CS 499/579: Trustworthy ML

04.20: Defenses I

Tu/Th 10:00 – 11:50 am

Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL
Secure AI Systems Lab

Heads-up!

- Due dates
  - 4/15: Checkpoint presentation I
- Announcement
  - 4/25: Checkpoint presentation I
    - 15-20 min presentation + 3-5 min Q&A
    - Presentation MUST cover:
      - A research problem your team chose
      - A review of the prior work relevant to your problem
        - How is your team's work different from the prior work?
        - What's the paper your team picked and the results your team will reproduce?
      - Next steps
  - 4/25: Checkpoint review assignments are out!
    - Check the Canvas for your assignment (you will be assigned to one project)

Recap

- Research questions
  - How can we find adversarial examples?
    - Threat model for evasion (test-time) attacks
    - White-box attacks: FGSM, BIM, C&W and PGD
    - Properties to exploit: linearity by computing input gradients
  - How can a real-world attacker exploit them in practice?
    - Black-box attacks:
      - Transfer attacks
      - Query-based attacks
    - Properties to exploit:
      - Transfer attacks: surrogate models (often ensembled)
      - Query-based attacks: data-dependent and time-dependent priors
  - How can we remove adversarial examples?

Topics for today

- How can we remove adversarial examples?
  - Systems approach
    - Training-time defense: “Adversarial Training”
    - Post-training defense: “Feature Squeezing”
  - Certified approach (next lecture)

Motivation

- Initial adversarial example research
  - FGSM[1]…

How Can We Train Models Robust to Adversarial Examples?

Goodfellow *et al.*, Explaining and Harnessing Adversarial Examples, ICLR 2015

The key idea

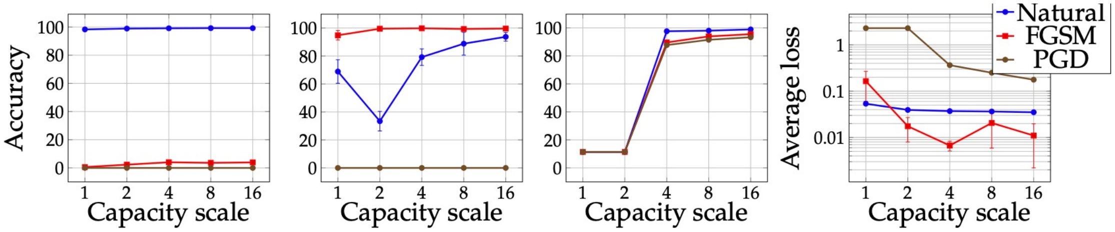
- Adversarial training
  - Deep neural networks (DNNs) are universal function approximators[1]
  - DNNs may learn to be resistant to adversarial examples (a desirable function)
  - Adversarial training (AT):

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)))$$

Hornik *et al.*, Multilayer feedforward networks are universal approximators, Neural Networks 1989

The key idea – cont'd

- Adversarial training
  - Deep neural networks (DNNs) are universal function approximators[1]
  - DNNs may learn to be resistant to adversarial examples (a desirable function)
  - Adversarial training (AT):
    - In MNIST, AT reduces an error rate from 89.4% to 17.9% on FGSM
    - AT with FGSM don't increase the robustness to strong attacks[2]



CIFAR10

	(a) Standard training Simple	(a) Standard training Wide	(b) FGSM training Simple	(b) FGSM training Wide	(c) PGD training Simple	(c) PGD training Wide	(d) Training Loss Simple	(d) Training Loss Wide
Natural	92.7%	95.2%	87.4%	90.3%	79.4%	87.3%	0.00357	0.00371
FGSM	27.5%	32.7%	90.9%	95.1%	51.7%	56.1%	0.0115	0.00557
PGD	0.8%	3.5%	0.0%	0.0%	43.7%	45.8%	1.11	0.0218

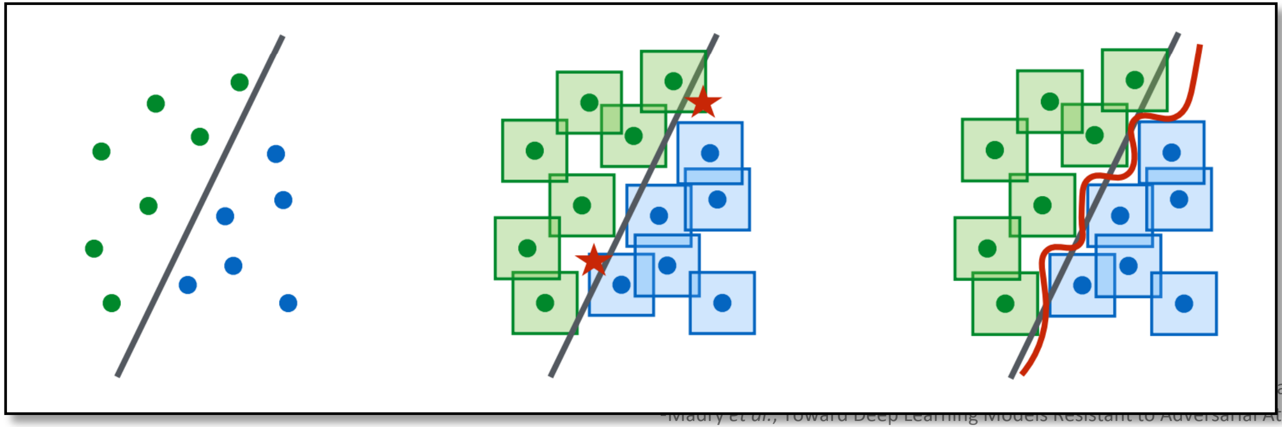
(a) Standard training (b) FGSM training (c) PGD training (d) Training Loss

ersal approximators, Neural Networks 1989

[2]Madry *et al.*, Toward Deep Learning Models Resistant to Adversarial Attacks, ICLR 2018

THE KEY IDEA – CONT'D

- Adversarial training
  - Deep neural networks (DNNs) are universal function approximators[1]
  - DNNs may learn to be resistant to adversarial examples (a desirable function)
  - Adversarial training (AT):
    - In MNIST, AT reduces an error rate from 89.4% to 17.9% on FGSM
    - AT with FGSM don't increase the robustness to strong attacks[2]
    - AT with strong attacks (e.g., PGD) require a large capacity model



ators, Neural Networks 1989
[2] Madry et al., Toward Deep Learning Models Resistant to Adversarial Attacks, ICLR 2018

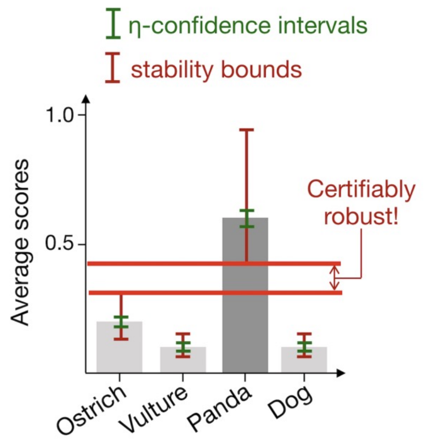
Adversarial training

- Sub-research questions:
  - SRQ 1: What does it mean by your model is robust?
  - SRQ 2: What is the upper-bound of the robustness?
  - SRQ 2: How can you certify that your model is robust?
  - SRQ 3: How can we make the certification computationally feasible?

SRQ 1: What does it mean by your model is robust?

- Suppose:
  - (x, y): a test-time input and its oracle label
  - $x + \delta$: an adversarial example of x with small l_p–bounded (ε) perturbation δ
  - f: a neural network

- Robustness
  - For any δ where $||\delta||_p \leq \varepsilon$
  - The most probable class y_M for $f(x + \delta)$
  - Make f to be $\mathrm{P}[f(x + \delta) = y_M] > \max_{y \neq y_M} \mathrm{P}[f(x + \delta) = y]$



(b) Robustness Test Example

SRQ 1: What does it mean by your model is robust?

- Smoothing:
  - In image processing: reduce noise (high frequency components)
  - In neural networks: make f less sensitive to noise

- Randomized:
  - In statistics: the practice of using chance methods (random)
  - In this work: add Gaussian random noise $\sim N(0, \sigma^2 I)$ to the input x

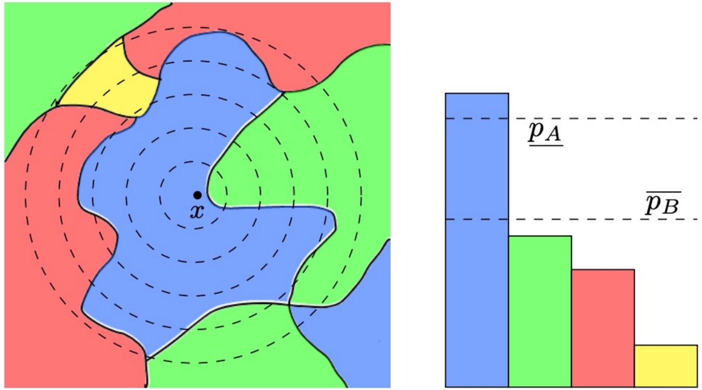
- Randomized Smoothing[1]:
  - [Train w. Gaussian noise to f's input]
    [to make it less sensitive to adversarial perturbations]

$$g(x) = \underset{c \in \mathcal{Y}}{\arg\max}\ \mathbb{P}(f(x + \varepsilon) = c)$$
$$\text{where } \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

[1] Cohen *et al.*, Certified Adversarial Robustness via Randomized Smoothing, ICML 2019

SRQ 2: What is the upper-bound of the robustness?

- Suppose
  - f: a base classifier (*e.g.*, a NN)
  - $\mathrm{P}[f(x+\delta) = c_A] \approx P_A$
  - $\max_{y \neq y_M} \mathrm{P}[f(x+\delta) = y] \approx P_B$



- Certified robustness
  - The smoothed classifier g is robust around x with the l_2 radius

$$\boxed{R = \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})\right)}$$

- Observations
  - f can be any classifier, *e.g.*, convolutional neural networks, …
  - R (Guarantee) is large when we use high noise, c_A is high, or c_B is low
  - R (Guarantee) is infinite as $P_A \approx 1$ and $P_B \approx 0$

SRQ 3: How you can certify a model is robust?

- Certification and classification with the robustness

Pseudocode for certification and prediction

 # evaluate g at x
function $\textsc{Predict}(f, \sigma, x, n, \alpha)$
  $\texttt{counts} \leftarrow \textsc{SampleUnderNoise}(f, x, n, \sigma)$
  $\hat{c}_A, \hat{c}_B \leftarrow$ top two indices in $\texttt{counts}$
  $n_A, n_B \leftarrow \texttt{counts}[\hat{c}_A], \texttt{counts}[\hat{c}_B]$
  if $\textsc{BinomPValue}(n_A, n_A + n_B, 0.5) \leq \alpha$ **return** $\hat{c}_A$
  else return ABSTAIN

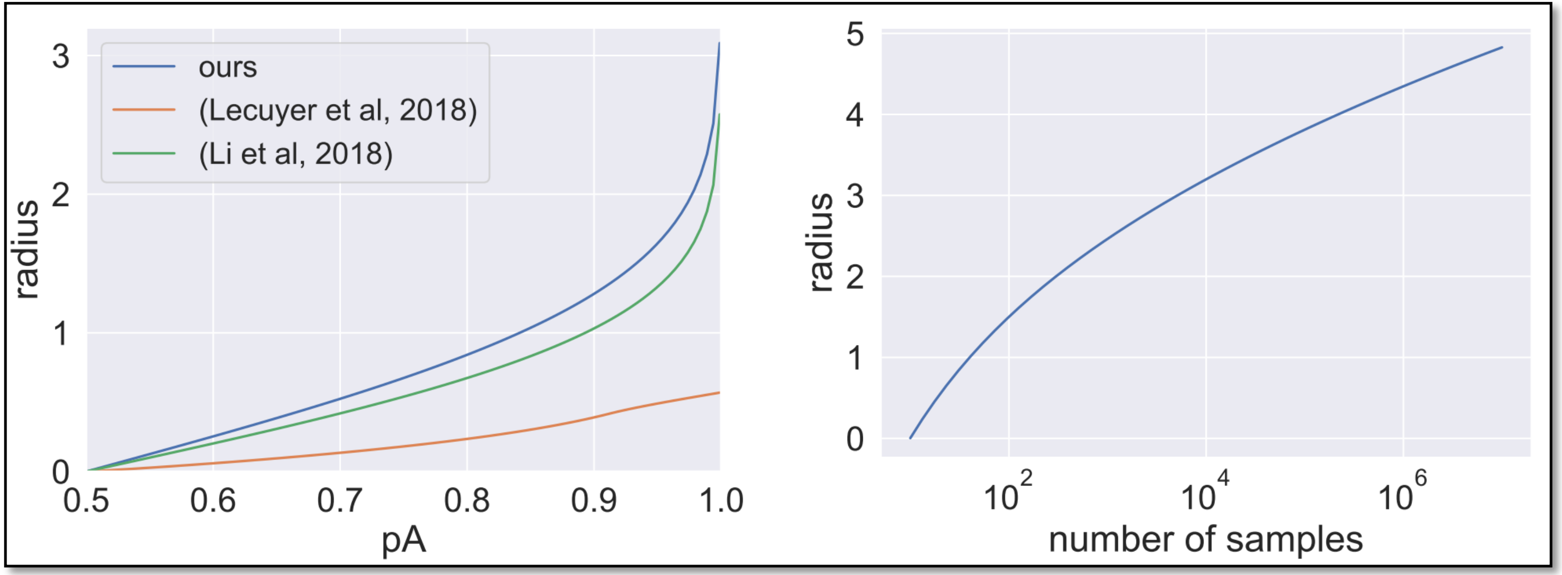
 # certify the robustness of g around x
function $\textsc{Certify}(f, \sigma, x, n_0, n, \alpha)$
  $\texttt{counts0} \leftarrow \textsc{SampleUnderNoise}(f, x, n_0, \sigma)$
  $\hat{c}_A \leftarrow$ top index in $\texttt{counts0}$
  $\texttt{counts} \leftarrow \textsc{SampleUnderNoise}(f, x, n, \sigma)$
  $\underline{p_A} \leftarrow \textsc{LowerConfBound}(\texttt{counts}[\hat{c}_A], n, 1 - \alpha)$
  if $\underline{p_A} > \frac{1}{2}$ **return** prediction $\hat{c}_A$ and radius $\sigma\, \Phi^{-1}(\underline{p_A})$
  else return ABSTAIN

Guarantee the probability of $PREDICT$ returning a class other than $g(x)$ is α

$CERTIFY$ returns a class c_A and a radius R for the $g(x)$ with the probability α

SRQ 3: How you can certify a model is robust?

- Certification and classification with the robustness



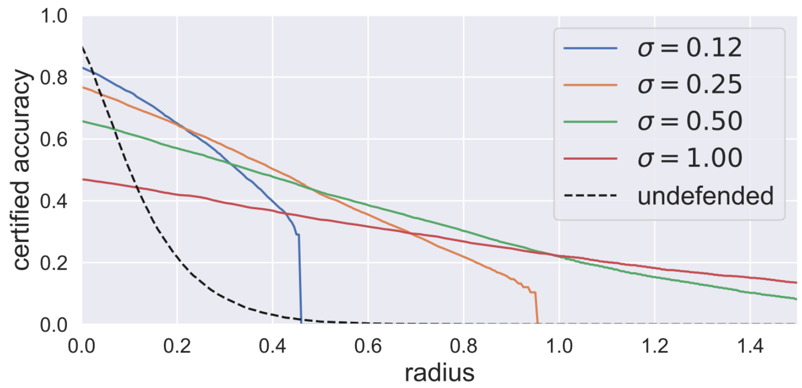
SRQ 3: How you can certify a model is robust?

- Setup
  - CIFAR10: ResNet-110 and its full test-set
  - ImageNet: ResNet-50 and 500 random chosen test-set samples

- Measure
  - (approximate) Certified test-set accuracy

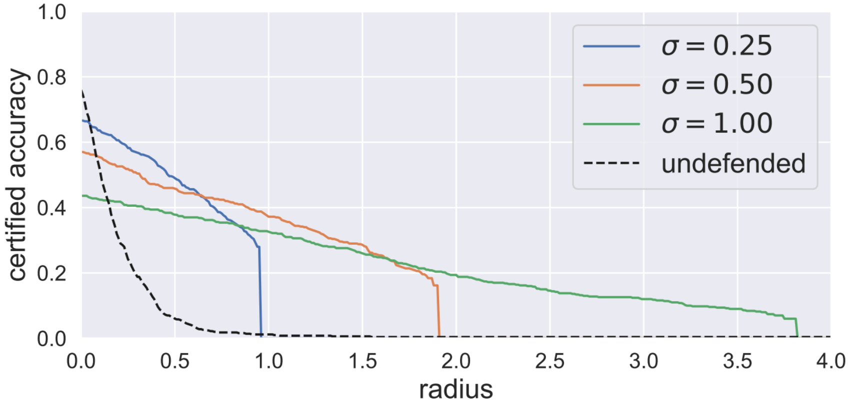
SRQ 3: How you can certify a model is robust?

- Radius R vs. certified accuracy (by smoothing with σ)



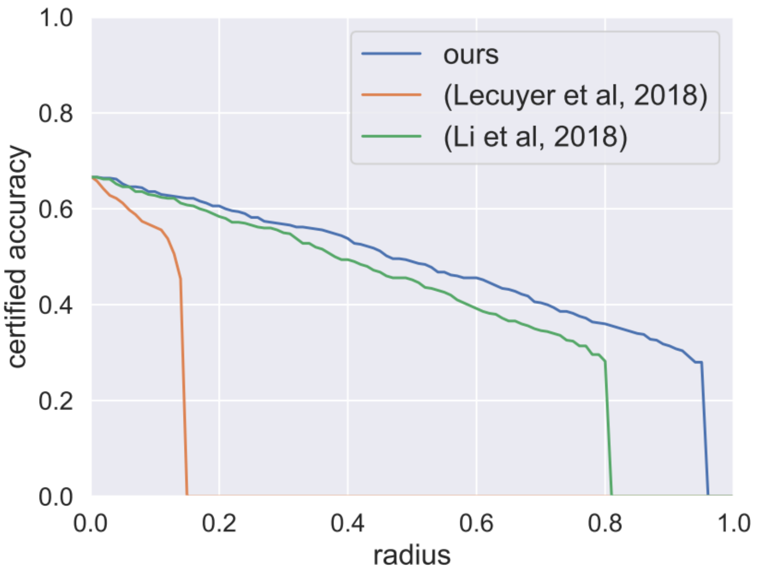
← CIFAR10

ImageNet →



SRQ 3: How you can certify a model is robust?

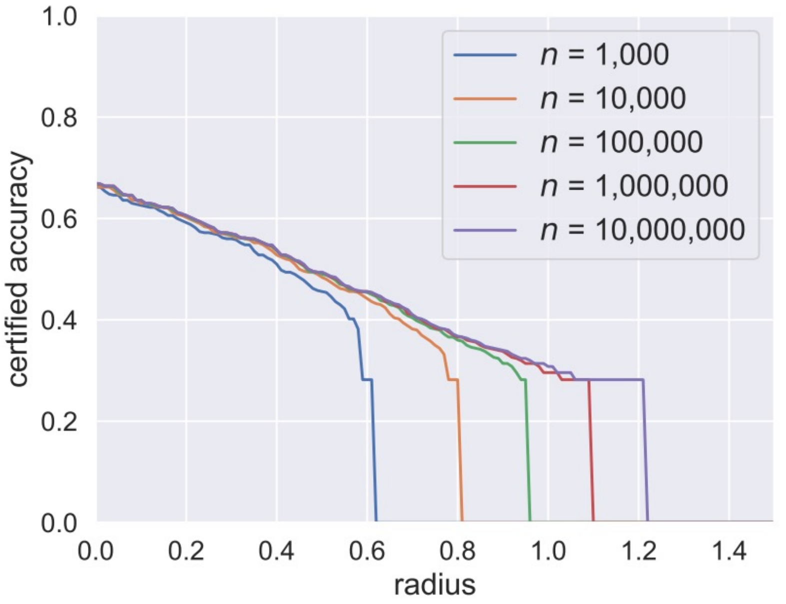
- Certified accuracy compared to prior work

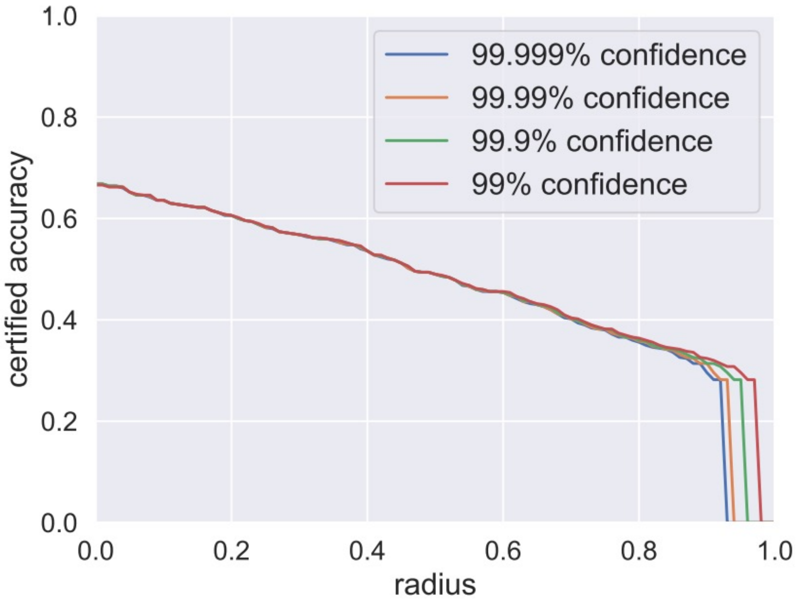


← ImageNet, smoothed by $\sigma = 0.25$

SRQ 3: How you can certify a model is robust?

- Certified accuracy vs. { # samples or confidence }





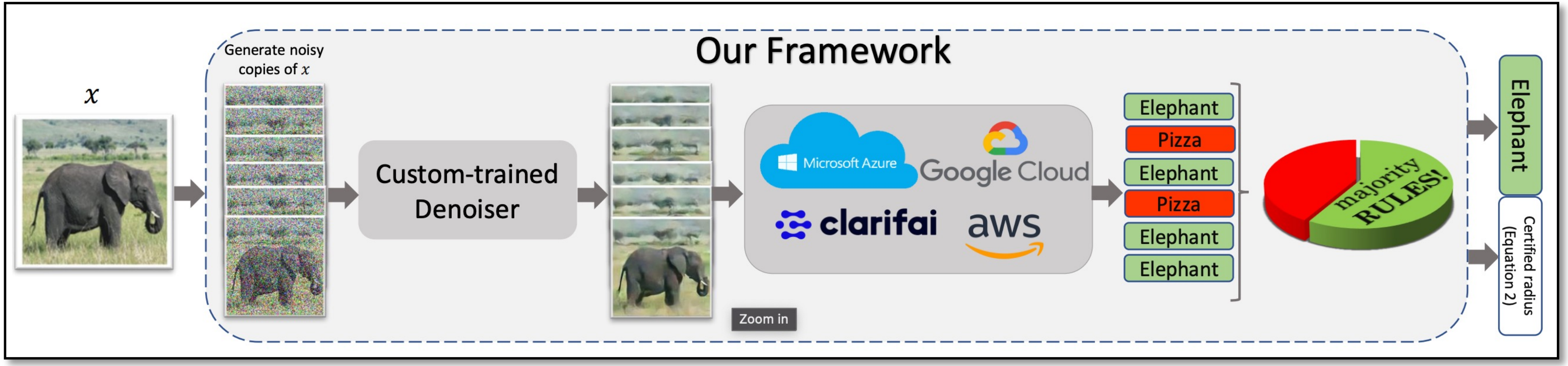
SRQ 4: How can we make this computationally feasible?

- Conversion to a robust classifier
  - Train a base classifier f with noised samples $\sim N(x, \sigma^2 I)$ with x's oracle label
  - Train a denoiser $D_\theta : R^d \rightarrow R^d$ that removes the input perturbations for f
- Problem:
  - Should we re-train all the classifiers, already trained and on-service?
  - How much would it be practical? [Consider ImageNet models]
- Solution:
  - Denoised smoothing[1]: add a denoiser on top of a pre-trained classifier

[1] Salman *et al.*, Denoised Smoothing: A Provable Defense for Pretrained Classifiers, NeurIPS 2020

SRQ 4: How can we make this computationally feasible?

- Conversion to a robust classifier
  - Train a base classifier f with noised samples $\sim N(x, \sigma^2 I)$ with x's oracle label
  - Train a denoiser $D_\theta : R^d \rightarrow R^d$ that removes the input perturbations for f



SRQ 4: How can we make this computationally feasible?

- Goal
  - Not to train f on noise
  - But, to provide certification to f

- Formally, We want
  - This: $g(x) = \underset{c \in \mathcal{Y}}{\arg\max}\ \mathbb{P}[f(x+\delta) = c] \quad \text{where } \delta \sim \mathcal{N}(0, \sigma^2 I)$
  - To be this: $g(x) = \underset{c \in \mathcal{Y}}{\arg\max}\ \mathbb{P}[f(\mathcal{D}_\theta(x+\delta)) = c] \quad \text{where } \delta \sim \mathcal{N}(0, \sigma^2 I)$

- Train D_θ
  - **MSE** objective: Just train D_θ to remove Gaussian noise $L_{\text{MSE}} = \underset{\mathcal{S},\delta}{\mathbb{E}} \|\mathcal{D}_\theta(x_i+\delta) - x_i\|_2^2$
  - **+ Stability** objective: (White-box) Preserve f's predictions $L_{\text{Stab}} = \underset{\mathcal{S},\delta}{\mathbb{E}}\ \ell_{\text{CE}}(F(\mathcal{D}_\theta(x_i+\delta)), f(x_i))$

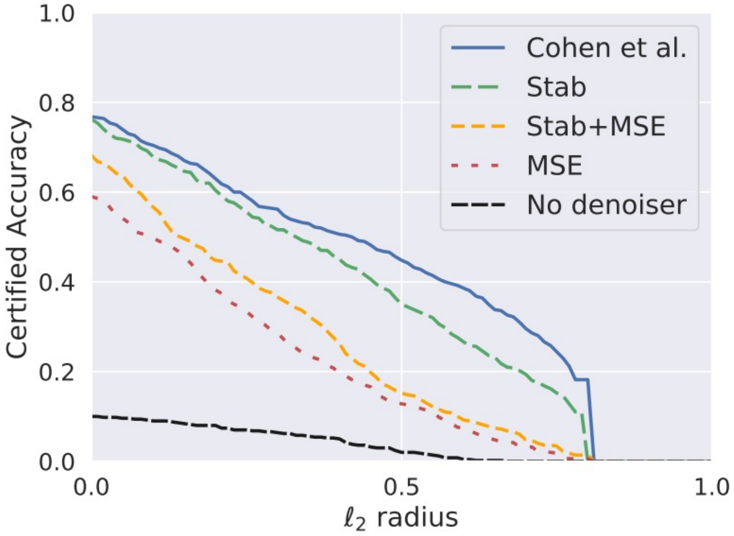
SRQ 4: How can we make this computationally feasible?

- Setup
  - ImageNet:
    - Pre-trained classifiers: ResNet-18/34/50 (white-box)
    - Baseline: ResNet-110 certified with $\sigma = 1.0$
  - Denoisers: DnCNN and MemNet trained with $\sigma = 0.25, 0.5, 1.0$
  - Objectives: MSE / Stab / Stab+MSE
    - White-box (as-is) | Black-box (14-surrogate models)

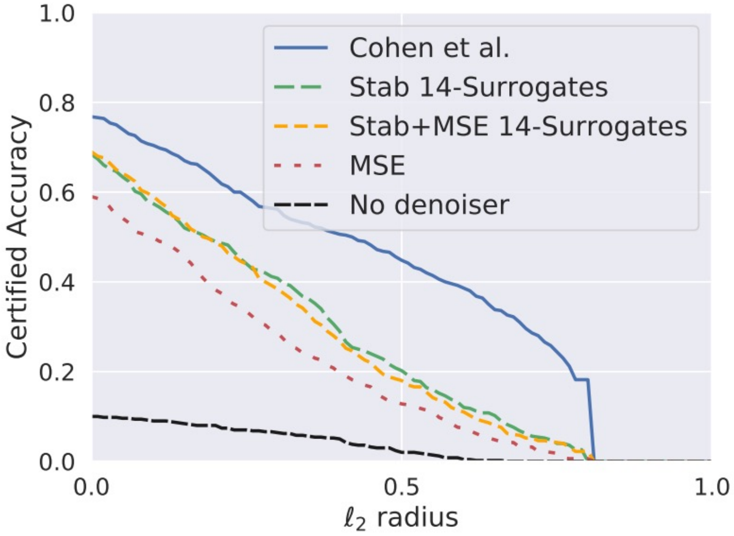
- Measure
  - (approximate) Certified test-set accuracy

SRQ 4: How can we make this computationally feasible?

- Radius R vs. certified accuracy (train denoisers with $\sigma = 0.25$)



(a) White-box



(b) Black-box

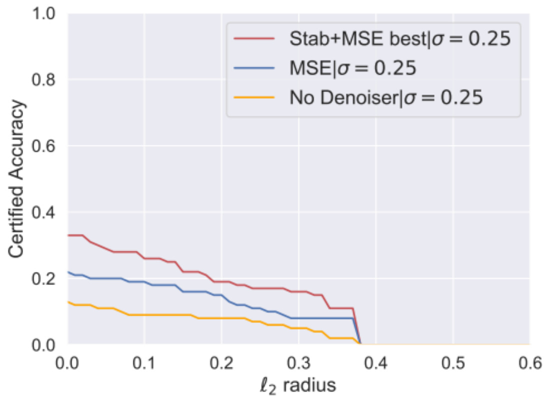
SRQ 4: How can we make this computationally feasible?

- Radius R vs. certified accuracy (train denoisers with $\sigma = 0.25$)

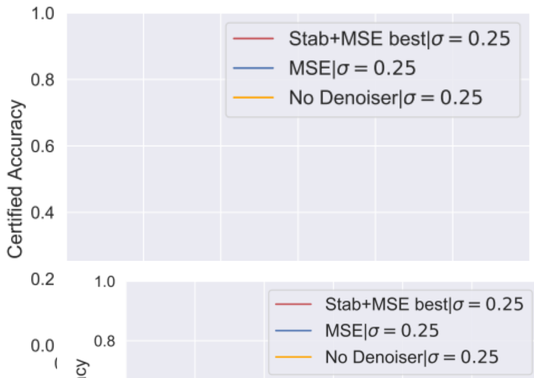
(a) Noisy (b) MSE (c) Stab+MSE

SRQ 4: How can we make this computationally feasible?

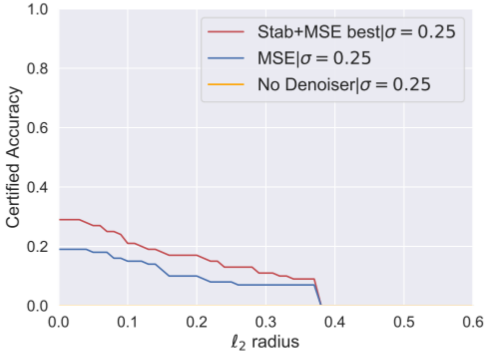
- Radius R vs. certified accuracy (train denoisers with $\sigma = 0.25$)



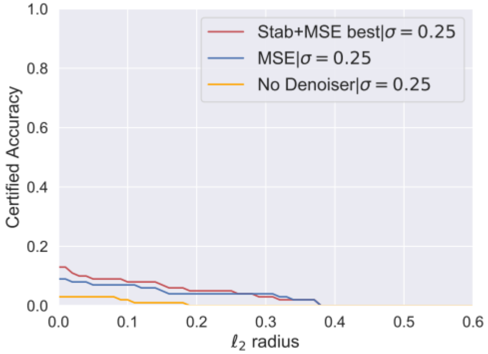
(a) Azure



(b



(c) Clarifai



(d) AWS

Topics for today

- How can we remove adversarial examples?
  - Systems approach
    - Training-time defense: “Adversarial Training”
    - Post-training defense: “Feature Squeezing”
  - Certified approach (next lecture)

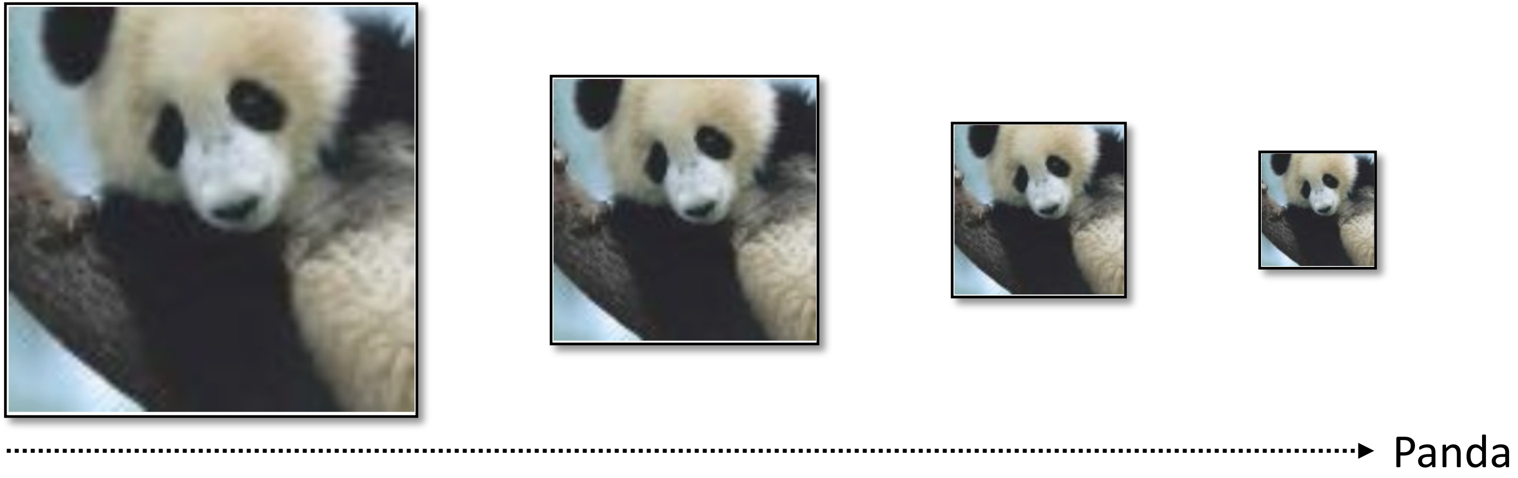
Motivation

- Existing Defenses
  - **Make** robust models:
    - (Gradient masking) Defensive distillation
    - Adversarial training
    - …
  - **Detect** adversarial examples:
    - Sample statistics
    - Train a detector model
    - Prediction inconsistency (majority vote…)
    - …

Can We Make Adversarial Perturbation Ineffective?

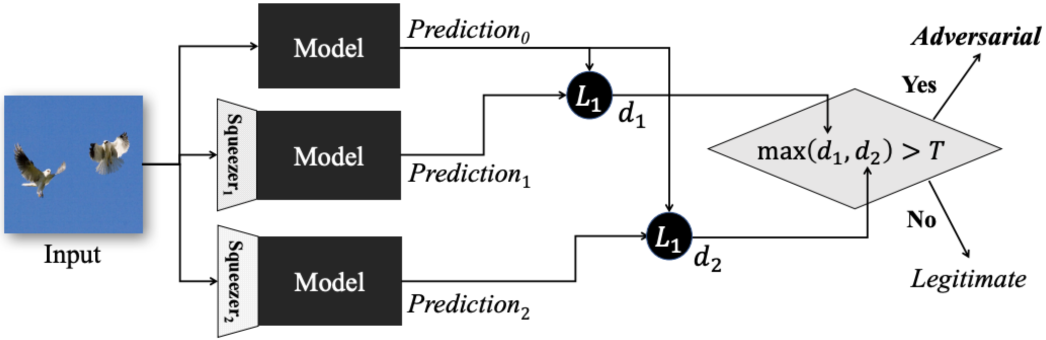
Motivation – cont'd

- Information-theoretical Perspective
  - Compression!



The key idea: feature squeezing

- FeatureSqueezing



  - (Goal) To detect whether an input is adversarial example or not
  - (Idea) A model should return similar predictions over squeezed samples

Feature squeezing

- Sub-research questions:
  - SRQ 1: What are the **squeezers** a defender can choose?
  - SRQ 2: How **effective** are they in defeating adversarial attacks?
  - SRQ 3: How **effective** are they when **combined with existing defenses**?
  - SRQ 4: How **effective** is feature-squeezing against **adaptive attacks**?

SRQ 1: What are the squeezers a defender can choose?

- H-space
  - Reduce the color depth (8-bit: 0-255 to lower-bit widths)
  - Reduce the variation among pixels
    - Local smoothing (*e.g.*, median filter)
    - Non-local smoothing (*e.g.*, denoiser filters)
  - More
    - JPEG compression [Kurakin *et al.*]
    - Dimensionality reduction [Turk and Pentland]

Local smoothing

SRQ 2: How effective are they in defeating adv. attacks?

- Empirical approach (Baseline)
  - Setup
    - MNIST, CIFAR10, ImageNet
    - 7-layer CNN, DenseNet, and MobileNet
    - 100 images correctly classified by them
  - Attacks
    - FGSM, BIM, C&W, JSMA
    - L0, L2, and L-inf distances

Dataset	Configration	Attack	Mode	Cost (s)	Success Rate	Prediction Confidence	Distortion L_∞	Distortion L_2	Distortion L_0
MNIST	L_∞	FGSM		0.002	46%	93.89%	0.302	5.905	0.560
		BIM		0.01	91%	99.62%	0.302	4.758	0.513
		CW_∞	Next	51.2	100%	99.99%	0.251	4.091	0.491
			LL	50.0	100%	99.98%	0.278	4.620	0.506
	L_2	CW_2	Next	0.3	99%	99.23%	0.656	2.866	0.440
			LL	0.4	100%	99.99%	0.734	3.218	0.436
	L_0	CW_0	Next	68.8	100%	99.99%	0.996	4.538	0.047
			LL	74.5	100%	99.99%	0.996	5.106	0.060
		JSMA	Next	0.8	71%	74.52%	1.000	4.328	0.047
			LL	1.0	48%	74.80%	1.000	4.565	0.053
CIFAR-10	L_∞	FGSM		0.02	85%	84.85%	0.016	0.863	0.997
		BIM		0.2	92%	95.29%	0.008	0.368	0.993
		CW_∞	Next	225	100%	98.22%	0.012	0.446	0.990
			LL	225	100%	97.79%	0.014	0.527	0.995
	L_2	DeepFool		0.4	98%	73.45%	0.028	0.235	0.995
		CW_2	Next	10.4	100%	97.90%	0.034	0.288	0.768
			LL	12.0	100%	97.35%	0.042	0.358	0.855
	L_0	CW_0	Next	367	100%	98.19%	0.650	2.103	0.019
			LL	426	100%	97.60%	0.712	2.530	0.024
		JSMA	Next	8.4	100%	43.29%	0.896	4.954	0.079
			LL	13.6	98%	39.75%	0.904	5.488	0.098
ImageNet	L_∞	FGSM		0.02	99%	63.99%	0.008	3.009	0.994
		BIM		0.2	100%	99.71%	0.004	1.406	0.984
		CW_∞	Next	211	99%	90.33%	0.006	1.312	0.850
			LL	269	99%	81.42%	0.010	1.909	0.952
	L_2	DeepFool		60.2	89%	79.59%	0.027	0.726	0.984
		CW_2	Next	20.6	90%	76.25%	0.019	0.666	0.323
			LL	29.1	97%	76.03%	0.031	1.027	0.543
	L_0	CW_0	Next	608	100%	91.78%	0.898	6.825	0.003
			LL	979	100%	80.67%	0.920	9.082	0.005

SRQ 2: How effective are they in defeating adv. attacks?

- Empirical approach (Feature Squeezing)

Dataset	Squeezer		L_∞ Attacks				L_2 Attacks			L_0 Attacks				All Attacks	Legitimate
	Name	Parameters	FGSM	BIM	CW_∞		Deep-Fool	CW_2		CW_0		JSMA			
					Next	LL		Next	LL	Next	LL	Next	LL		
MNIST	None		54%	9%	0%	0%	-	0%	0%	0%	0%	27%	40%	13.00%	99.43%
	Bit Depth	1-bit	**92%**	**87%**	**100%**	**100%**	-	**83%**	**66%**	0%	0%	50%	49%	**62.70%**	99.33%
	Median Smoothing	2x2	61%	16%	70%	55%	-	51%	35%	39%	36%	62%	56%	48.10%	99.28%
		3x3	59%	14%	43%	46%	-	51%	53%	**67%**	**59%**	**82%**	**79%**	55.30%	98.95%
CIFAR-10	None		15%	8%	0%	0%	2%	0%	0%	0%	0%	0%	0%	2.27%	94.84%
	Bit Depth	5-bit	17%	13%	12%	19%	40%	40%	47%	0%	0%	21%	17%	20.55%	94.55%
		4-bit	21%	29%	69%	74%	72%	84%	84%	7%	10%	23%	20%	44.82%	93.11%
	Median Smoothing	2x2	**38%**	**56%**	**84%**	**86%**	**83%**	**87%**	83%	**88%**	**85%**	**84%**	**76%**	**77.27%**	89.29%
	Non-local Means	11-3-4	27%	46%	80%	84%	76%	84%	**88%**	11%	11%	44%	32%	53.00%	91.18%
ImageNet	None		1%	0%	0%	0%	11%	10%	3%	0%	0%	-	-	2.78%	69.70%
	Bit Depth	4-bit	5%	4%	66%	79%	44%	84%	82%	38%	67%	-	-	52.11%	68.00%
		5-bit	2%	0%	33%	60%	21%	68%	66%	7%	18%	-	-	30.56%	69.40%
	Median Smoothing	2x2	22%	28%	75%	81%	**72%**	81%	84%	**85%**	**85%**	-	-	**68.11%**	65.40%
		3x3	**33%**	**41%**	73%	76%	66%	77%	79%	81%	79%	-	-	67.22%	62.10%
	Non-local Means	11-3-4	10%	25%	**77%**	**82%**	57%	**87%**	**86%**	43%	47%	-	-	57.11%	65.40%

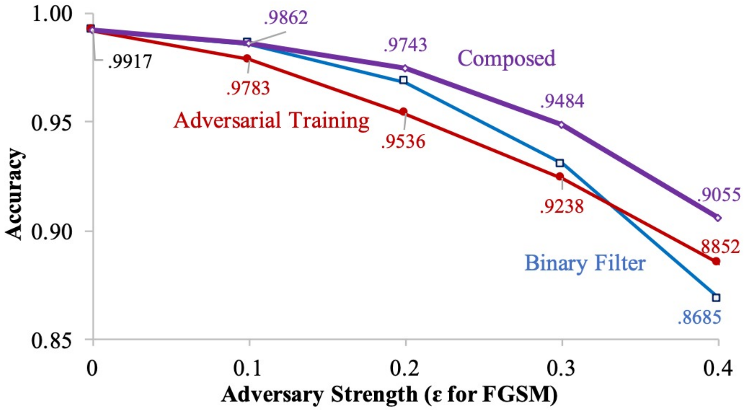
SRQ 2: How effective are they in defeating adv. attacks?

- Detection:
  - Metric:
    - Used with a single squeezer "score = $||f(x) - f(x^{squeezed})||_{l_1}$"
    - Used with multiple squeezer "score = $\max(score^{squeezer_1}, score^{squeezer_2}, \dots)$"

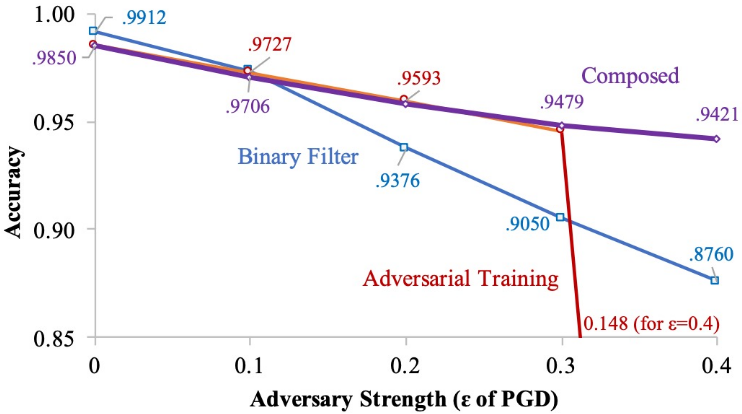
	Configuration			L_∞ Attacks				L_2 Attacks			L_0 Attacks				Overall Detection Rate
	Squeezer	Parameters	Threshold	FGSM	BIM	CW_∞ Next	CW_∞ LL	Deep Fool	CW_2 Next	CW_2 LL	CW_0 Next	CW_0 LL	JSMA Next	JSMA LL	
CIFAR-10	Bit Depth	1-bit	1.9997	0.063	0.075	0.000	0.000	0.019	0.000	0.000	0.000	0.000	0.000	0.000	0.013
		2-bit	1.9967	0.083	0.175	0.000	0.000	0.000	0.000	0.000	0.000	0.018	0.000	0.000	0.022
		3-bit	1.7822	0.125	0.250	0.755	0.977	0.170	0.787	0.939	0.365	0.214	0.000	0.000	0.409
		4-bit	0.7930	0.125	0.150	0.811	0.886	0.642	0.936	0.980	0.192	0.179	0.041	0.000	0.446
		5-bit	0.3301	0.000	0.050	0.377	0.636	0.509	0.809	0.878	0.096	0.018	0.041	0.038	0.309
	Median Smoothing	2x2	1.1296	0.188	**0.550**	**0.981**	**1.000**	0.717	0.979	**1.000**	**0.981**	**1.000**	**0.837**	**0.885**	0.836
		3x3	1.9431	0.042	0.250	0.660	0.932	0.038	0.681	0.918	0.750	0.929	0.041	0.077	0.486
	Non-local Mean	11-3-2	0.2770	0.125	0.400	0.830	0.955	0.717	0.915	0.939	0.077	0.054	0.265	0.154	0.484
		11-3-4	0.7537	0.167	0.525	0.868	0.977	0.679	0.936	**1.000**	0.250	0.232	0.245	0.269	0.551
		13-3-2	0.2910	0.125	0.375	0.849	0.977	0.717	0.915	0.939	0.077	0.054	0.286	0.173	0.490
		13-3-4	0.8290	0.167	0.525	0.887	0.977	0.642	0.936	**1.000**	0.269	0.232	0.224	0.250	0.547
	Best Attack-Specific Single Squeezer		-	0.188	**0.550**	**0.981**	**1.000**	0.717	0.979	**1.000**	**0.981**	**1.000**	**0.837**	**0.885**	-
	Best Joint Detection (5-bit, 2x2, 13-3-2)		1.1402	**0.208**	**0.550**	**0.981**	**1.000**	**0.774**	**1.000**	**1.000**	**0.981**	**1.000**	**0.837**	**0.885**	**0.845**

SRQ 3: How effective is this when combined with other defenses?

- FeatureSqueezing + AT
  - Setup
    - MNIST
    - AT (with epsilon 0.3) + Use 2-bit for Pixels
    - Use FGSM and PGD attacks (epsilon 0.1 – 0.4)



(a) FGSM attacks.



(b) PGD attacks.

SRQ 4: How effective is feature squeezing against adaptive att.?

- (Adaptive) attack
  - Attackers who know this feature squeezing is deployed
  - Adaptive attack (using C&W + L2 or L-inf):
    - Reduce the prediction difference between x and x^{adv} under a threshold
    - Set the threshold is the one used by the detector
  - Result on MNIST:

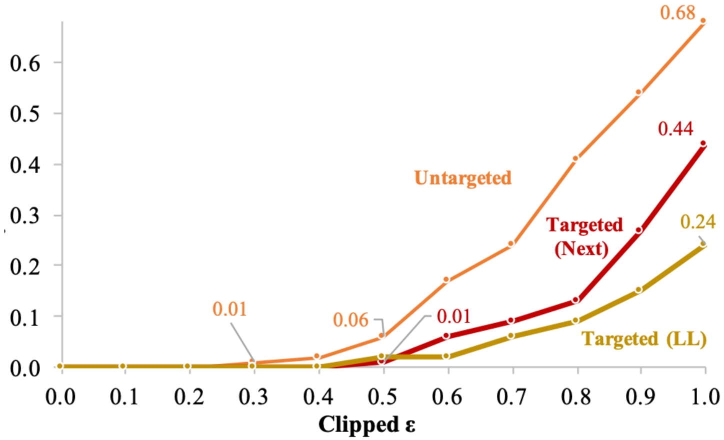


Fig. 7: Adaptive adversary success rates.

Motivation

- Research Questions
  - SRQ 1: What are the **squeezers** a defender can choose?
    - Bit-width reduction
    - Smoothing (local or non-local)
  - SRQ 2: How **effective** are they in defeating adversarial attacks?
    - Reduce the attack success rate by 87—100%
    - Detection rate is up to 100% when squeezers are jointly used
  - SRQ 3: How **effective** are they when **combined with existing defenses**?
    - On MNIST, it improves the robustness over what AT can provides
  - SRQ 4: How **effective** is feature-squeezing against **adaptive attacks**?
    - On MNIST, the attack success rate increases to 0-68%
    - One can choose a filter size randomly to defeat adaptive attacks (68% to 17%)

Thank You!

Tu/Th 10:00 – 11:50 am

Sanghyun Hong

https://secure-ai.systems/courses/MLSec/Sp23

Oregon State University

SAIL

Secure AI Systems Lab