# CS 499/599: Machine Learning Security
## 04.27: Defense II

Tu/Th 10:00 – 11:50 am

Sanghyun Hong
sanghyun.hong@oregonstate.edu

Oregon State University

SAIL
Secure AI Systems Lab

# Checkpoint Presentation II

Group 6 and Group 7

# Heads-up!

- Note
  - Great job for introducing intriguing mini-research ideas!
  - Do not forget to write reviews for others
  - SH will collect all the feedback from us and anonymously send to each group
- Due dates
  - 4/27: Homework 2
  - 5/02: Written paper critique (we will start looking at data poisoning!)
  - 5/04: SH's business travel; no lecture
- Recommendation
  - Discuss slides with SH for in-class paper presentation

# Recap

- Defenses
  - How can we remove adversarial examples?
    - Systems approach
      - Training-time defense: “adversarial-training”
      - Post-training defense: “feature squeezing”
    - Certified approach
      - (Revisit) Training-time defense: “adversarial-training”

# Towards Deep Learning Models Resistant to Adversarial Attacks

Amelia and Maha!

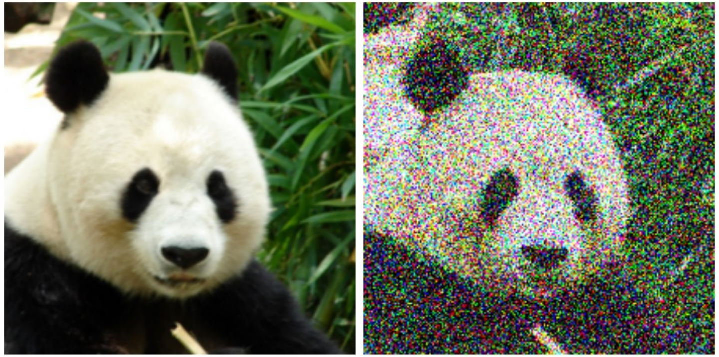
# Motivation

- Questions:
  - What does it mean by "**robust**" in ML?
  - How can we make ML models "**robust**"?

# Motivation

- Questions:
  - What does it mean by "**robust**" in ML?
  - How can we make ML models "**robust**"?

- Problems in the previous defenses
  - Are they "**really**" robust?
  - Are these solutions "**scalable**"?



# Motivation – cont'd

- Research Questions:
  - **RQ 1:** What is the "**upper-bound**" of the robustness?
  - **RQ 2:** How can you "**certify**" that yours is the upper-bound?
  - **RQ 3:** How can we make the certification "**computationally feasible**"?

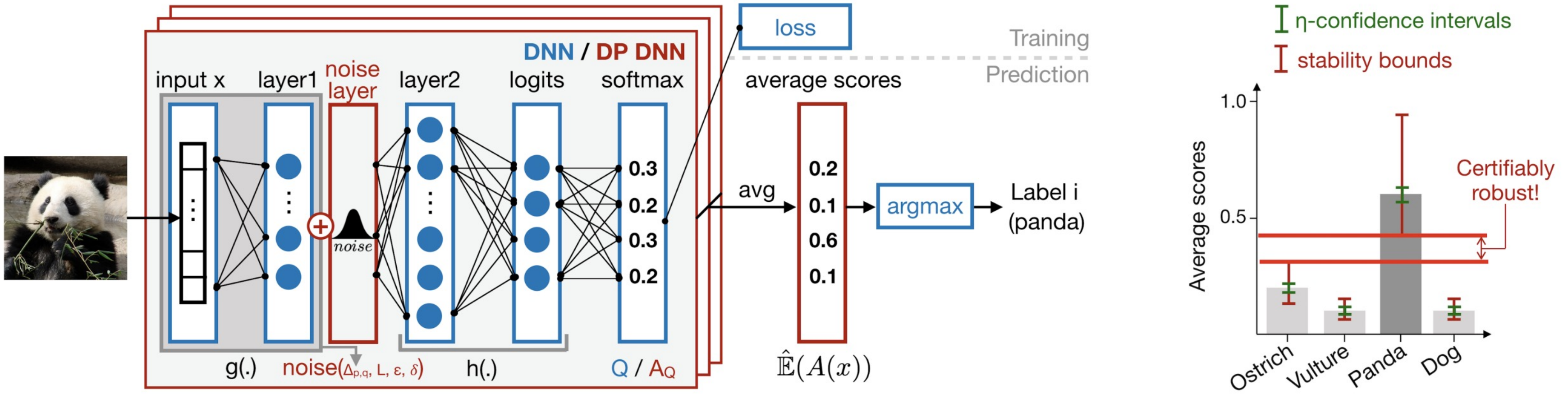
# Robustness

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

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

# What does it mean by "certified"?

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




(a) PixelDP DNN Architecture (b) Robustness Test Example

**Good, But What'd Be the Upper Bound?**

Lecuyer *et al.*, Certified Robustness to Adversarial Examples with Differential Privacy, IEEE S&P 2019

# Randomized smoothing

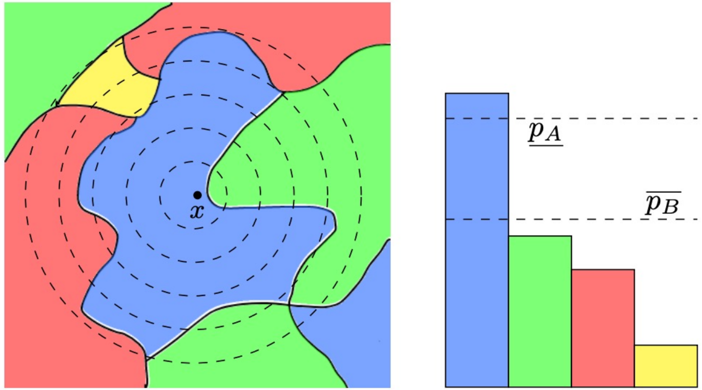
- **Smoothing**:
  - In image processing: reduce noise (high frequency components)
  - In neural networks: make $f$ less sensitive to noise

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

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

# Randomized smoothing: guarantee




- 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 $P_B \approx 0$

# Randomized smoothing: practicality

- Conversion to a robust classifier

**Pseudocode** for certification and prediction

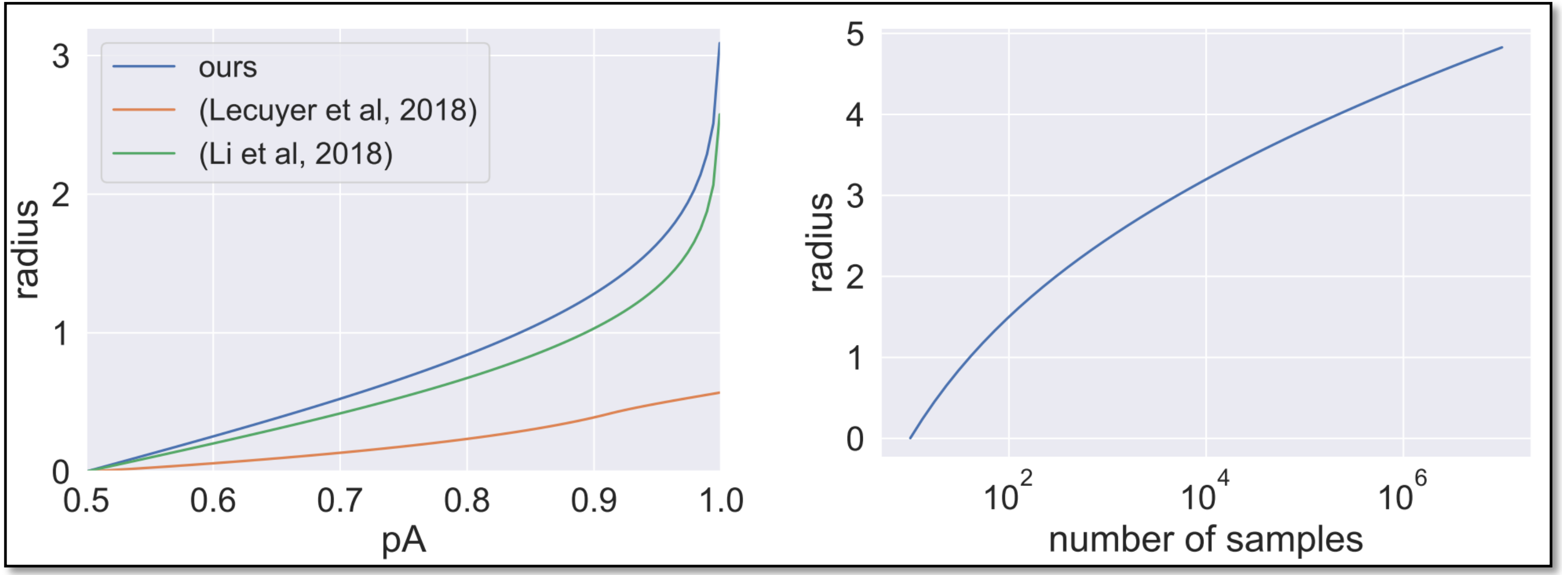
```
# evaluate g at x
function PREDICT(f, σ, x, n, α)
    counts ← SAMPLEUNDERNOISE(f, x, n, σ)
    ĉ_A, ĉ_B ← top two indices in counts
    n_A, n_B ← counts[ĉ_A], counts[ĉ_B]
    if BINOMPVALUE(n_A, n_A + n_B, 0.5) ≤ α return ĉ_A
    else return ABSTAIN

# certify the robustness of g around x
function CERTIFY(f, σ, x, n_0, n, α)
    counts0 ← SAMPLEUNDERNOISE(f, x, n_0, σ)
    ĉ_A ← top index in counts0
    counts ← SAMPLEUNDERNOISE(f, x, n, σ)
    p_A ← LOWERCONFBOUND(counts[ĉ_A], n, 1 − α)
    if p_A > 1/2 return prediction ĉ_A and radius σ Φ^{-1}(p_A)
    else return ABSTAIN
```

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

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

# Randomized smoothing: practicality

- Conversion to a robust classifier



# Randomized smoothing: implementations

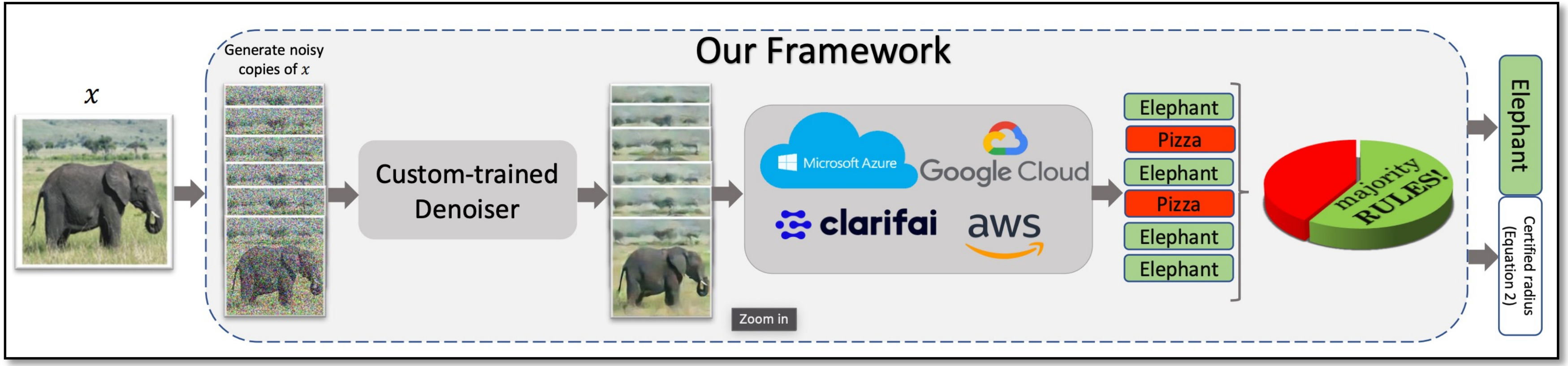
- Conversion to a robust classifier
  - Train a base classifier $f$ with noised samples $\sim N(x, \sigma^2 I)$ with $x$'s oracle label
  - Train a denoiser $D_\theta : R^d \to R^d$ that removes the input perturbations for $f$

- 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**: add a denoiser on top of a pre-trained classifier

# Randomized smoothing: implementations

- 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 $f$



# Denoised smoothing

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

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

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

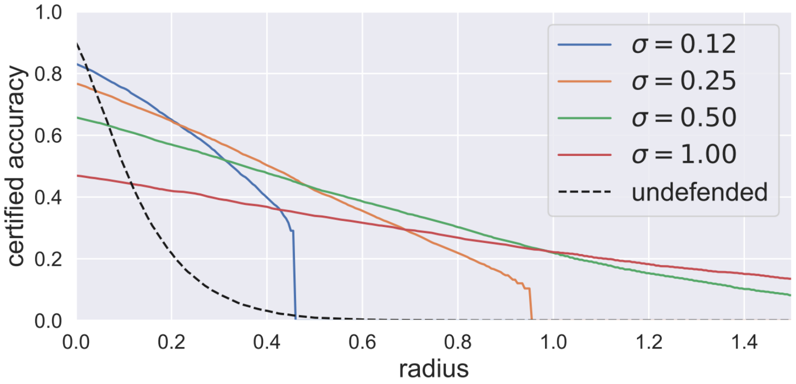
# Evaluation: randomized smoothing

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

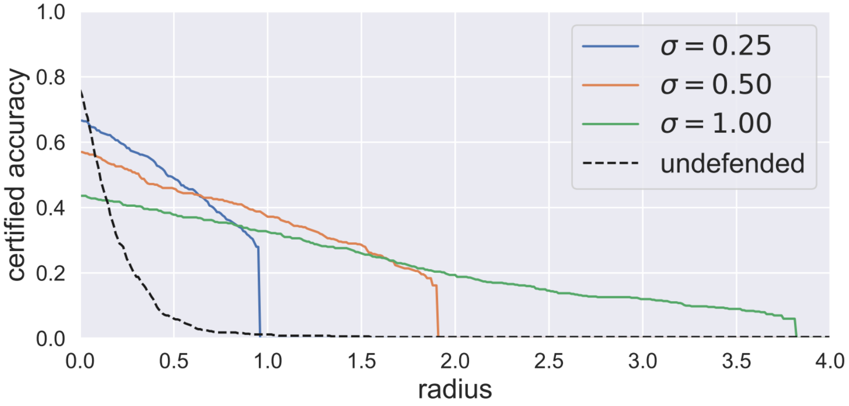
- Measure
  - (approximate) Certified test-set accuracy

# Evaluation: randomized smoothing

- Radius $R$ vs. certified accuracy (by smoothing with $\sigma$)



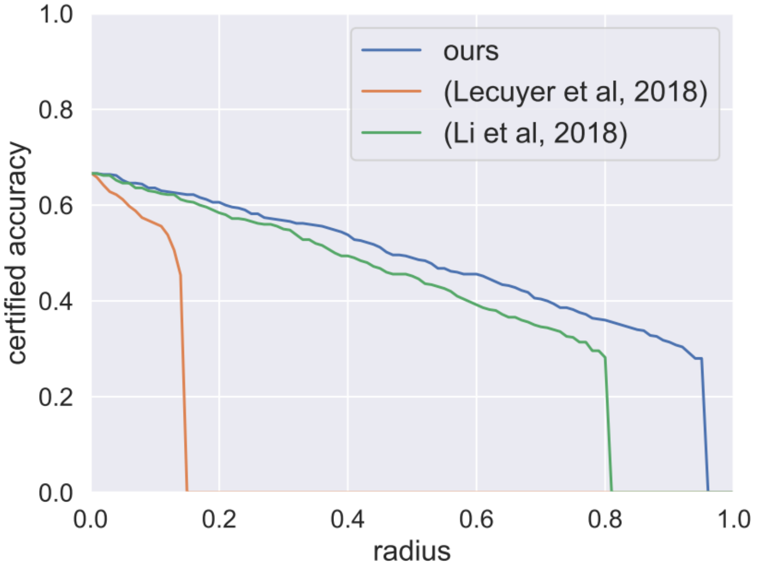

← CIFAR10




ImageNet →

# Evaluation: randomized smoothing

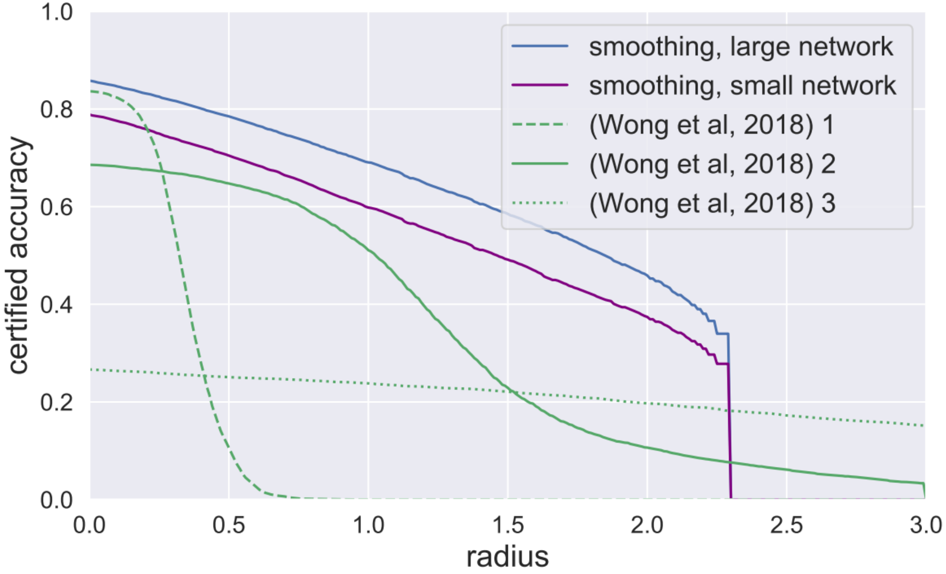
- Certified accuracy compared to prior work




← ImageNet, smoothed by $\sigma = 0.25$

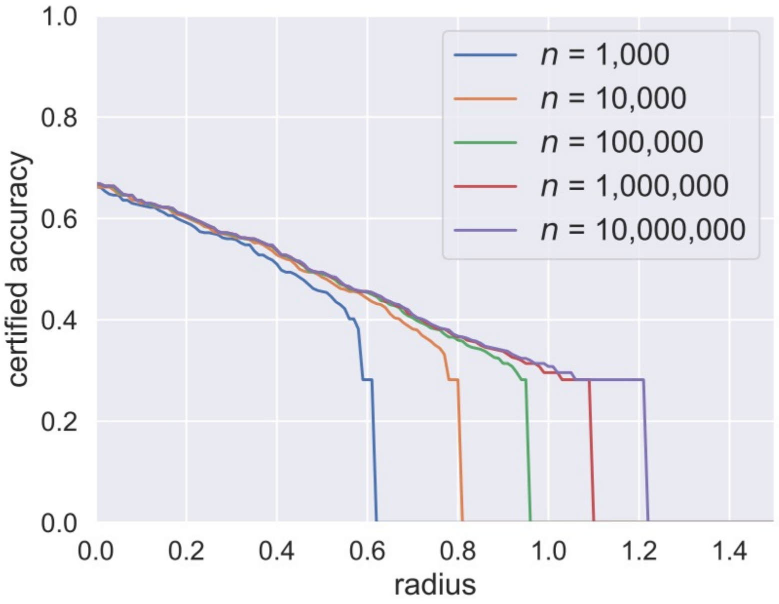
# Evaluation: randomized smoothing

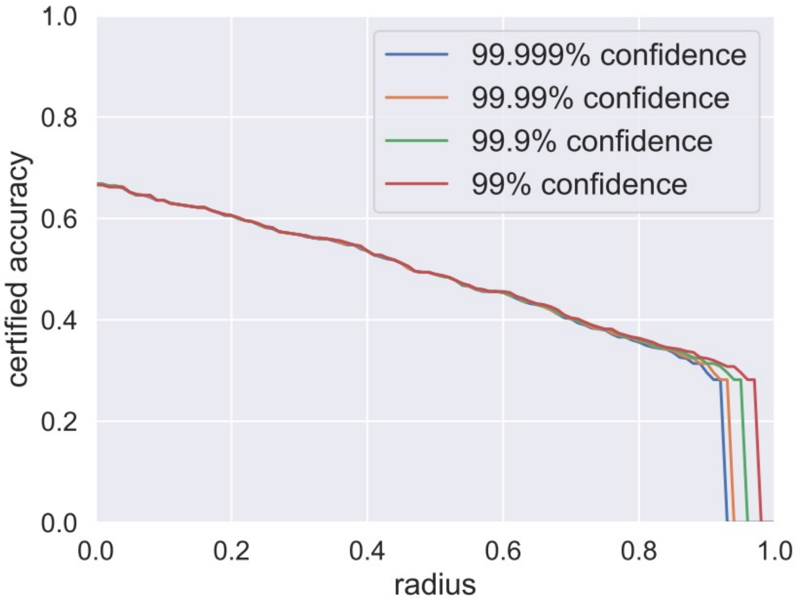
- Certified accuracy vs. other baselines



# Evaluation: randomized smoothing

- Certified accuracy vs. { # samples or confidence }



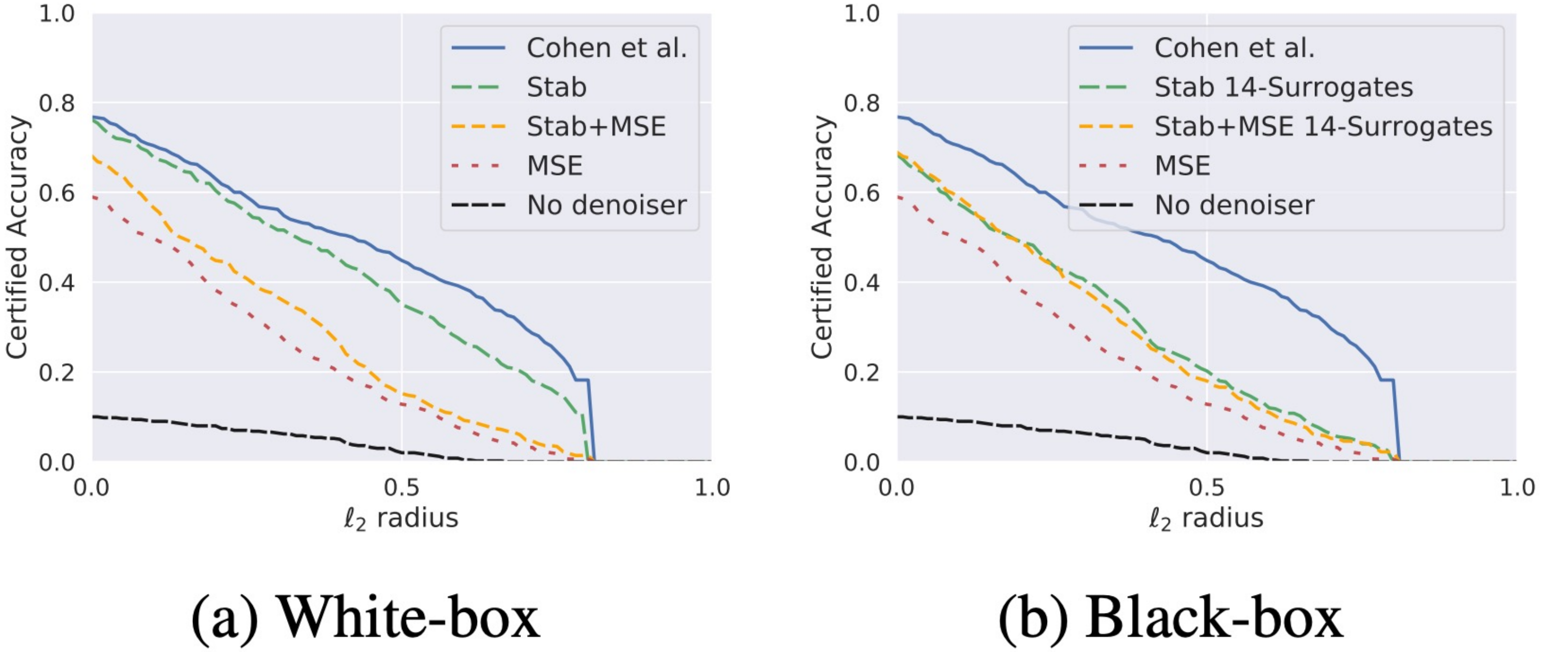
# Evaluation: denoised smoothing

- 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

# Evaluation: denoised smoothing

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



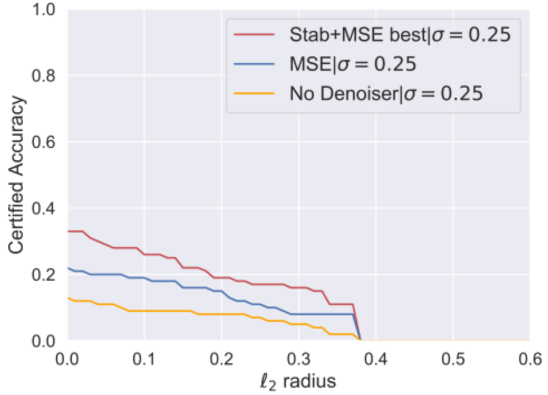
# Evaluation: denoised smoothing

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

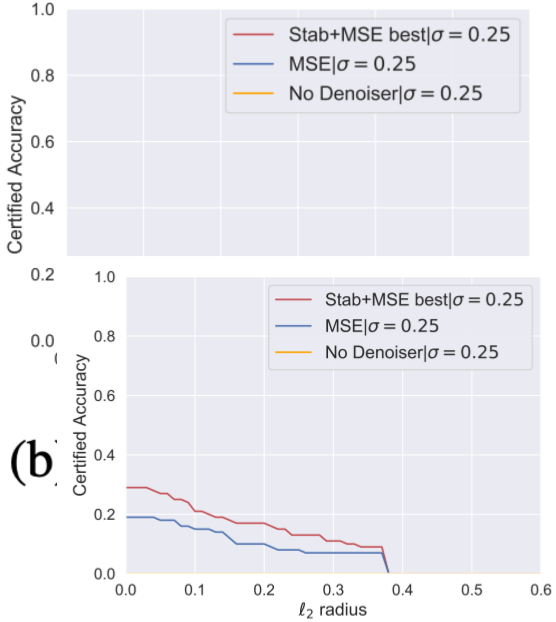

(a) Noisy (b) MSE (c) Stab+MSE

# Evaluation: denoised smoothing in the real-world

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

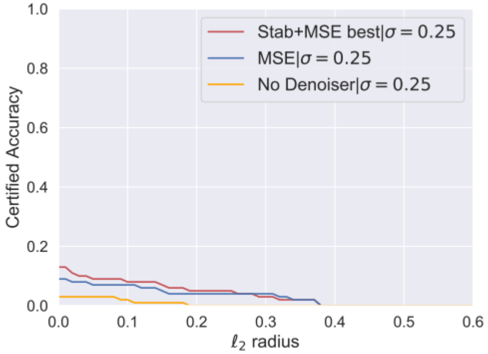



(a) Azure




(b

(c) Clarifai




(d) AWS

# Conclusion so far

- Research Questions:
  - **RQ 1:** What is the "**upper-bound**" of the robustness?
    - Certified accuracy offered by randomized smoothing
  - **RQ 2:** How can you "**certify**" that yours is the upper-bound?
    - Predict and Certify functions
  - **RQ 3:** How can we make the certification "**computationally feasible**"?
    - Train a base classifier with smoothing
    - Train a denoiser with a base classifier, and attach it to the input

# (Certified!!) Adversarial Robustness for Free!

Ethan Nechanicky!

# Thank You!

Tu/Th 10:00 – 11:50 am

Sanghyun Hong

https://secure-ai.systems/courses/MLSec/W22

Oregon State University

SAIL
Secure AI Systems Lab