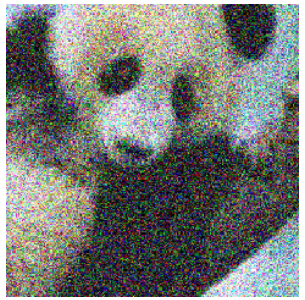
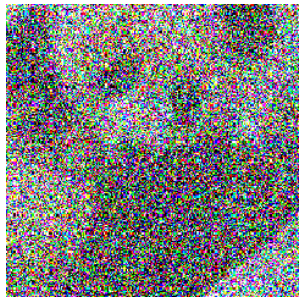
# Certified Adversarial Robustness via Randomized Smoothing



Jeremy Cohen Elan Rosenfeld Zico Kolter

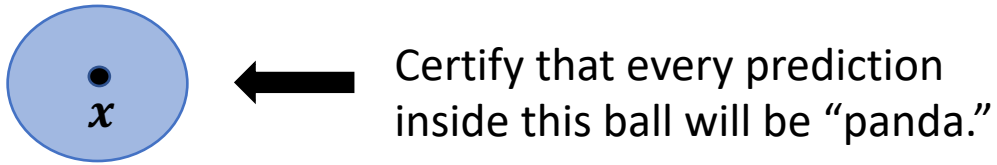
Carnegie Mellon University

# Introduction

We study a **<u>certified</u>** adversarial defense in $\ell_2$ norm which **<u>scales</u>** to ImageNet

Background:

- Many adversarial defenses have been “broken”
- A *certified* defense (in $\ell_2$ norm) is a classifier which returns *both* a prediction *and* a certificate that the prediction is constant within an $\ell_2$ around the input




- Most certified defenses don’t scale to networks of realistic size

# Prior work on randomized smoothing

- Randomized smoothing was proposed as a certified defense by [1]
- The analysis was improved upon by [2]
- Our main contribution is the *tight* analysis of this algorithm

[1] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. “Certified Robustness to Adversarial Examples with Differential Privacy,” IEEE S&P 2019.

[2] B. Li, C. Chen, W. Wang, and L. Carin. “Second-Order Adversarial Attack and Certifiable Robustness,” arXiv 2018.

# Randomized smoothing

- First, train a neural net $f$ (the "base classifier") with Gaussian data augmentation:

clean image

corrupted by Gaussian noise

- Then, *smooth* $f$ into a new classifier $g$ (the "smoothed classifier"), defined as follows:

# Randomized smoothing

$g(\boldsymbol{x})$ = the most probable prediction by $f$ of random Gaussian corruptions of $\boldsymbol{x}$

Example: consider the input $\boldsymbol{x}$ = 

Suppose that when $f$ classifies $\mathcal{N}(\boldsymbol{x}, \sigma^2 I)$  ,

- 🐼 is returned with probability 0.80
- 🐒 is returned with probability 0.15
- 🐱 is returned with probability 0.05

Then $g(\boldsymbol{x}) =$ 🐼

# Randomized smoothing

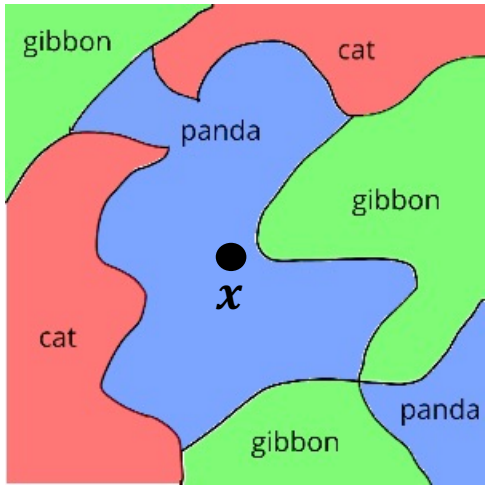
$g(\boldsymbol{x})$ = the most probable prediction by $f$ of random Gaussian corruptions of $\boldsymbol{x}$

Example: consider the input $\boldsymbol{x}$ = 

Suppose that when $f$ classifies $\mathcal{N}(\boldsymbol{x}, \sigma^2 I)$  ,

- 🐼 is returned with probability 0.80
- 🐒 is returned with probability 0.15
- 🐱 is returned with probability 0.05

Then $g(\boldsymbol{x})$ = 🐼



# Randomized smoothing

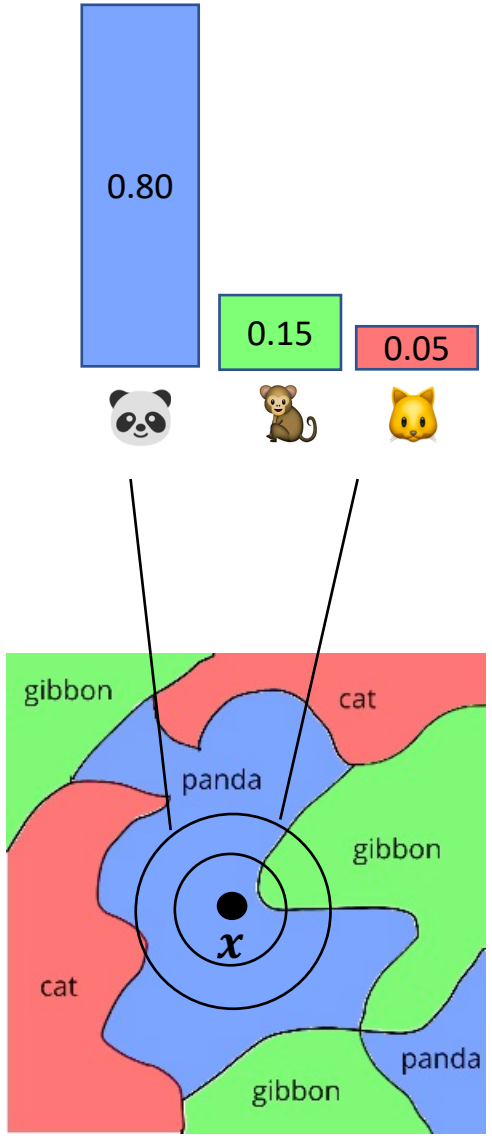
$g(\boldsymbol{x})$ = the most probable prediction by $f$ of random Gaussian corruptions of $\boldsymbol{x}$

Example: consider the input $\boldsymbol{x}$ = 

Suppose that when $f$ classifies $\mathcal{N}(\boldsymbol{x}, \sigma^2 I)$  ,

- 🐼 is returned with probability 0.80
- 🐒 is returned with probability 0.15
- 🐱 is returned with probability 0.05

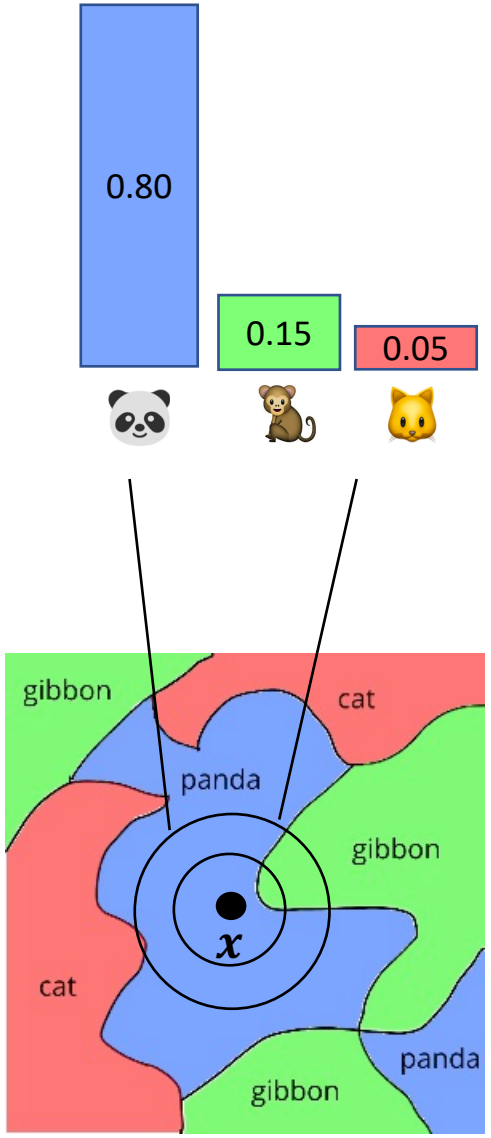
Then $g(\boldsymbol{x}) =$ 🐼



# Class probabilities vary slowly

If we shift this Gaussian, the probabilities of each class can't change by too much.

Therefore, if we know the class probabilities at the input $\boldsymbol{x}$, we can *certify* that for sufficiently small perturbations of $\boldsymbol{x}$, the 🐼 probability will remain higher than the 🐒 probability.

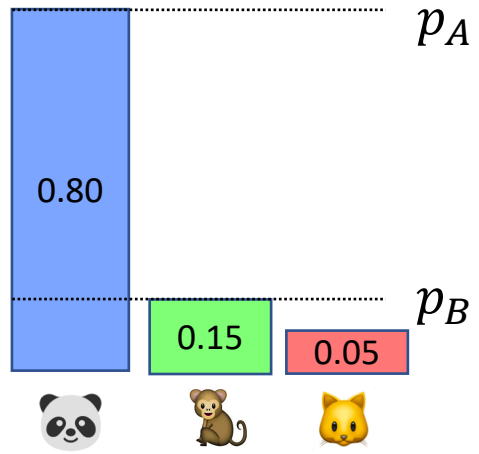


# Robustness guarantee (main result)

- Let $p_A$ be the probability of the top class (🐼)
- Let $p_B$ be the probability of the runner-up class (🐒).
- Then $g$ provably returns the top class 🐼 within an $\ell_2$ ball around $\boldsymbol{x}$ of radius

$$R = \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$$

where $\Phi^{-1}$ is the inverse standard Gaussian CDF.

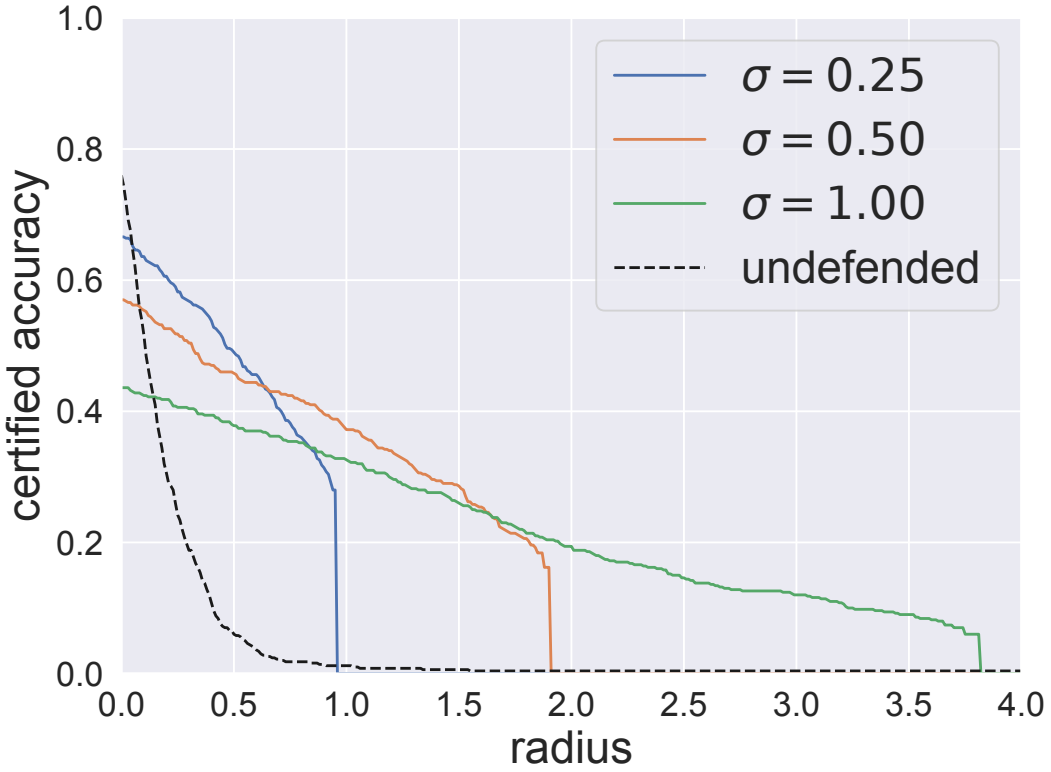


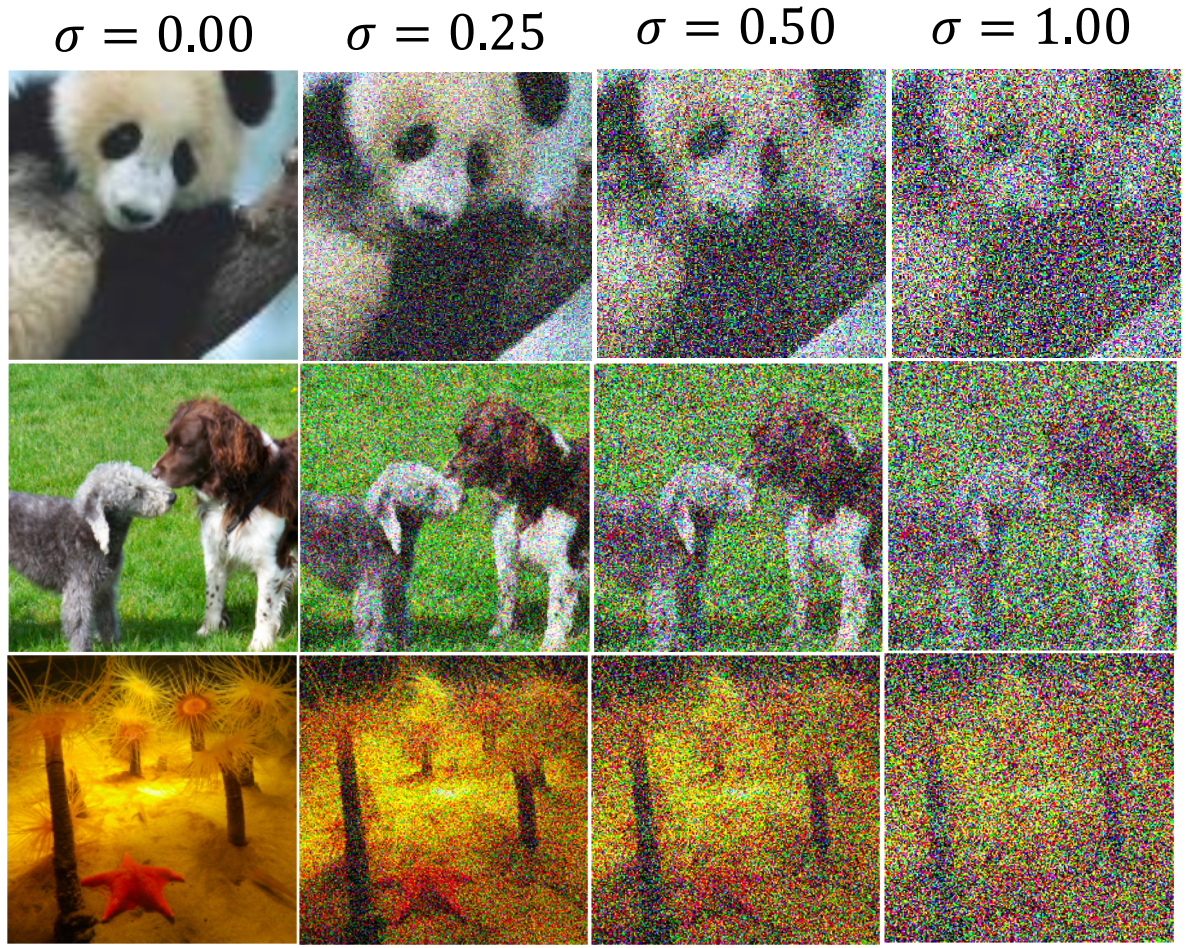
# There's one catch

- When $f$ is a neural network, it's not possible to exactly
  - evaluate the smoothed classifier
  - certify the robustness of the smoothed classifier

- However, by sampling the prediction of $f$ under Gaussian noise, you can obtain answers guaranteed to be correct with arbitrarily high probability

# ImageNet performance




Note: the certified radii are much smaller than this noise.

# Thanks for listening!

Poster #64,  6:30 PM – 9:00 PM tonight

Code and trained models:
`http://github.com/locuslab/smoothing`