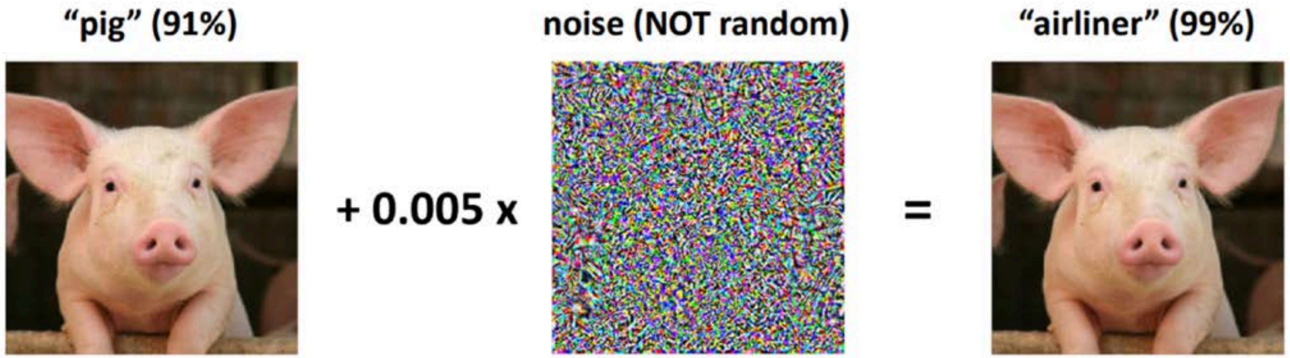
MACER: Attack-Free and Scalable Robust Training via Maximizing Certified Radius

Runtian Zhai, Chen Dan, Di He, Huan Zhang

Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh & Liwei Wang

A provable, fast and scalable adversarial defense

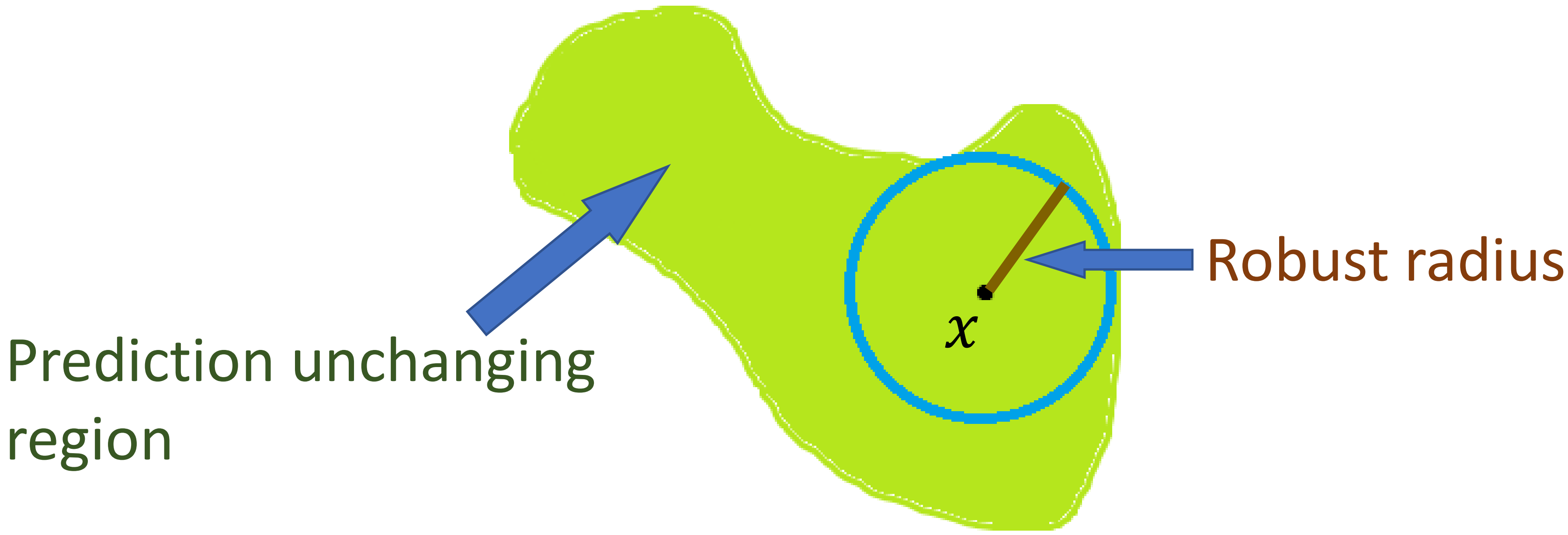
- **Provable**: Model robustness can be certified
- **Fast**: No expensive attack operation in training
- **Scalable**: Applicable to deep neural networks



Robust radius

Training a robust model ⇔ Maximizing the robust radius

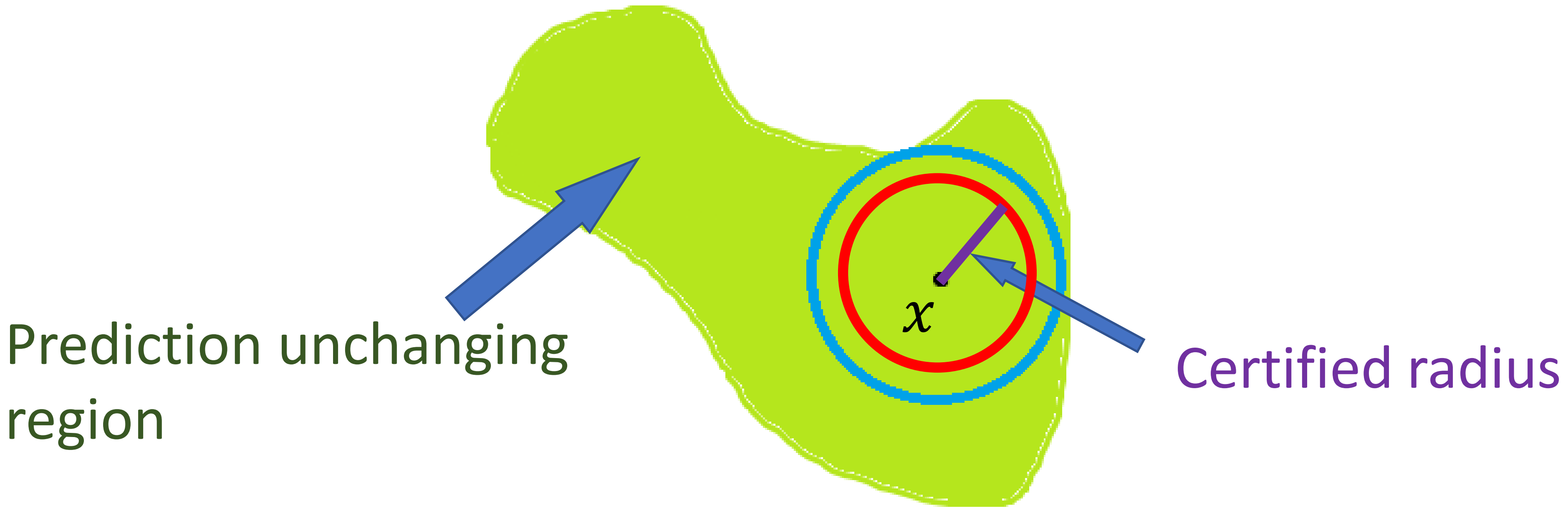
Computing the robust radius is NP-hard



Certified radius

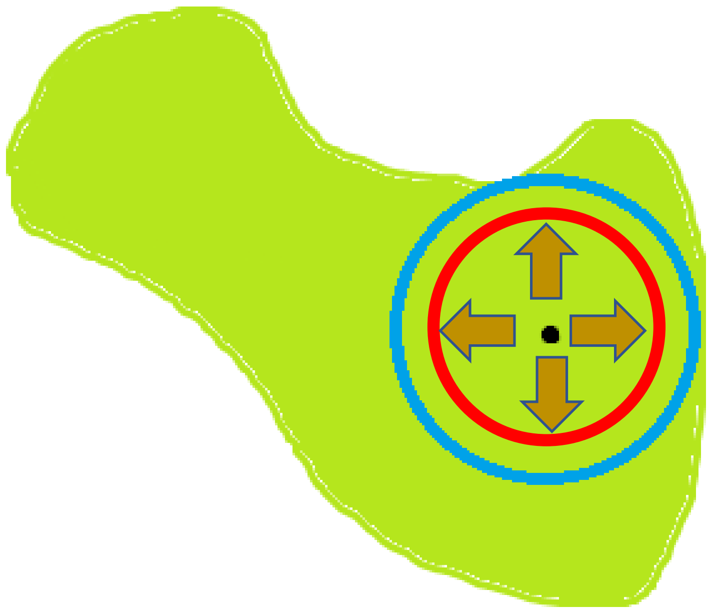
Certified radius: a lower bound of the robust radius

Can be efficiently computed with a certification method



MACER: MAximizing the CErtified Radius

MACER indirectly maximizes the robust radius

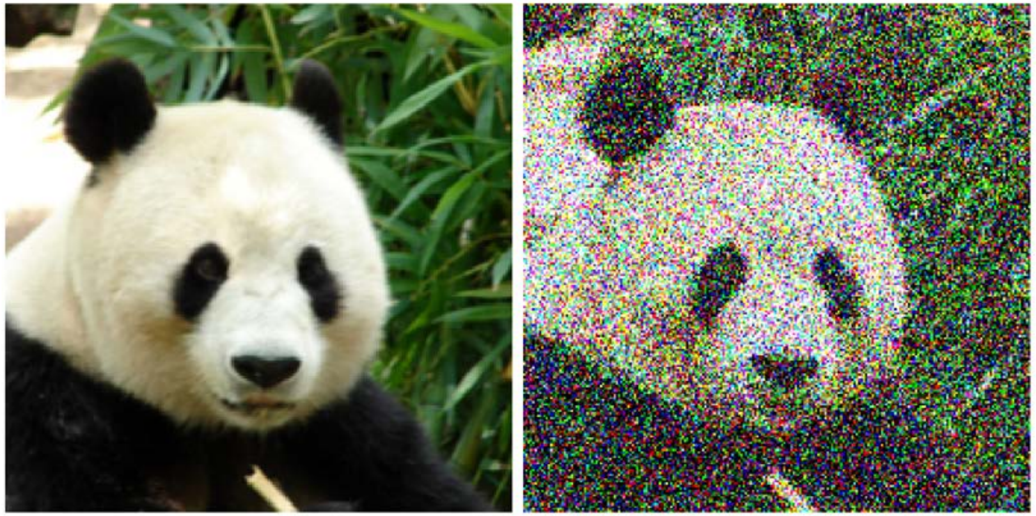


Computing the certified radius via Randomized Smoothing[1]

Smoothed classifier $g(x)$

Base classifier $f(x)$

$$g(x) = \arg\max_c P_{\eta \sim N(0,\sigma^2 I)}(f(x+\eta) = c)$$



Randomized Smoothing Theorem: The certified radius of $g(x)$ is

$$\frac{\sigma}{2}\left[\Phi^{-1}\left(P_{\eta \sim N(0,\sigma^2 I)}(f(x+\eta) = y)\right) - \Phi^{-1}\left(\max_{c \neq y} P_{\eta \sim N(0,\sigma^2 I)}(f(x+\eta) = c)\right)\right]$$

where Φ is the c.d.f. of the standard Gaussian distribution

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

Step 1: Surrogate loss

0/1 robust classification error:

$$\max_{\|\delta\| \leq \epsilon} 1_{\{f(x+\delta) \neq y\}}$$

Surrogate loss:

Classification loss — Robustness loss

$$1_{\{g(x) \neq y\}} + 1_{\{g(x)=y, CR(g;x,y)<\epsilon\}}$$

where $CR(g; x, y)$ is the certified radius

Step 2: Differentiable certified radius

We introduce soft randomized smoothing to make the certified radius differentiable

- Original (hard) randomized smoothing:

$$g(x) = \underset{c}{\operatorname{argmax}}\, P_{\eta \sim N(0,\sigma^2 I)}(f(x+\eta) = c)$$

- Soft randomized smoothing:

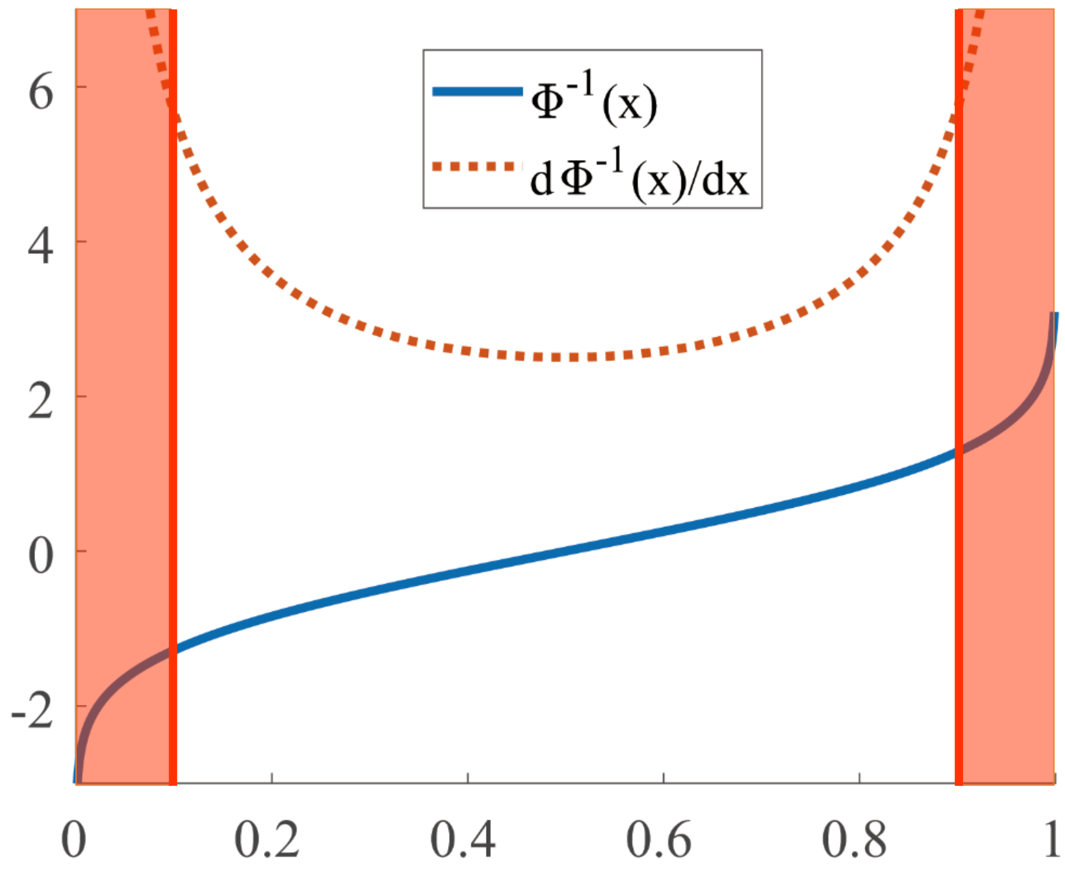
Softmax output

$$\tilde{g}(x) = \underset{c}{\operatorname{argmax}}\, \mathbb{E}_{\eta \sim N(0,\sigma^2 I)} z^c(x+\eta)$$

Step 3: Numerical stability

Use hinge loss to maintain numerical stability

$\Phi^{-1}(x)$ has exploding gradients near 0 and 1



Experimental results

Better performance and faster speed than previous work

Table 3: Training time and performance of $\sigma = 0.25$ models.

Dataset	Model	sec/epoch	Epochs	Total hrs	ACR
Cifar-10	Cohen-0.25 (Cohen et al., 2019)	31.4	150	1.31	0.416
	Salman-0.25 (Salman et al., 2019)	1990.1	150	82.92	0.538
	MACER-0.25 (ours)	504.0	440	**61.60**	**0.556**
ImageNet	Cohen-0.25 (Cohen et al., 2019)	2154.5	90	53.86	0.470
	Salman-0.25 (Salman et al., 2019)	7723.8	90	193.10	0.528
	MACER-0.25 (ours)	3537.1	120	**117.90**	**0.544**

Thank you

Paper

Code