NOTES

- Call for actions
  - In-class presentation sign-ups
    - Do not forget to check-in slides with Sanghyun, ~0.5 week before your presentation day
  - Checkpoint presentation I (on the 30th)
    - 10 min presentation + 3 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 (+ how each member will contribute to the work)
  - No class before the presentation day (Tuesday, the 28th)

CS 499/579: Trustworthy ML

(Certified) defenses against adversarial examples

Sanghyun Hong

sanghyun.hong@oregonstate.edu

Oregon State University

SAIL
Secure AI Systems Lab

HOW CAN WE DEFEAT ADVERSARIAL ATTACKS?

DEFENSES SO FAR

- Existing defenses
  - Defensive distillation
  - Feature squeezing
  - Adversarial training

DEFENSES SO FAR

- Existing defenses
  - Defensive distillation
  - Feature squeezing
  - Adversarial training
  - Many more on heuristics… but broken if one relies on “obfuscated gradients”

Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)	0%*
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)	0%*
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)	9%*
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)	55%**
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	15%

DEFENSES SO FAR

- Existing defenses
  - Defensive distillation
  - Feature squeezing
  - Adversarial training
  - Many more on heuristics… but broken if one relies on “obfuscated gradients”

Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)	0%*
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)	0%*
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)	9%*
[illegible]	[illegible]	[illegible]	[illegible]%**
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	15%

How Can We Make Sure They Are “Provably” Robust?

"PROVABLY" ROBUST

- Research questions:
  - What does it mean by your model is robust?
  - How can you make your model provably robust?
  - How can you certify that your model is robust?
  - How can we make the certification computationally feasible?

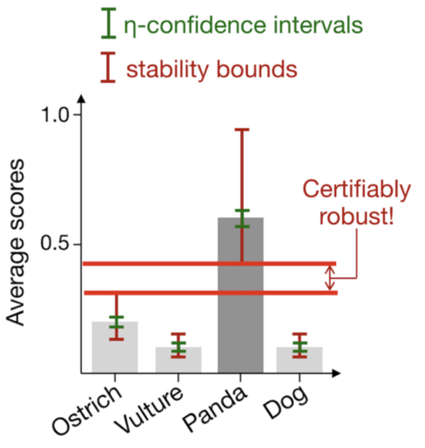
HOW CAN WE MAKE MODELS "PROVABLY" ROBUST?

CERTIFIED ADVERSARIAL ROBUSTNESS VIA RANDOMIZED SMOOTHING, COHEN ET AL., ICML 2019

WHAT DOES IT MEAN BY "PROVABLY" 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 $\mathrm{P}[f(x+\delta) = y_M] > \max_{y \neq y_M} \mathrm{P}[f(x+\delta) = y]$

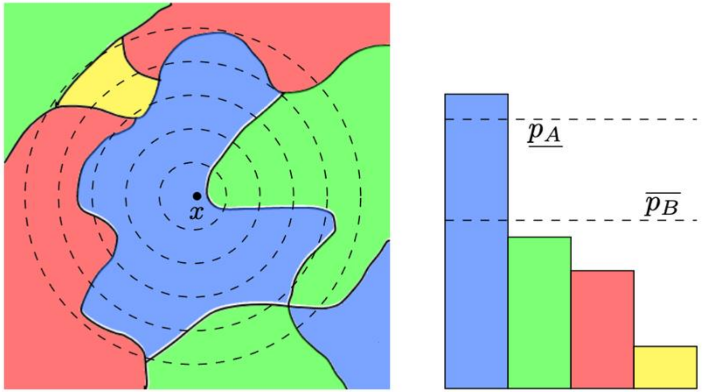


(b) Robustness Test Example

What does it mean by "provably" 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:
  - Most probable class: $\mathrm{P}[f(x+\delta) = c_A] \approx P_A$
  - A runner-up class : $\max_{y \neq y_M} \mathrm{P}[f(x+\delta) = y] \approx P_B$
  - "Provably" robust : $P_A > P_B$



HOW CAN YOU MAKE YOUR MODEL PROVABLY ROBUST?

- Randomized Smoothing:
  - Make a neural network f less sensitive to input details
  - Prior work:
    - Adversarial training (or robust training)
    - Denoising (we will talk about it in a bit later)

- Smoothing
  - In image processing: reducing noise (high frequency components)
  - In our context: reduce noise in inputs

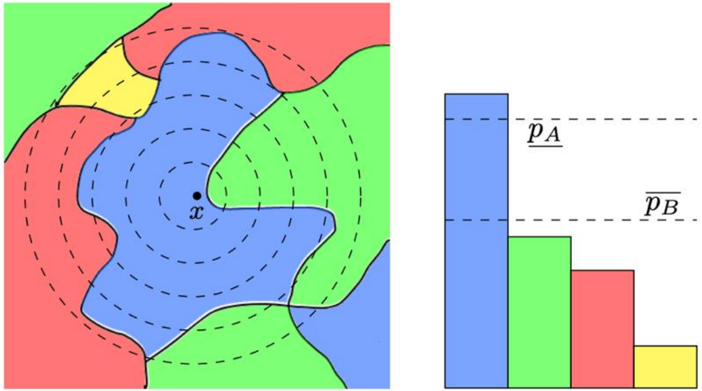
- Randomized
  - In statistics: the practice of using chance methods (random)
  - In this context: add Gaussian *random* noise to the input

How can you make your model provably robust?

- Certified robustness
  - Randomized smoothing transforms a base classifier f into a smoothed classifier g
  - The smoothed classifier g is robust around x with the l_2 radius of R

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

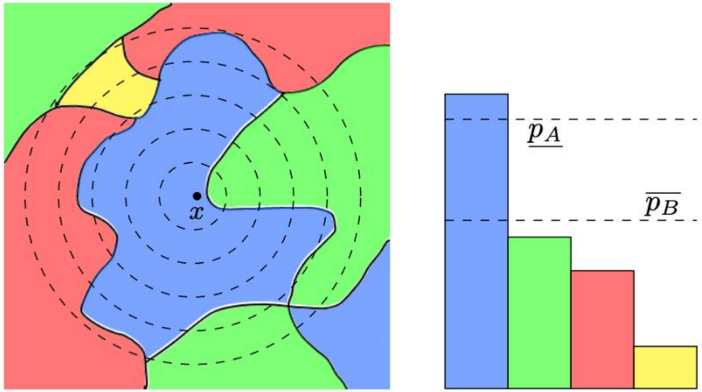
- Certification
  - g is a smoothed classifier
  - g outputs a prediction of c_A (a class)
  - within radius R around x
  - with a confidence of α



HOW CAN YOU MAKE YOUR MODEL PROVABLY ROBUST?

- Certification
  - g is a smoothed classifier
  - g outputs a prediction of c_A (a class)
  - within radius R around x
  - with a confidence of α

- Observations
  - R becomes large when we use high noise
  - R becomes infinite as $P_A \approx 1$ and $P_B \approx 0$



How can we certify the robustness?

- Practical algorithms for prediction and certification

Pseudocode for certification and prediction

 # evaluate g at x
function PREDICT(f, σ, x, n, α)
  `counts` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n, σ)
  $\hat{c}_A, \hat{c}_B \leftarrow$ top two indices in `counts`
  $n_A, n_B \leftarrow$ `counts`$[\hat{c}_A]$, `counts`$[\hat{c}_B]$
  if 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 CERTIFY(f, σ, x, n_0, n, α)
  `counts0` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n_0, σ)
  $\hat{c}_A \leftarrow$ top index in `counts0`
  `counts` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n, σ)
  $\underline{p_A} \leftarrow$ LOWERCONFBOUND(`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

How can we certify the robustness?

- Practical algorithms for prediction and certification

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

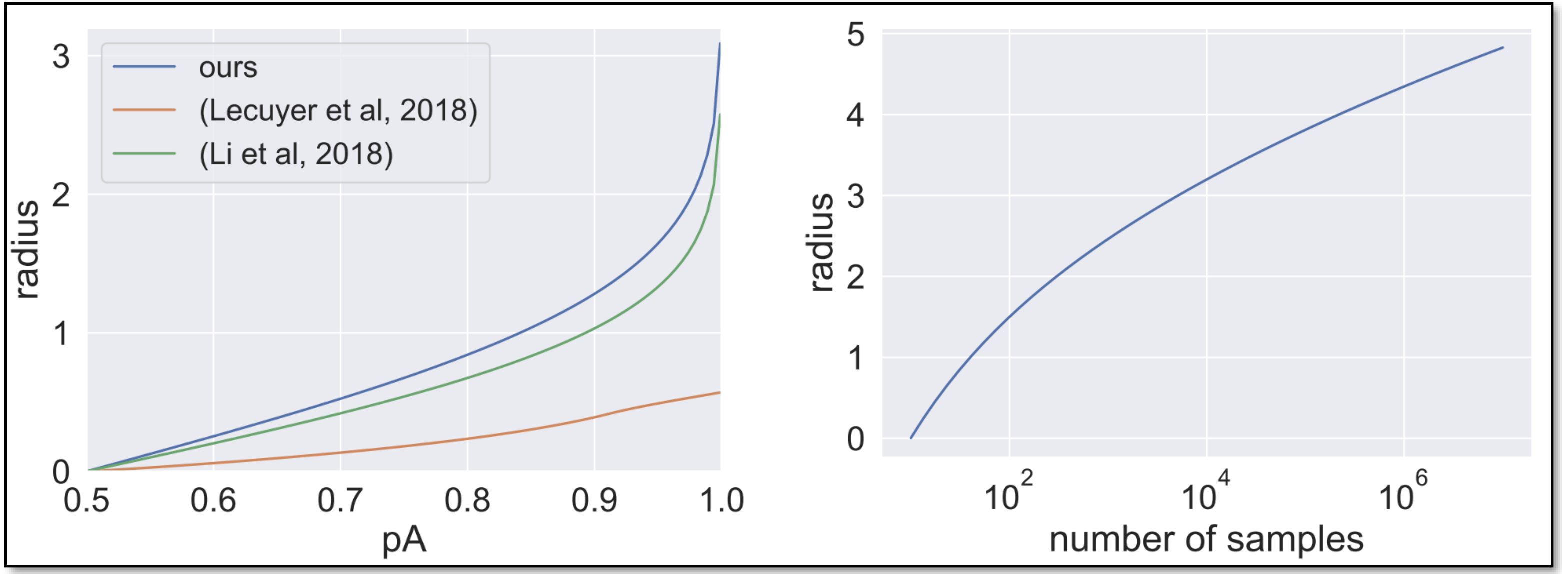
 # certify the robustness of g around x
function $\textsc{Certify}(f, \sigma, x, n_0, n, \alpha)$
  $\texttt{counts0} \leftarrow \textsc{SampleUnderNoise}(f, x, n_0, \sigma)$
  $\hat{c}_A \leftarrow$ top index in $\texttt{counts0}$
  $\texttt{counts} \leftarrow \textsc{SampleUnderNoise}(f, x, n, \sigma)$
  $\underline{p_A} \leftarrow \textsc{LowerConfBound}(\texttt{counts}[\hat{c}_A], n, 1 - \alpha)$
  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

HOW CAN WE CERTIFY THE ROBUSTNESS?

- Practical algorithms for prediction and certification (empirical observation)
  - R becomes infinite as $P_A \approx 1$ and $P_B \approx 0$
  - The paper's algorithm offers a tighter estimation of R
  - The approximation of R becomes accurate if we use more samples



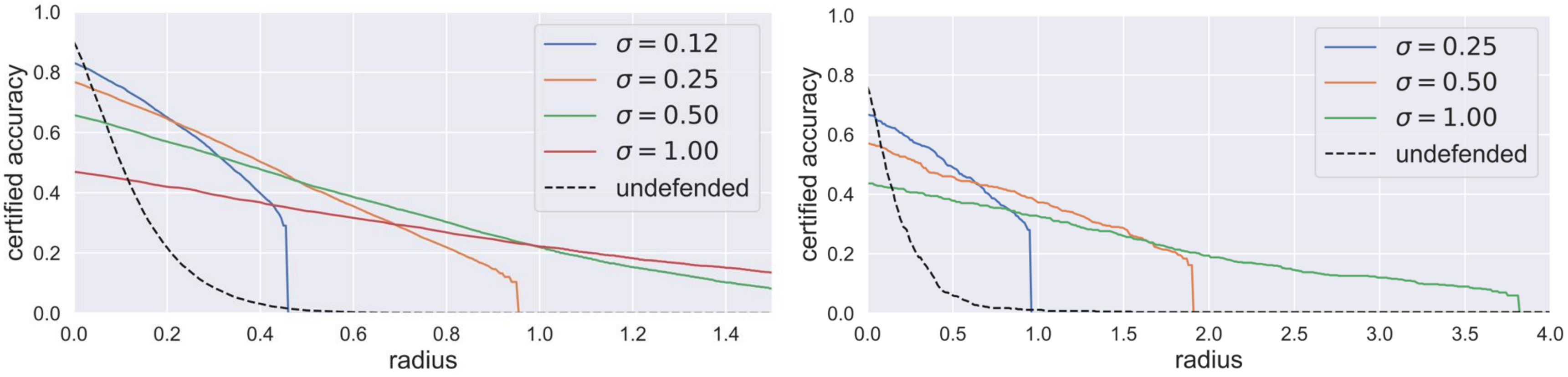
How can we certify the robustness?

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

- Measure
  - Certified test-set accuracy under a radius R with a confidence of α
  - Under various smoothing factor σ (std. of Gaussian noise used)

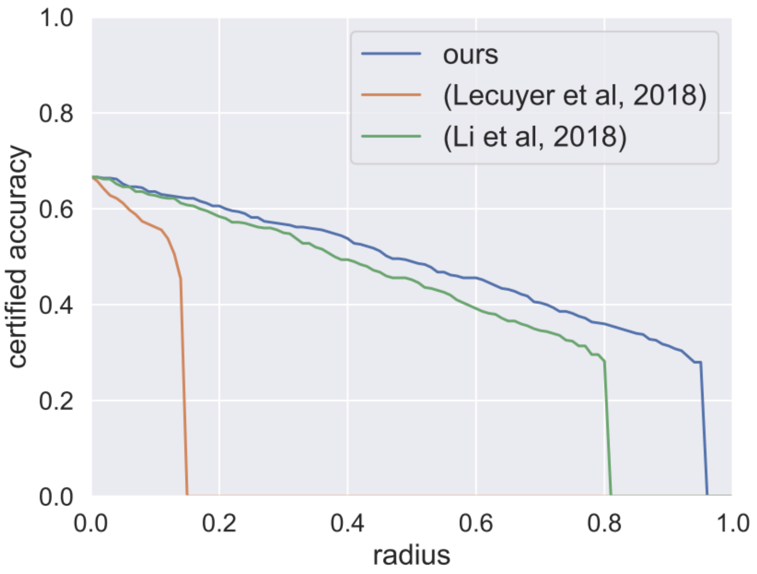
How can we certify the robustness?

- Radius R vs. certified accuracy
  (left: CIFAR10, right: ImageNet)



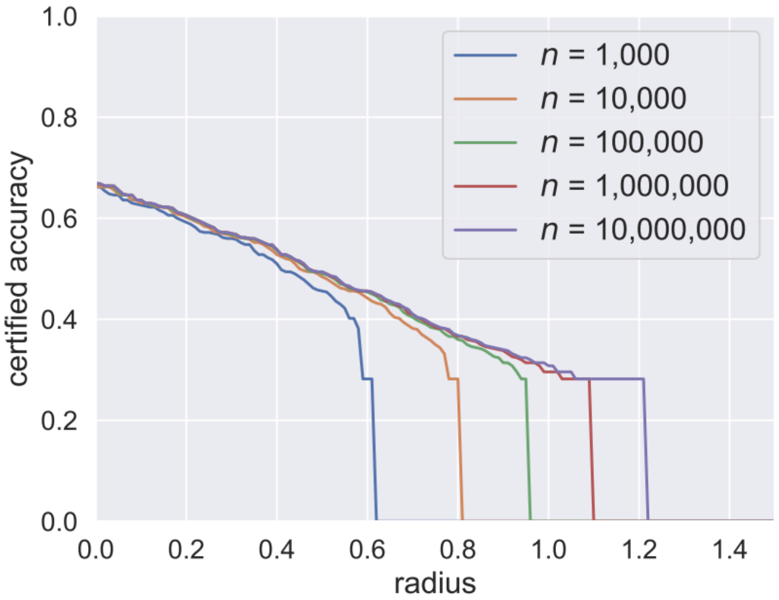
HOW CAN WE CERTIFY THE ROBUSTNESS?

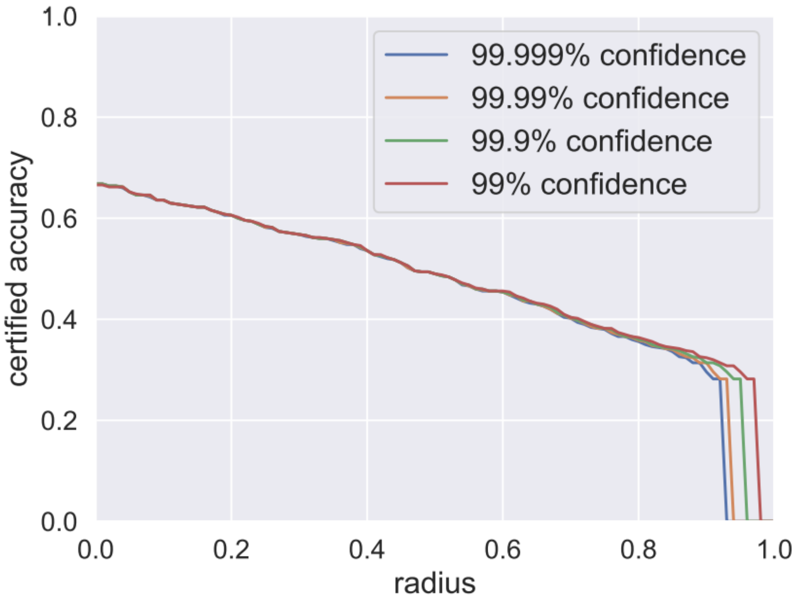
- Certified accuracy vs. prior work
  (ImageNet, $\sigma = 0.25$)



How can we certify the robustness?

- Certified accuracy vs. { # samples or confidence α }





"PROVABLY" ROBUST

- Research questions:
  - What does it mean by your model is robust?
    - A classifier f returns a prediction c within a radius R with a confidence α
  - How can you make your model provably robust?
    - Randomized smoothing (by Cohen et al.)
  - How can you certify that your model is robust?
    - Cohen et al., present practical algorithms for prediction and certification

HOW CAN WE MAKE CERTIFIED DEFENSES COMPUTATIONALLY FEASIBLE?

DENOISED SMOOTHING: A PROVABLE DEFENSE FOR PRETRAINED CLASSIFIERS, SALMAN ET AL., NEURIPS 2020

MAKING A SMOOTHED CLASSIFIER

- Conversion to a smoothed classifier g
  - Adversarial (or robust) training
  - Train a classifier f with noised samples $\sim N(x, \sigma^2 I)$ with x's oracle label

- Problem:
  - What if a classifier f is already trained?
  - Should we re-train all the classifiers, already on-service?

- Solution:
  - Denoised smoothing: train a denoiser that works with a pre-trained classifier

DENOISED SMOOTHING

- Conversion to a smoothed classifier
  - Train a denoiser $D_\theta: R^d \rightarrow R^d$ that removes the input perturbations for f
  - Pre-process an input x with the denoiser D_θ before x is fed to f
  - Pre-process step: generate noisy versions of x, denoise, and fed them to f

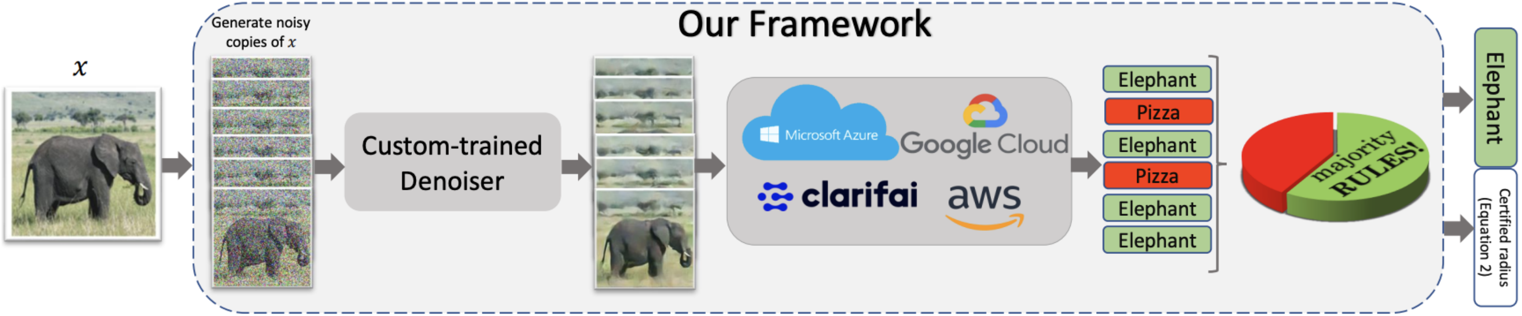


Figure 1: Given a clean image x, our denoised smoothing procedure creates a smoothed classifier by appending a denoiser to any pretrained classifier (e.g. online commercial APIs) so that the pipeline predicts in majority the correct class under Gaussian noise corrupted-copies of x. The resultant classifier is *certifiably* robust against ℓ_2-perturbations of its input.

DENOISED SMOOTHING

- Goal
  - Not to train f on noise
  - But, to provide certification to f
- Denoiser $D_\theta: R^d \rightarrow R^d$
  - $g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}[f(\mathcal{D}_\theta(x+\delta)) = c] \quad \text{where } \delta \sim \mathcal{N}(0, \sigma^2 I)$
- Training D_θ
  - **MSE** objective: Just train D_θ 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 $L_{\text{Stab}} = \mathbb{E}_{\mathcal{S},\delta} \ell_{\text{CE}}(F(\mathcal{D}_\theta(x_i+\delta)), f(x_i))$

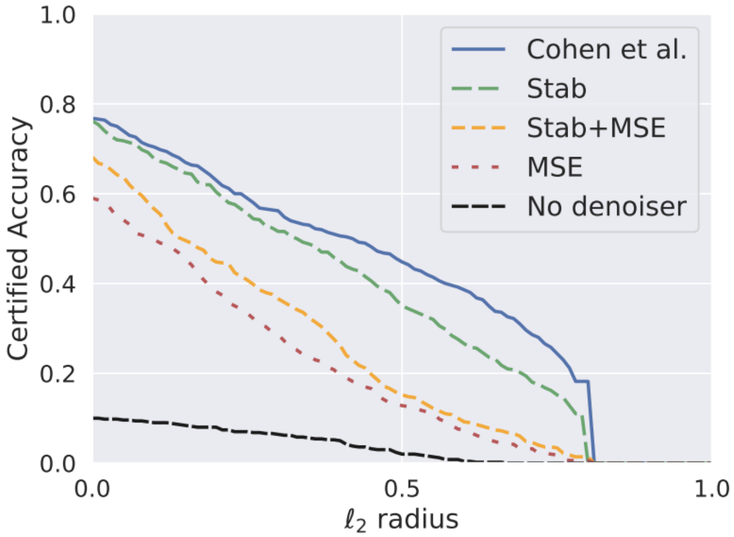
How can we certify the denoiser's robustness?

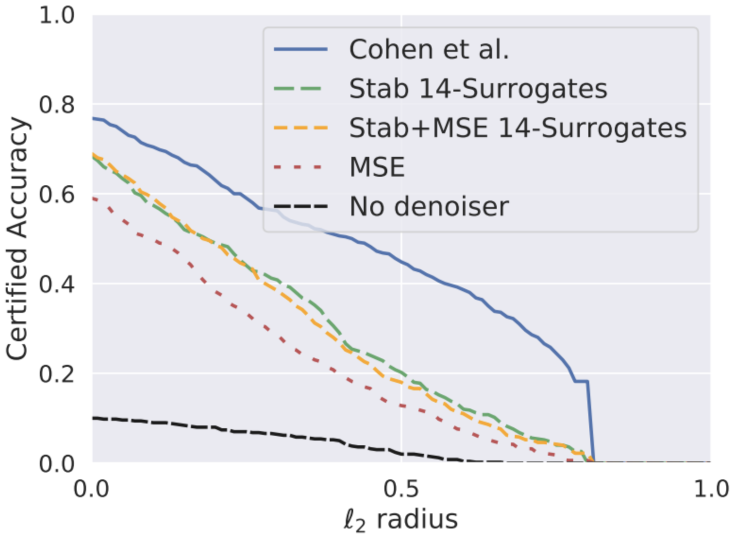
- Setup
  - ImageNet:
    - Pre-trained classifiers: ResNet-18/34/50 (white-box)
    - Baseline: ResNet-110 certified with $\sigma = 1.0$
  - Denoisers: DnCNN and MemNet trained with $\sigma = 0.25, 0.5, 1.0$
  - Objectives: MSE / Stab / Stab+MSE
    - White-box (as-is) | Black-box (14-surrogate models)

- Measure
  - Certified test-set accuracy under a radius R with a confidence of α
  - Under various smoothing factor σ (std. of Gaussian noise used)

HOW CAN WE CERTIFY THE DENOISER'S ROBUSTNESS?

- Certified accuracy vs. prior work (ImageNet, $\sigma = 0.25$)
  - (left: white-box) Denoiser offers certified accuracy close to that of Cohen et al.
  - (right: black-box) The certified accuracy is slightly smaller than the white-box case





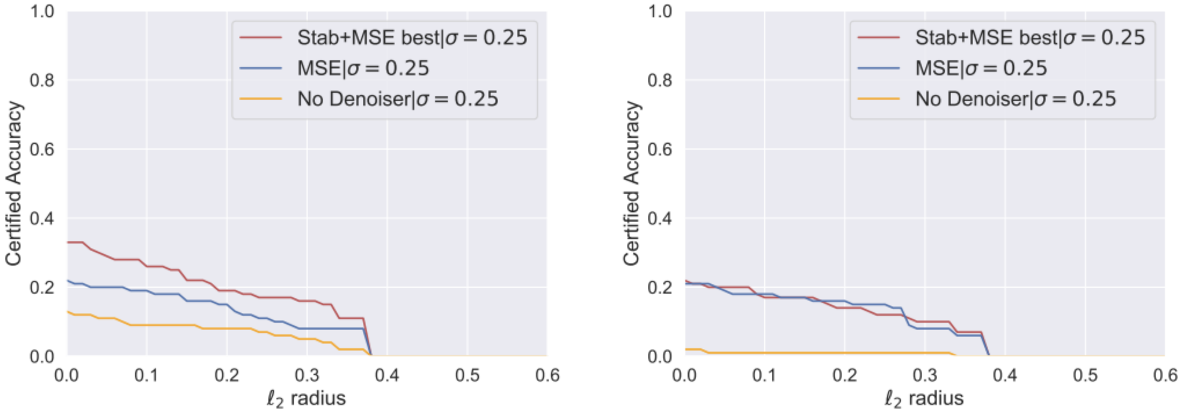
HOW CAN WE CERTIFY THE DENOISER'S ROBUSTNESS?

- Certified accuracy vs. prior work (ImageNet, $\sigma = 0.25$)
  - (left: white-box) Denoiser offers certified accuracy close to that of Cohen et al.
  - (right: black-box) The certified accuracy is slightly smaller than the white-box case

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

Can we certify off-the-shelf models?

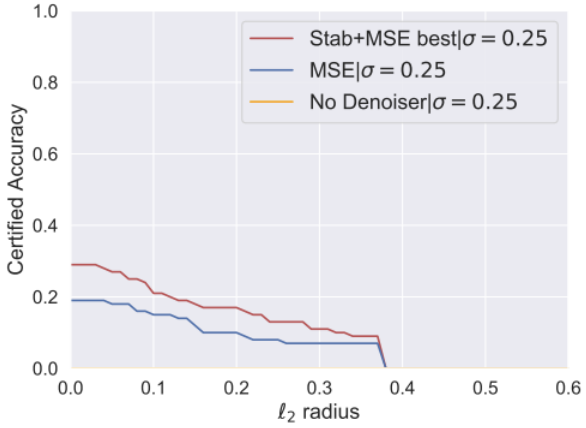
- Radius R vs. certified accuracy (with $\sigma = 0.25$)



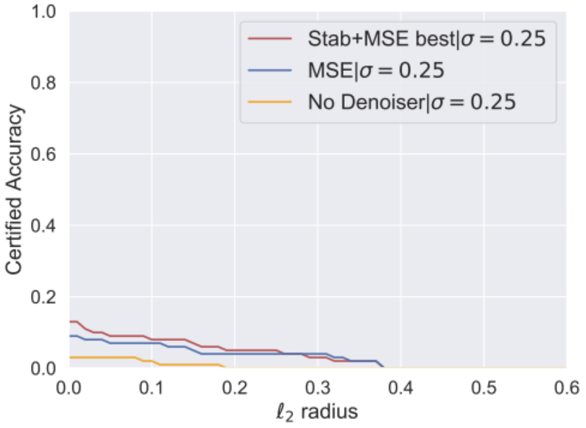
(a) Azure (b) Google Cloud Vision

CAN WE CERTIFY OFF-THE-SHELF MODELS?

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



(c) Clarifai



(d) AWS

How can we get certified defenses for free?

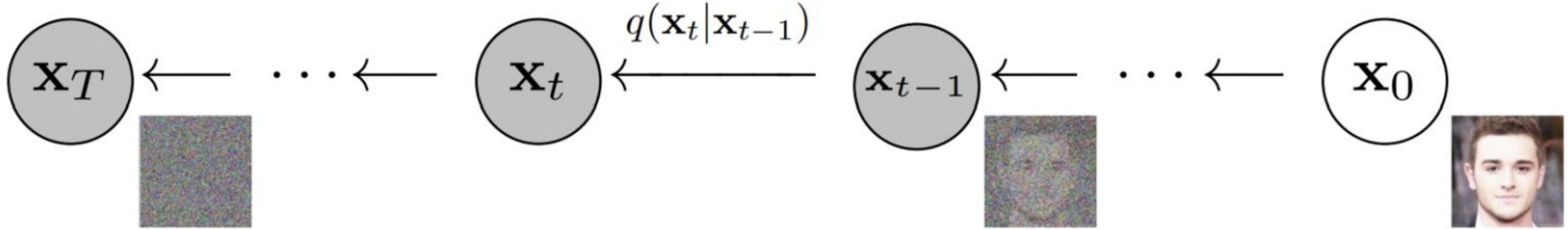
(Certified!!) Adversarial robustness for free!, Calnini et al., ICLR 2023

DENOISED SMOOTHING: WHAT STILL NEEDS COMPUTATIONS?

- Goal
  - Not to train f on noise
  - But, to provide certification to f
- Denoiser $D_\theta : R^d \to R^d$
  - $$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}[f(\mathcal{D}_\theta(x+\delta)) = c] \quad \text{where } \delta \sim \mathcal{N}(0, \sigma^2 I)$$
- Training D_θ
  - **MSE** objective: Just train D_θ to remove Gaussian noise $L_{\text{MSE}} = \mathop{\mathbb{E}}_{\mathcal{S},\delta} \|\mathcal{D}_\theta(x_i+\delta) - x_i\|_2^2$
  - **+ Stability** objective: (White-box) Preserve f's predictions $L_{\text{Stab}} = \mathop{\mathbb{E}}_{\mathcal{S},\delta} \ell_{\text{CE}}(F(\mathcal{D}_\theta(x_i+\delta)), f(x_i))$

WE HAVE PRE-TRAINED DENOISERS

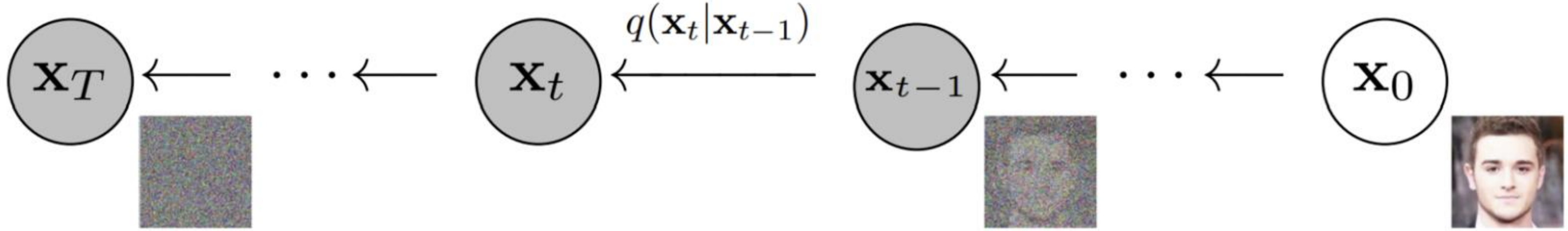
- Denoising diffusion probabilistic models (DDPMs)
  - Generative models trained to gradually denoise the data
  - The *diffusion* process transforms an image x to the purely random noise



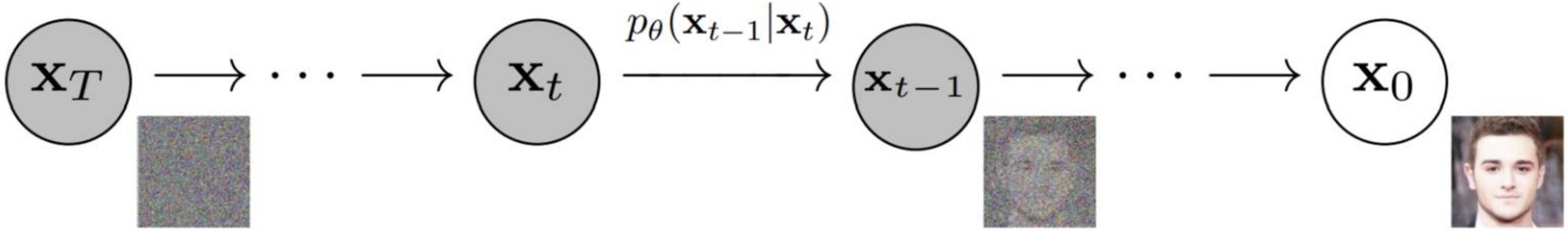
  - Given an image x, the model samples a noisy image: $x_t := \sqrt{\alpha_t} \cdot x + \sqrt{1-\alpha_t} \cdot \mathcal{N}(0, \mathbf{I})$
    α is a constant derived from t and determines the amount of noise to be added

WE HAVE PRE-TRAINED DENOISERS

- Denoising diffusion probabilistic models (DDPMs)
  - Generative models trained to gradually denoise the data
  - The *diffusion* process transforms an image x to the purely random noise



  - The *reverse* process synthesizes x from random Gaussian noise



WE HAVE PRE-TRAINED DENOISERS

- Denoising diffusion probabilistic models (DDPMs)
  - Generative models trained to gradually denoise the data
  - The *diffusion* process transforms an image x to the purely random noise
  - The *reverse* process synthesizes x from random Gaussian noise

- Use DDPMs as a denoiser $D_\theta: R^d \rightarrow R^d$
  - *One-shot* denoising: apply the diffusion model once for a fixed noise level
  - *Multi-step* denoising: apply the diffusion process multiple times

How can we certify the robustness?

- Practical algorithms for prediction and certification

Algorithm 2 Randomized smoothing (Cohen et al., 2019)

PREDICT(x, σ, N, η):
  counts $\leftarrow \mathbf{0}$
  for $i \in \{1, 2, \ldots, N\}$ **do**
    $y \leftarrow$ NOISEANDCLASSIFY(x, σ)
    counts$[y] \leftarrow$ counts$[y] + 1$
  $\hat{y}_A, \hat{y}_B \leftarrow$ top two labels in counts
  $n_A, n_B \leftarrow$ counts$[\hat{y}_A]$, counts$[\hat{y}_B]$
  if BINOMPTEST$(n_A, n_A + n_B, 1/2) \leq \eta$ **then**
    return $\hat{y}_A$
  else
    return Abstain

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

Algorithm 1 Noise, denoise, classify

NOISEANDCLASSIFY(x, σ):
  $t^\star, \alpha_{t^\star} \leftarrow$ GETTIMESTEP(σ)
  $x_{t^\star} \leftarrow \sqrt{\alpha_{t^\star}}(x + \mathcal{N}(0, \sigma^2 \mathbf{I}))$
  $\hat{x} \leftarrow$ denoise$(x_{t^\star}; t^\star)$
  $y \leftarrow f_{\text{clf}}(\hat{x})$
  return y

GETTIMESTEP(σ):
  $t^\star \leftarrow$ find t s.t. $\frac{1-\alpha_t}{\alpha_t} = \sigma^2$
  return $t^\star, \alpha_{t^\star}$

HOW CAN WE CERTIFY THE DENOISER'S ROBUSTNESS?

- Setup
  - Data: CIFAR-10 and ImageNet-21k
  - Model: Wide-ResNet-28-10 (white-box)
  - Denoisers: DDPMs

- Measure
  - Certified test-set accuracy under a radius R with a confidence of α
  - Under various smoothing factor ε (std. of Gaussian noise used)

How can we certify the denoiser's robustness?

- Certified accuracy vs. prior work (ImageNet-21k)
  - DDPM denoisers offer the highest certified accuracy compared to the prior work
  - To achieve the highest accuracy, one can use this off-the-shelf model w/o training

			Certified Accuracy at ε (%)				
Method	Off-the-shelf	Extra data	0.5	1.0	1.5	2.0	3.0
PixelDP (Lecuyer et al., 2019)	○	✗	(33.0) 16.0	-	-		
RS (Cohen et al., 2019)	○	✗	(67.0) 49.0	(57.0) 37.0	(57.0) 29.0	(44.0) 19.0	(44.0) 12.0
SmoothAdv (Salman et al., 2019)	○	✗	(65.0) 56.0	(54.0) 43.0	(54.0) 37.0	(40.0) 27.0	(40.0) 20.0
Consistency (Jeong & Shin, 2020)	○	✗	(55.0) 50.0	(55.0) 44.0	(55.0) 34.0	(41.0) 24.0	(41.0) 17.0
MACER (Zhai et al., 2020)	○	✗	(68.0) 57.0	(64.0) 43.0	(64.0) 31.0	(48.0) 25.0	(48.0) 14.0
Boosting (Horváth et al., 2022a)	○	✗	(65.6) 57.0	(57.0) 44.6	(57.0) **38.4**	(44.6) **28.6**	(38.6) **21.2**
DRT (Yang et al., 2021)	○	✗	(52.2) 46.8	(55.2) 44.4	(49.8) **39.8**	(49.8) **30.4**	(49.8) **23.4**
SmoothMix (Jeong et al., 2021)	○	✗	(55.0) 50.0	(55.0) 43.0	(55.0) **38.0**	(40.0) 26.0	(40.0) 20.0
ACES (Horváth et al., 2022b)	◐	✗	(63.8) 54.0	(57.2) 42.2	(55.6) 35.6	(39.8) 25.6	(44.0) 19.8
Denoised (Salman et al., 2020)	◐	✗	(60.0) 33.0	(38.0) 14.0	(38.0) 6.0	-	-
Lee (Lee, 2021)	●	✗	41.0	24.0	11.0	-	-
Ours	●	✓	(82.8) **71.1**	(77.1) **54.3**	(77.1) **38.1**	(60.0) **29.5**	(60.0) 13.1

How can we certify the denoiser's robustness?

- One-shot vs. multi-step denoising (ImageNet-21k)
  - One-shot denoising offers more faithful results
  - Multi-step denoising destroys the information about the original image

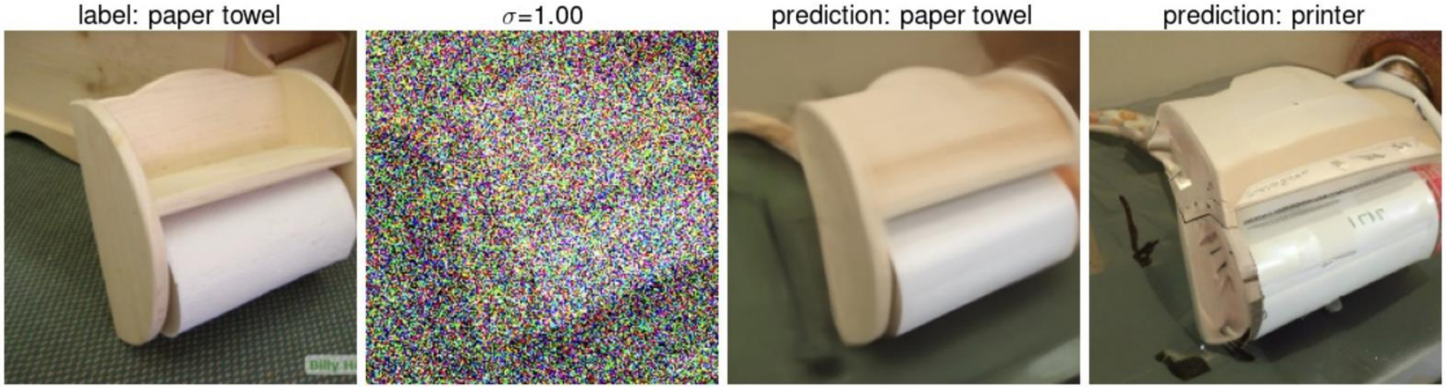


Figure 3: Intuitive examples for why multi-step denoised images are less recognized by the classifier. From left to right: clean images, noisy images with $\sigma = 1.0$, one-step denoised images, multi-step denoised images. For the denoised images, we show the prediction by the pretrained BEiT model.

OTHER WORK ON THE "PROVABLE" ROBUSTNESS

- Further readings
  - PixelDP (Lecuyer *et al.*): Use differential privacy (DP) for the certification
  - Li *et al.*: Propose a tighter bound for the certification, based on Renyi-divergence

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

Thank You!

Sanghyun Hong

https://secure-ai.systems/courses/MLSec/F23

Oregon State University

SAIL
Secure AI Systems Lab