack	Submission Date
Pytorch ResNet50 (trained on bird-or-bicycle extras)	TRADESv2	100.0%	100.0%	99.5%	100.0%	95.0%	Jan 17th, 2019 (EST)
Keras ResNet (trained on ImageNet)	Google Brain	100.0%	99.2%	92.2%	1.6%	4.0%	Sept 29th, 2018
Pytorch ResNet (trained on bird-or-bicycle extras)	Google Brain	98.8%	74.6%	49.5%	2.5%	8.0%	Oct 1st, 2018

Competition II: Unrestricted Adversarial Example

Defense	Submitted by	Clean data	Common corruptions	Spatial grid attack	SPSA attack	Boundary attack	Submission Date
Pytorch ResNet50 (trained on bird-or-bicycle extras)	TRADESv2	100.0%	100.0%	99.5%	100.0%	95.0%	Jan 17th, 2019 (EST)
Keras ResNet (trained on ImageNet)	Google Brain	100.0%	99.2%	92.2%	1.6%	4.0%	Sept 29th, 2018
Pytorch ResNet (trained on bird-or-bicycle extras)	Google Brain	98.8%	74.6%	49.5%	2.5%	8.0%	Oct 1st, 2018

Competition II: Unrestricted Adversarial Example


Defense	Submitted by	Clean data	Common corruptions	Spatial grid attack	SPSA attack	Boundary attack	Submission Date
Pytorch ResNet50 (trained on bird-or-bicycle extras)	TRADESv2	100.0%	100.0%	99.5%	100.0%	95.0%	Jan 17th, 2019 (EST)
Keras ResNet (trained on ImageNet)	Google Brain	100.0%	99.2%	92.2%	1.6%	4.0%	Sept 29th, 2018
Pytorch ResNet (trained on bird-or-bicycle extras)	Google Brain	98.8%	74.6%	49.5%	2.5%	8.0%	Oct 1st, 2018

Competition II: Unrestricted Adversarial Example

Defense	Submitted by	Clean data	Common corruptions	Spatial grid attack	SPSA attack	Boundary attack	Submission Date
Pytorch ResNet50 (trained on bird-or-bicycle extras)	TRADESv2	100.0%	100.0%	99.5%	100.0%	95.0%	Jan 17th, 2019 (EST)
Keras ResNet (trained on ImageNet)	Google Brain	100.0%	99.2%	92.2%	1.6%	4.0%	Sept 29th, 2018
Pytorch ResNet (trained on bird-or-bicycle extras)	Google Brain	98.8%	74.6%	49.5%	2.5%	8.0%	Oct 1st, 2018

Conclusions

- Adversarial Robustness
  - Trade-off matters in the adversarial defense
  - Matching upper and lower bounds on $R_{rob}(f) - R^*_{nat}$
  - New surrogate loss for adversarial defense
  - PyTorch package
  - Winners of NeurIPS 2018 Adversarial Vision Challenge
    Unrestricted Adversarial Example Challenge

Future Directions about Robustness

- Computational and Statistical Theory
  - Understand the optimization principal of new surrogate loss
  - (Tight) sample complexity of adversarial learning
- Applications of AI Security
  - Robotics, autonomous cars
  - Medical diagnose
- Extensions with other frameworks
  - Self-supervised/semi-supervised learning
  - Neural ODE

Thank You