Security and Privacy of ML

Certified Defenses

3/14/2024

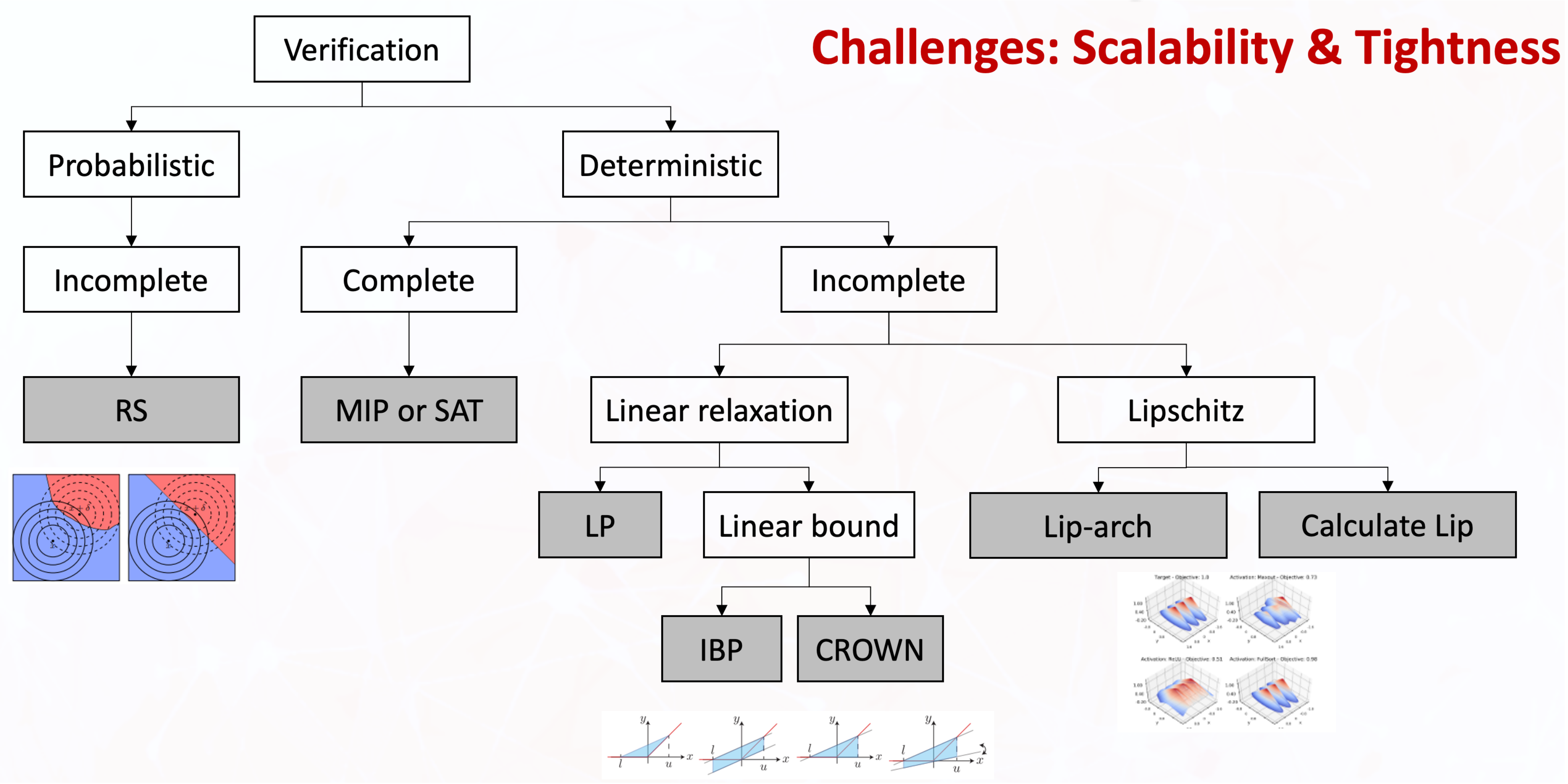
Shang-Tse Chen

Department of Computer Science

& Information Engineering

National Taiwan University

Theoretical verification taxonomy



Li, Linyi, Tao Xie, and Bo Li. "Sok: Certified robustness for deep neural networks." *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023.
Cohen, Jeremy, Elan Rosenfeld, and Zico Kolter. "Certified adversarial robustness via randomized smoothing." *ICML*, 2019.

Certified Robustness

- Given a model f and a test sample (x, y)
- Exact certification:
  - Answer YES if any allowed perturbation can not change y
  - Answer No if successful adversarial perturbation exists
- Relaxed certification:
  - Answer YES if any allowed perturbation can not change y
  - Answer MAYBE if adversarial perturbation may exist

Exact Certification

Can be computed by **Mixed integer linear programming** (MILP)

Linear Programming (LP):

$$\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x}$$

$$\text{s.t. } A\,\mathbf{x} \le \mathbf{b}$$

MILP: some of the x variables are constrained to be integers

Exact Certification by MILP

$$
\begin{aligned}
z_1 &= x \\
z_{i+1} &= \mathrm{ReLU}(W_i z_i + b_i), \qquad i = 1, \ldots, d-1 \\
h_\theta(x) &= W_d z_d + b_d
\end{aligned}
$$

Targeted attack in ℓ_∞ norm can be written as the optimization problem

$$
\begin{aligned}
\underset{z_{1:d}}{\text{minimize}} \quad & \left(e_y - e_{y_{\mathrm{targ}}}\right)^T (W_d z_d + b_d) \\
\text{subject to} \quad & z_{i+1} = \mathrm{ReLU}(W_i z_i + b_i), \qquad i = 1, \ldots d-1 \\
& \|z_1 - x\|_\infty \le \epsilon
\end{aligned}
$$

Exact Certification by MILP

$$\min \; (e_7 - e_0)^T (W_d z_d + b_d)$$
$$\text{s.t.} \;\; \ldots$$

= -2.54 (exists adversarial example for target class zero or another class)

$$\min \; (e_7 - e_1)^T (W_d z_d + b_d)$$
$$\text{s.t.} \;\; \ldots$$

= 3.04 (there is *no* adversarial example to make classifier predict class 1)

Exact Certification by MILP

$$\|z_1 - x\|_\infty \leq \epsilon \quad \Longleftrightarrow \quad \begin{aligned} z_1 - x &\leq \epsilon \\ z_1 - x &\geq -\epsilon \end{aligned}$$

If we have a lower and upper bounds on x, i.e., $x \in [\ell, u]$

z = ReLU(x) $\quad \Longleftrightarrow \quad$

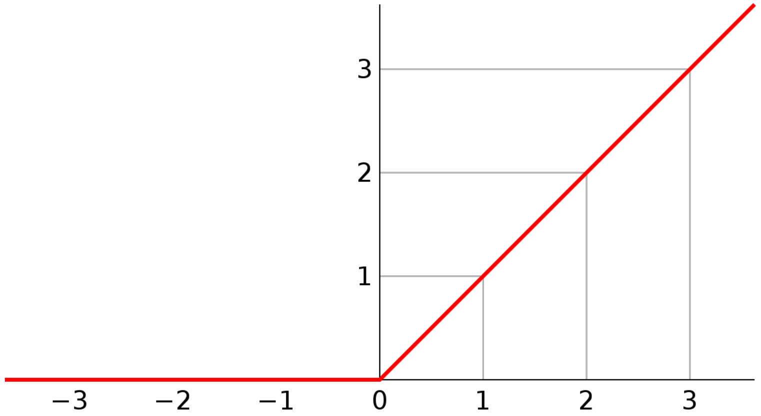
$$a = \mathbf{1}[x \geq 0]$$

$$z \leq x - \ell(1 - a)$$

$$z \leq u \cdot a$$

$$z \geq x$$

$$z \geq 0$$



MILP is NP-hard

- In practice, off-the-shelf solvers (CPLEX, Gurobi, etc) can scale to ~100 hidden units, but size depends heavily on problem structure (including ϵ)
- How do we get the bounds ℓ and u?
  - If $\ell \leq z \leq u$, then

$$[W]_+ l - [W]_- u + b \leq Wz + b \leq [W]_+ u - [W]_- l + b$$

where $W^+ = \max(W, 0)$ and $W^- = \min(W, 0)$

ReLU Stability

- Difficulty of MILP comes from the binary variables
- If $\mathrm{sgn}(\ell) = \mathrm{sgn}(u)$, we can remove the binary variable
- We can add a regularizer to encourage that
  - $-\tanh(1 + \ell \cdot u)$

[Xiao et al. ICLR'19]

- We can also increase weight matrix sparsity, which makes MILP solver run faster

Limitations of Exact Certification

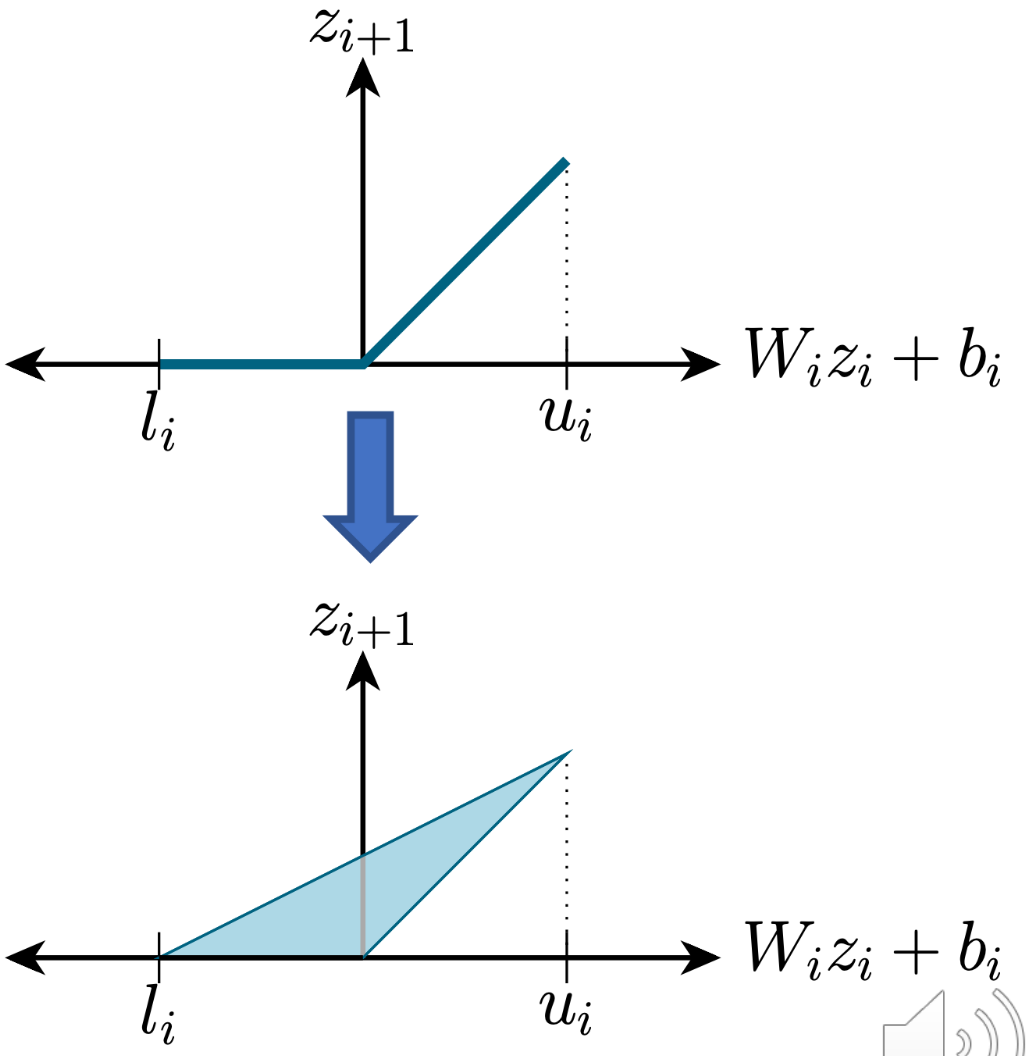
- Still very slow and not scalable
- Can only run on small models, which cannot obtain state-of-the-art robust accuracy

Convex Relaxation

Solving the integer program is too computationally expensive, so let's consider a *convex relaxation*

Replace the bounded ReLU constraints with their convex hull

Optimization problem becomes a *linear program*



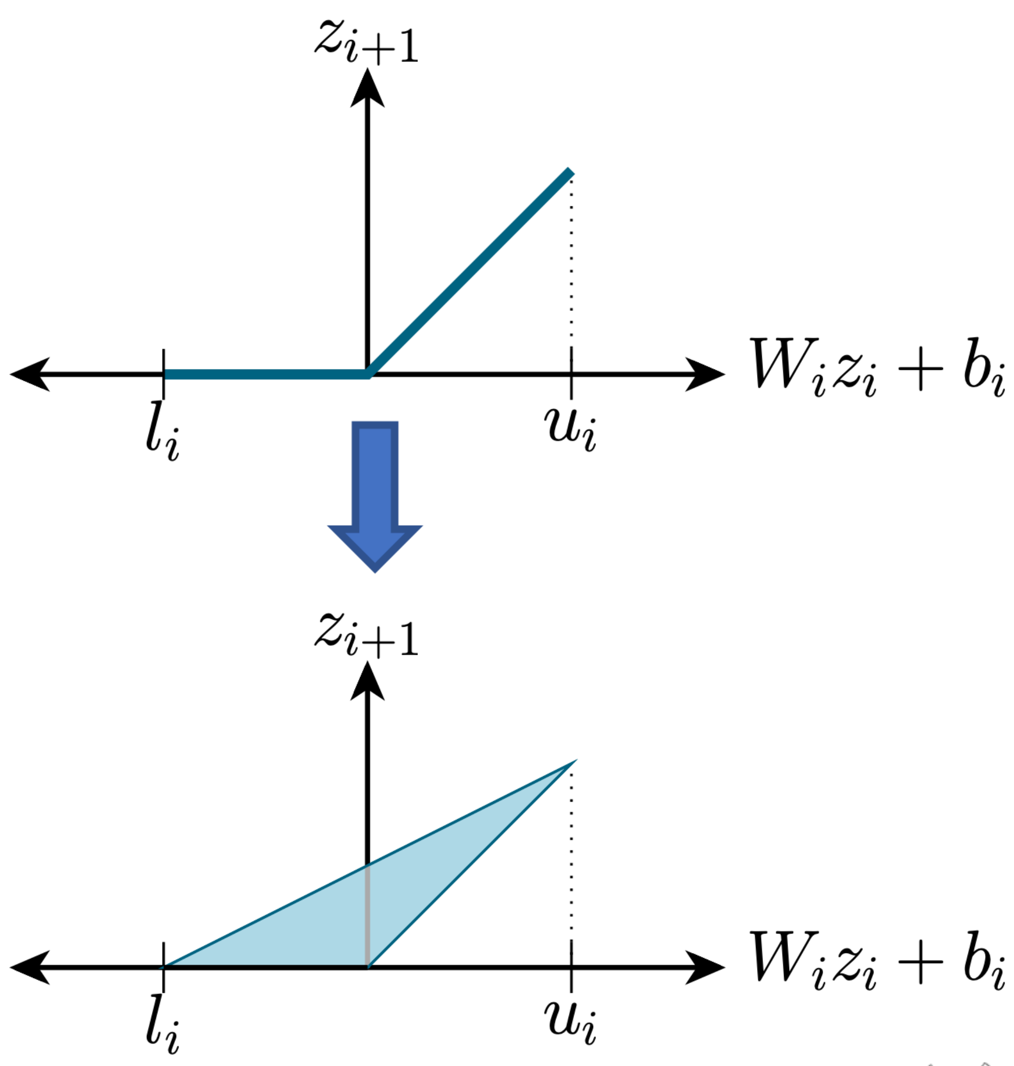
Convex Relaxation

Original constraints:

$$z_{i+1} = \mathrm{ReLU}(W_i z_i + b_i)$$

New constraints:

$$z_{i+1} \geq 0$$
$$z_{i+1} \geq W_i z_i + b_i$$
$$z_{i+1} \leq \frac{u_i}{u_i - l_i}(z_i - l_i)$$

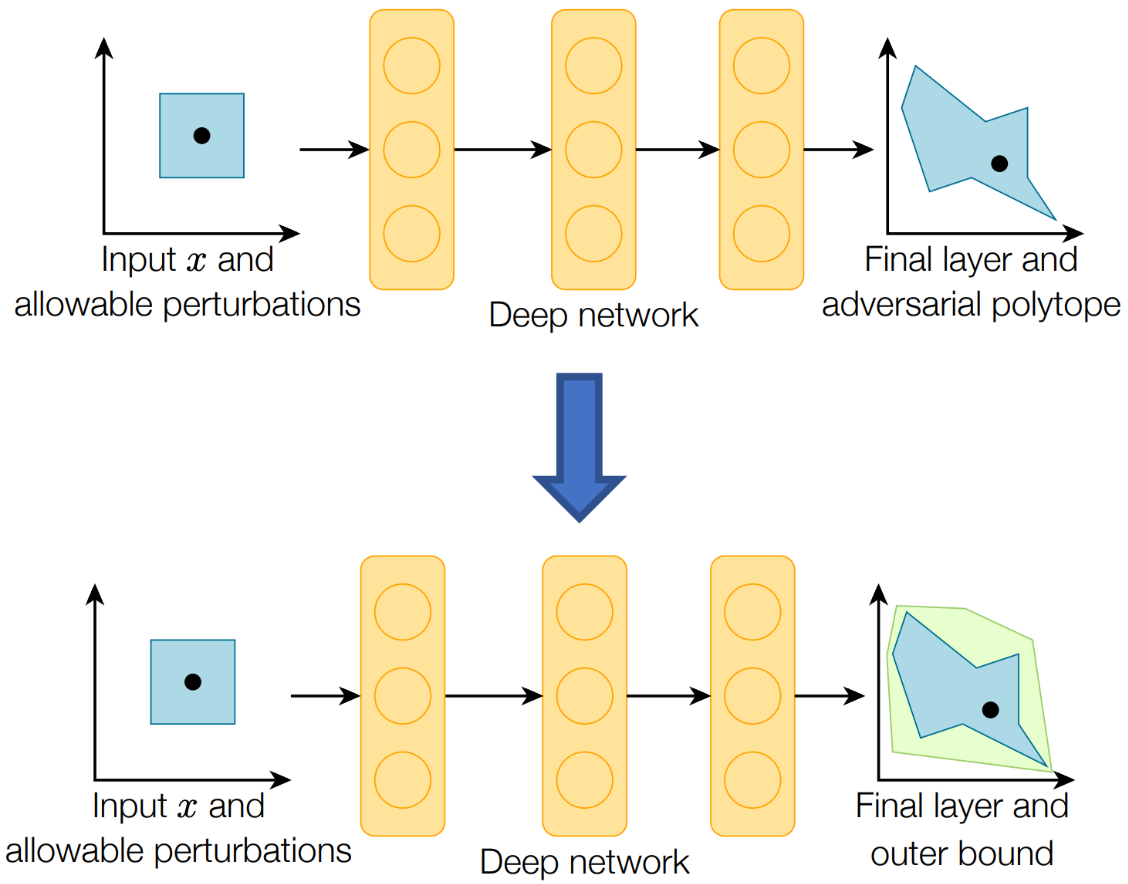


Convex Relaxation

Convex relaxation provides a strict *lower bound* on integer programming objective (because feasible set is larger)

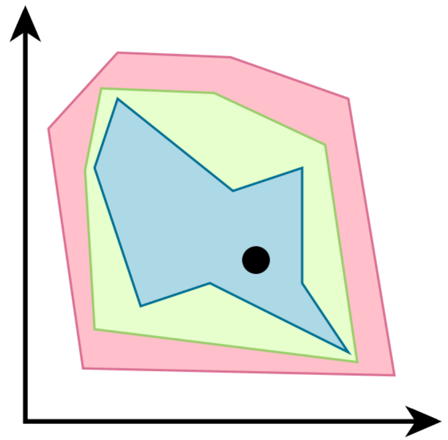
$$\text{Objective}(\text{LP}) \leq \text{Objective}(\text{IP})$$

So if the objective of LP is still positive for all target classes, the relaxation gives a verifiable proof that no adversarial example exists

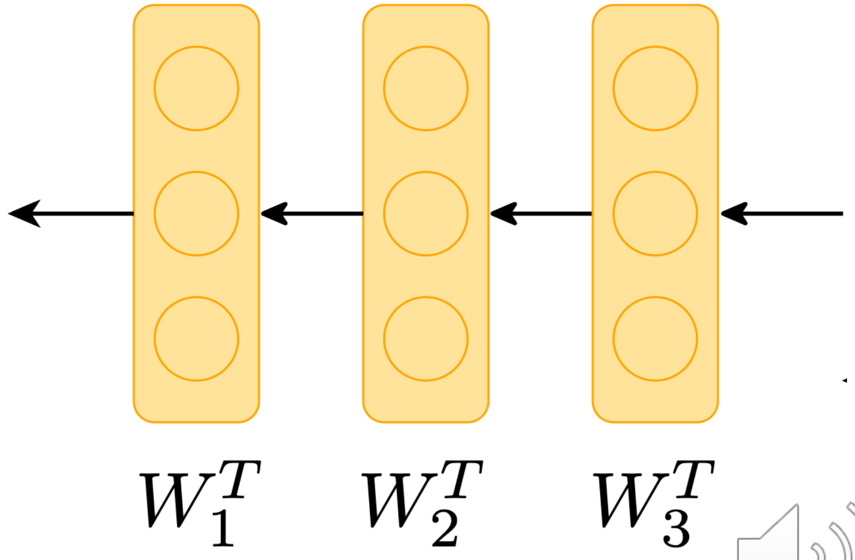


Fast solutions to the relaxation

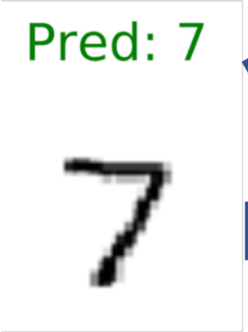
Solving a linear program with size equal to the number of hidden units in the network (once per example), is still not particularly efficient



Using linear programming duality, it is possible to achieve a lower bound on the LP program, via a single backward pass through the network [Wong and Kolter, 2018]



Convex Relaxation



$$\begin{aligned} \min\ & (e_7 - e_0)^T (W_d z_d + b_d) \\ \text{s.t.}\ & \text{Binary IP constraints} \end{aligned}$$

= -2.54 (*exists* adversarial example for target class zero or another class)

$$\begin{aligned} \min\ & (e_7 - e_0)^T (W_d z_d + b_d) \\ \text{s.t.}\ & \text{Convex constraints} \end{aligned}$$

= -6.28 (*may or may not* exist adversarial example)

$$\begin{aligned} \min\ & (e_7 - e_1)^T (W_d z_d + b_d) \\ \text{s.t.}\ & \text{Convex constraints} \end{aligned}$$

= 1.78 (there is *no* adversarial example to make classifier predict class 1)

Brief Summary

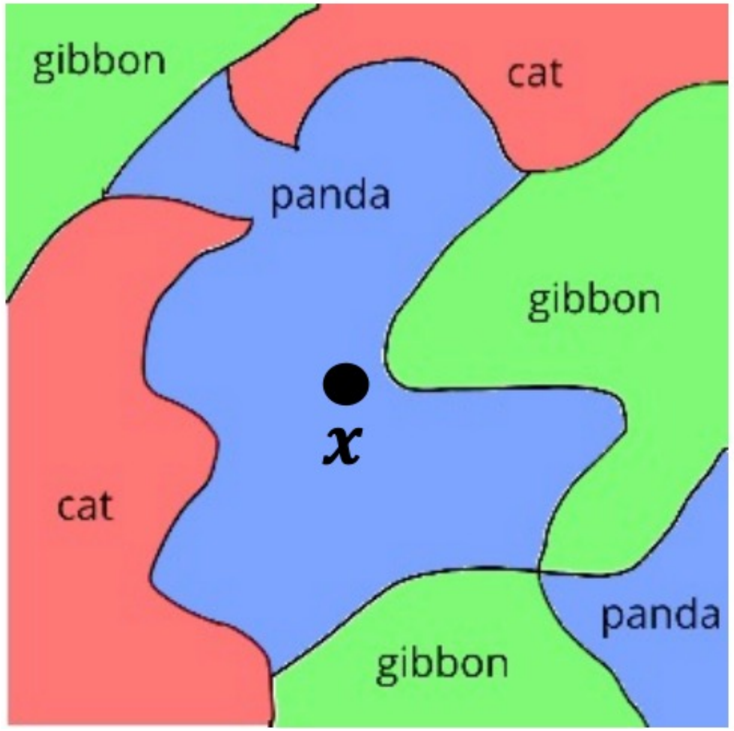
- Certified robustness guarantees that the all allowed perturbations can not change classification output
  - Exact certification via MILP
  - Relaxed certification via convex relaxation to LP

Certified Robustness by Randomized Smoothing

[Cohen et al. ICML'19]

It is easy to adversarially perturb x such that the classifier f misclassifies it as "gibbon"

Decision boundary of classifier f



Randomized Smoothing [Cohen et al. ICML'19]

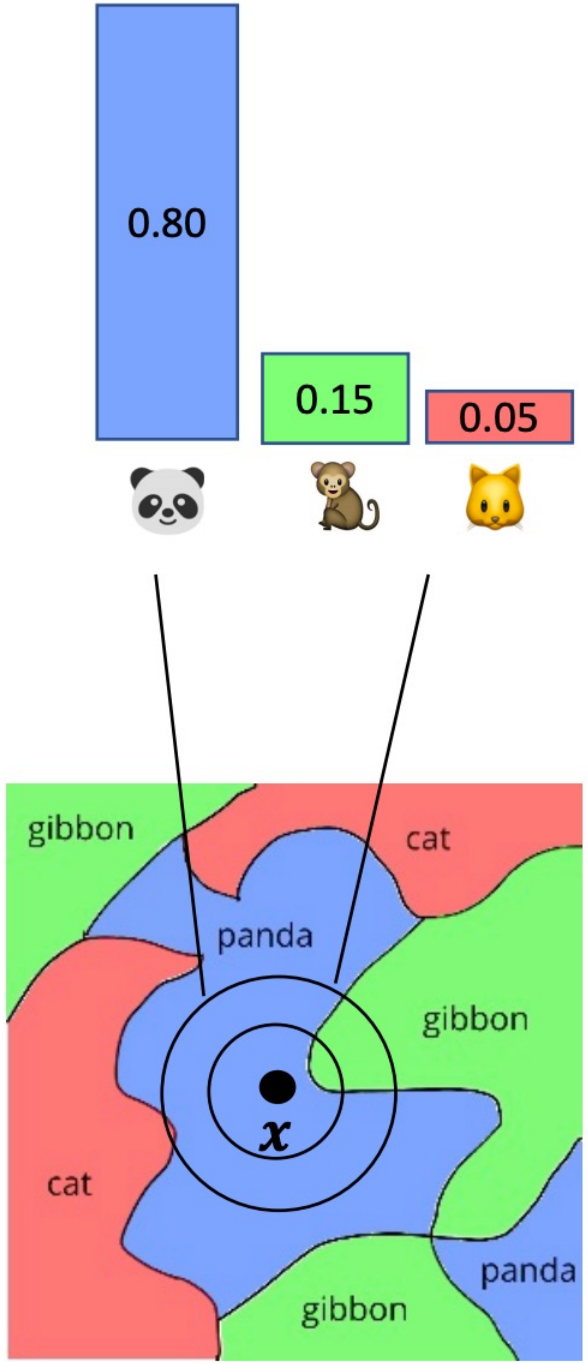
$g(\boldsymbol{x})$ = the most probable prediction by f of random Gaussian corruptions of $\boldsymbol{x}$

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

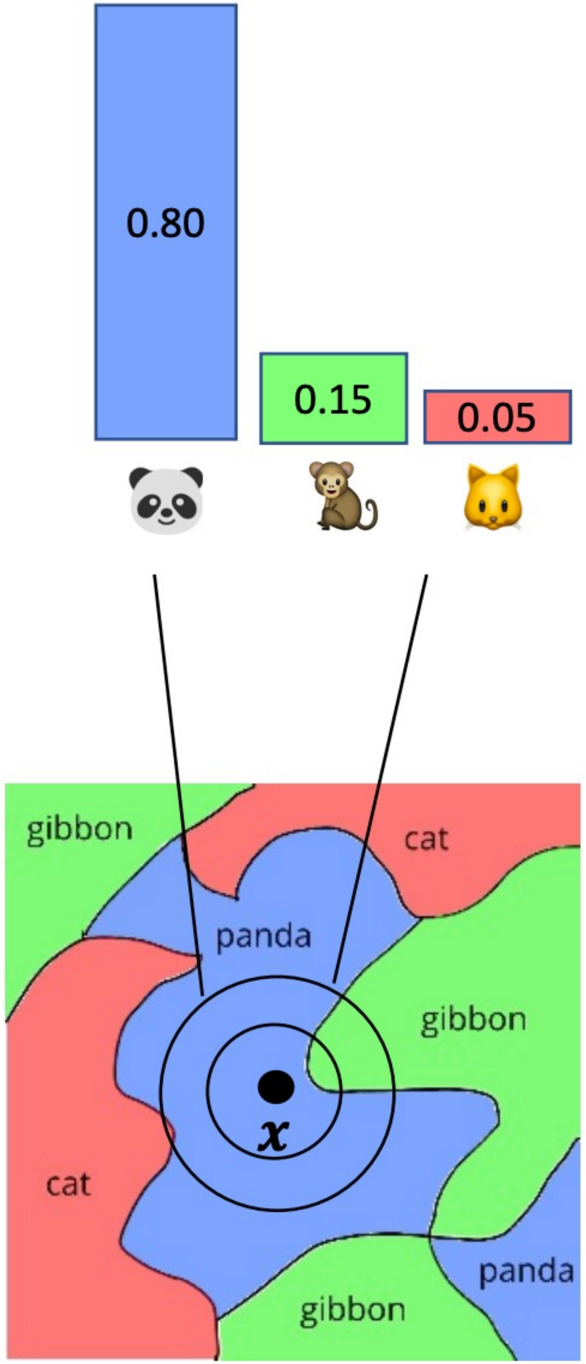
Then $g(\boldsymbol{x}) =$ 🐼



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.

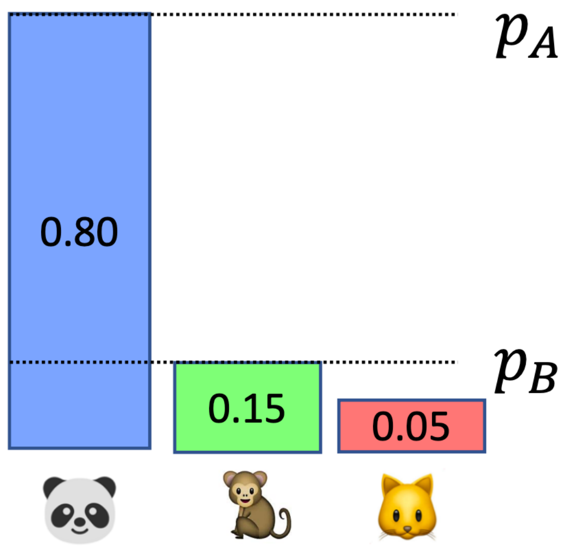


Robustness Guarantee

- 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 ℓ_2 ball around $\boldsymbol{x}$ of radius

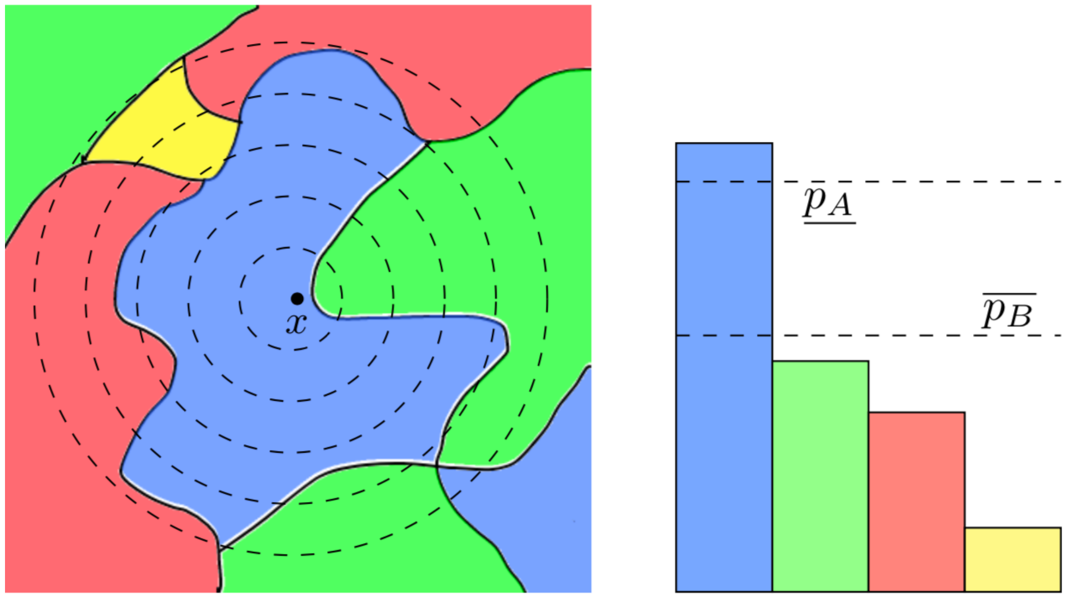
$$R = \frac{\sigma}{2}\left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\right)$$

where Φ^{-1} is the inverse standard Gaussian CDF.



Approximation by Sampling

- When f is a neural network, we can't get the exact probabilities of the smoothed classifier
- Use Monte Carlo sampling to compute upper and lower bounds with high confidence



Gaussian data augmentation

Makes the base classifier f more robust to Gaussian noise

clean image

corrupted by Gaussian noise

Formal Notations

Given a base classifier $f\colon \mathbb{R}^d \to \mathcal{Y}$,

construct a smoothed classifier g as follows:

$$g(x) := \mathbf{argmax}_{c \in \mathcal{Y}} \; \mathbb{P}_\epsilon(f(x+\epsilon) = c)$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$

σ controls the amount of noise

Isotropic Gaussian: restricted co-variance matrix

Robustness Guarantee

Suppose that: $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:

c_A is the most likely class

$$\mathbb{P}_\epsilon(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}_\epsilon(f(x+\epsilon) = c)$$

A **lower bound** on the true highest probability p_A

An **upper bound** on the true second-highest probability p_B

Robustness Guarantee

Suppose that: $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:

$$\mathbb{P}_\epsilon(f(x+\epsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}_\epsilon(f(x+\epsilon) = c)$$

Then:

$$g(x+\delta) = c_A \text{ for all } \| \boldsymbol{\delta} \|_2 < R \qquad \text{where:}$$

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

and Φ^{-1} is the inverse of the standard Gaussian CDF.

Robustness Guarantee

If $x \sim \mathcal{N}(0,1)$ and probability $p \in [0,1]$, then $\Phi^{-1}(p) = v$ s.t. $\mathbb{P}_x(x \leq v) = p$

Φ^{-1} is **monotone**: higher values of p produce higher values for $\Phi^{-1}(p)$

For fixed noise σ, to increase radius R, we want higher $\underline{p_A}$ and lower $\overline{p_B}$.

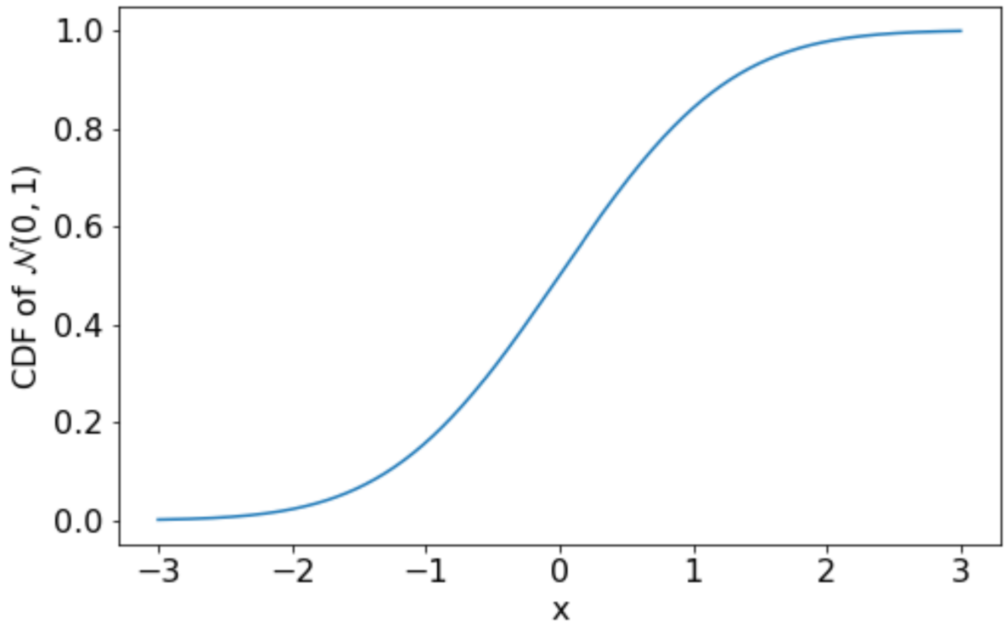
Thus, it is important that classifier f is pre-trained to perform well under Gaussian noise.

Increasing noise σ can increase certified R but can reduce accuracy.

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

Φ^{-1} is the inverse of the standard Gaussian CDF.

Robustness Guarantee



Note: result of $\Phi^{-1}(p)$ can be negative but radius R is always positive due to Φ^{-1} being monotone and the theorem requiring $\underline{p_A} \geq \overline{p_B}$

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

Φ^{-1} is the inverse of the standard Gaussian CDF.

Certified and Standard Accuracy

Note: the certified radius R we obtain may differ between different input x's because the true probabilities p_A and p_B and correspondingly their lower and upper bounds, depend on the input x.

Thus, to compute **certified accuracy**, we pick a target radius T and count the number of points in the test set whose certified radius $R \geq T$ and where the predicted c_A matches the test set label. **Standard accuracy** is instantiated with $T = 0$.

Then:

$$g(x + \delta) = c_A \text{ for all } \| \delta \|_2 < R \qquad \text{where:}$$

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

and Φ^{-1} is the inverse of the standard Gaussian CDF.

Certification Procedure

function $\textsc{Certify}(f, \sigma, x, n_0, n, \alpha)$

$\quad$ `counts0` $\leftarrow \textsc{SampleUnderNoise}(f, x, n_0, \sigma)$

$\quad$ $\hat{c}_A \leftarrow$ top index in `counts0`

$\quad$ `counts` $\leftarrow \textsc{SampleUnderNoise}(f, x, n, \sigma)$

$\quad$ $\underline{p_A} \leftarrow \textsc{LowerConfBound}(\texttt{counts}[\hat{c}_A], n, 1-\alpha)$

$\quad$ **if** $\underline{p_A} > \frac{1}{2}$ **return** prediction $\hat{c}_A$ and radius $\sigma\,\Phi^{-1}(\underline{p_A})$

$\quad$ **else return** ABSTAIN

Certification Procedure

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
`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

To prevent selection bias, sample first to find top label, then sample again with the number of samples $n >> n_0$

SampleUnderNoise(f, x, n, σ):

evaluates f at $x + \epsilon_i$ for $i \in \{1, \ldots, n\}$ where $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$ and returns a dictionary of class counts.

Certification Procedure

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  `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

LowerConfBound($k, n, 1-\alpha$):

assuming $k \sim$ Binomial(n, p) for some unknown p, it returns probability p_l such that $p_l \leq p$ with probability $1-\alpha$. That is, it finds a lower bound on this unknown probability of success p.

There are many methods to compute confidence intervals, the smoothing paper uses Clopper-Pearson.

Certification: Guarantees

function $\textsc{Certify}(f, \sigma, x, n_0, n, \alpha)$
  `counts0` $\leftarrow \textsc{SampleUnderNoise}(f, x, n_0, \sigma)$
  $\hat{c}_A \leftarrow$ top index in `counts0`
  `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

To get the radius:

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

$$= \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) + \Phi^{-1}(\underline{p_A})\right)$$

$$= \sigma\,\Phi^{-1}(\underline{p_A})$$

Certification: Guarantees

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  `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

Then we get the guarantee from the theorem:

with probability at least $1-\alpha$, if CERTIFY returns class $\widehat{c_A}$ and radius $R = \sigma\,\Phi^{-1}(\underline{p_A})$, then $g(x+\delta) = \widehat{c_A}$ for all $\| \delta \|_2 < R$.

Robustness vs. Accuracy

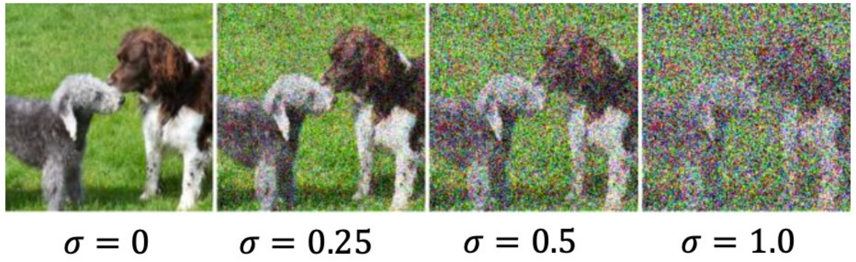
function CERTIFY(f, σ, x, n_0, n, α)
  `counts0` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n_0, σ)
  $\hat{c}_A \leftarrow$ top index in `counts0`
  `counts` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n, σ)
  $\underline{p_A} \leftarrow$ LOWERCONFBOUND(`counts`$[\hat{c}_A]$, n, $1 - \alpha$)
  if $\underline{p_A} > \frac{1}{2}$ **return** prediction $\hat{c}_A$ and radius $\sigma\,\Phi^{-1}(\underline{p_A})$
  else return ABSTAIN

Note: We certify that g returns the same class for all inputs in radius R not that this output is necessarily correct (that is, same label as in the test set)!

There are several reasons why one may obtain an incorrect label (incorrect includes abstentions).

Robustness vs. Accuracy

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



Reason I:

With increasing noise σ, it is more likely that the perfect smoothed classifier $g(x)$ returns c_A which may not be the label in the test set.

Robustness vs. Accuracy

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

Reason II:

Even if the perfect smoothed classifier returns c_A in the test set, it is possible that because: (i) n_0 is small, or (ii) the true probabilities p_A and the next-best probability are similar, we obtain a label $\widehat{c_A}$ which differs from the c_A. And then, this almost certainly will lead to abstention which will be counted as incorrect label.

Effect of Noise σ on Robustness and Accuracy

Each entry shows % of images in the test set (in this case ImageNet images), with provable radius $\geq r$ and label as in test set.

	$r = 0.0$	$r = 0.5$	$r = 1.0$	$r = 1.5$	$r = 2.0$	$r = 2.5$	$r = 3.0$
$\sigma = 0.25$	**0.67**	**0.49**	0.00	0.00	0.00	0.00	0.00
$\sigma = 0.50$	0.57	0.46	**0.37**	**0.29**	0.00	0.00	0.00
$\sigma = 1.00$	0.44	0.38	0.33	0.26	**0.19**	**0.15**	**0.12**

Standard Accuracy

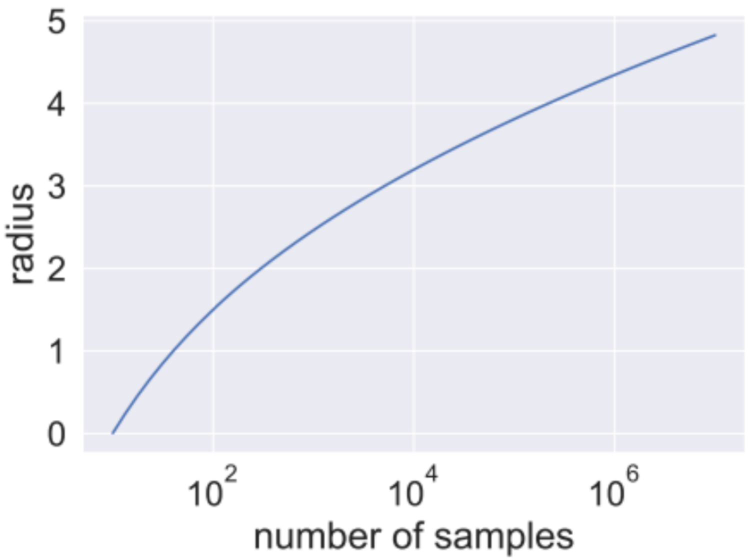
We see that as noise increases, the standard accuracy drops but the certified robust radius increases, the same trade-off between accuracy and robustness we discussed before with adversarial training.

Reminder: all of these results are statistical in nature and not deterministic (due to sampling). That is, they hold with **high probability**.

Increasing Certified Radius for Fixed Noise σ May Require Many Samples

function $\textsc{Certify}(f, \sigma, x, n_0, n, \alpha)$
  `counts0` $\leftarrow \textsc{SampleUnderNoise}(f, x, n_0, \sigma)$
  $\hat{c}_A \leftarrow$ top index in `counts0`
  `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

In the best case scenario where f always classifies to c_A, we have that with confidence $1-\alpha$, a tight $\underline{p_A}$ lower bound is $\alpha^{\frac{1}{n}}$. Plotting the resulting radius $\sigma \cdot \Phi^{-1}(\alpha^{\frac{1}{n}})$ for $\alpha = 0.001$ and $\sigma = 1$, we see that increasing the number of samples will only slowly grow the radius.



Inference

function $\text{PREDICT}(f, \sigma, x, n, \alpha)$
  $\texttt{counts} \leftarrow \text{SAMPLEUNDERNOISE}(f, x, n, \sigma)$
  $\hat{c}_A, \hat{c}_B \leftarrow$ top two indices in $\texttt{counts}$
  $n_A, n_B \leftarrow \texttt{counts}[\hat{c}_A], \texttt{counts}[\hat{c}_B]$
  if $\text{BINOMPVALUE}(n_A, n_A + n_B, 0.5) \leq \alpha$ **return** $\hat{c}_A$
  else return ABSTAIN

Additional work needed at inference time, which can be expensive, depending on the number of samples

The **null hypothesis** is: the true probability of success of a Bernoulli trial is q.

BinomialPValue(i, n, q): returns the p-value of the null hypothesis, evaluated on n statistically independent samples with i successes.

In our case, the null hypothesis: the true probability of f returning $\widehat{c_A}$ is $q = 0.5$ (meaning the classes are indistinguishable).

Inference

function $\text{PREDICT}(f, \sigma, x, n, \alpha)$
  $\texttt{counts} \leftarrow \text{SAMPLEUNDERNOISE}(f, x, n, \sigma)$
  $\hat{c}_A, \hat{c}_B \leftarrow$ top two indices in $\texttt{counts}$
  $n_A, n_B \leftarrow \texttt{counts}[\hat{c}_A], \texttt{counts}[\hat{c}_B]$
  if $\text{BINOMPVALUE}(n_A, n_A + n_B, 0.5) \leq \alpha$ **return** $\hat{c}_A$
  else return ABSTAIN

We accept the null hypothesis if the returned p-value is $> \alpha$

We reject the null hypothesis if the returned p-value is $\leq \alpha$

If α is small (typically 0.001) , then we may often accept the null hypothesis and ABSTAIN, but we will be more confident in our predictions. If α is higher , then we may make prediction more often, but make more mistakes.

Inference: Guarantees

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

We can prove that:

it returns the wrong class $\widehat{c_A} \neq c_A$ with probability at most α

Inference Guarantees: Proof Sketch

$$\mathbb{P}(\widehat{c_A} \neq c_A, \text{no abstain})$$

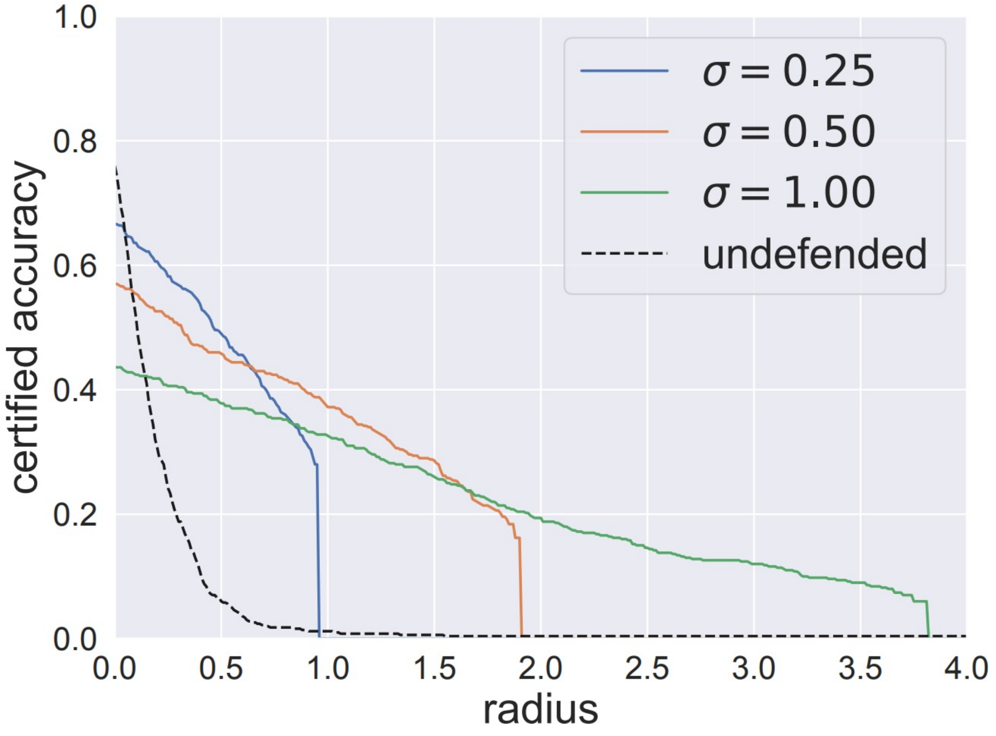
$$= \mathbb{P}(\widehat{c_A} \neq c_A) \cdot \mathbb{P}(\text{no abstain} \mid \widehat{c_A} \neq c_A)$$

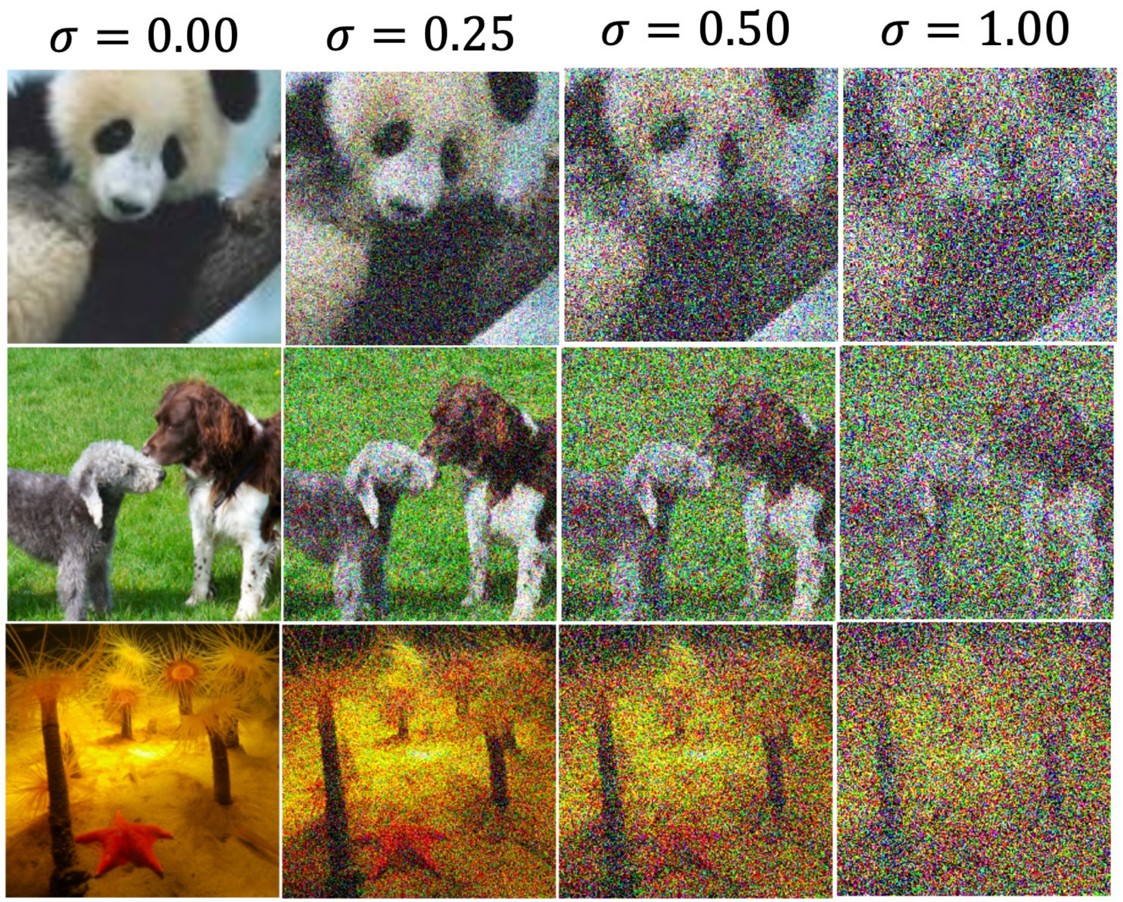
$$\leq \underbrace{\mathbb{P}(\text{no abstain} \mid \widehat{c_A} \neq c_A)}_{= \alpha}$$

see *Rank verification for exponential families*, Hung & Fithian
The Annals of Statistics, 2019
https://arxiv.org/abs/1610.03944

$$\leq \alpha$$

Scalable to ImageNet





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

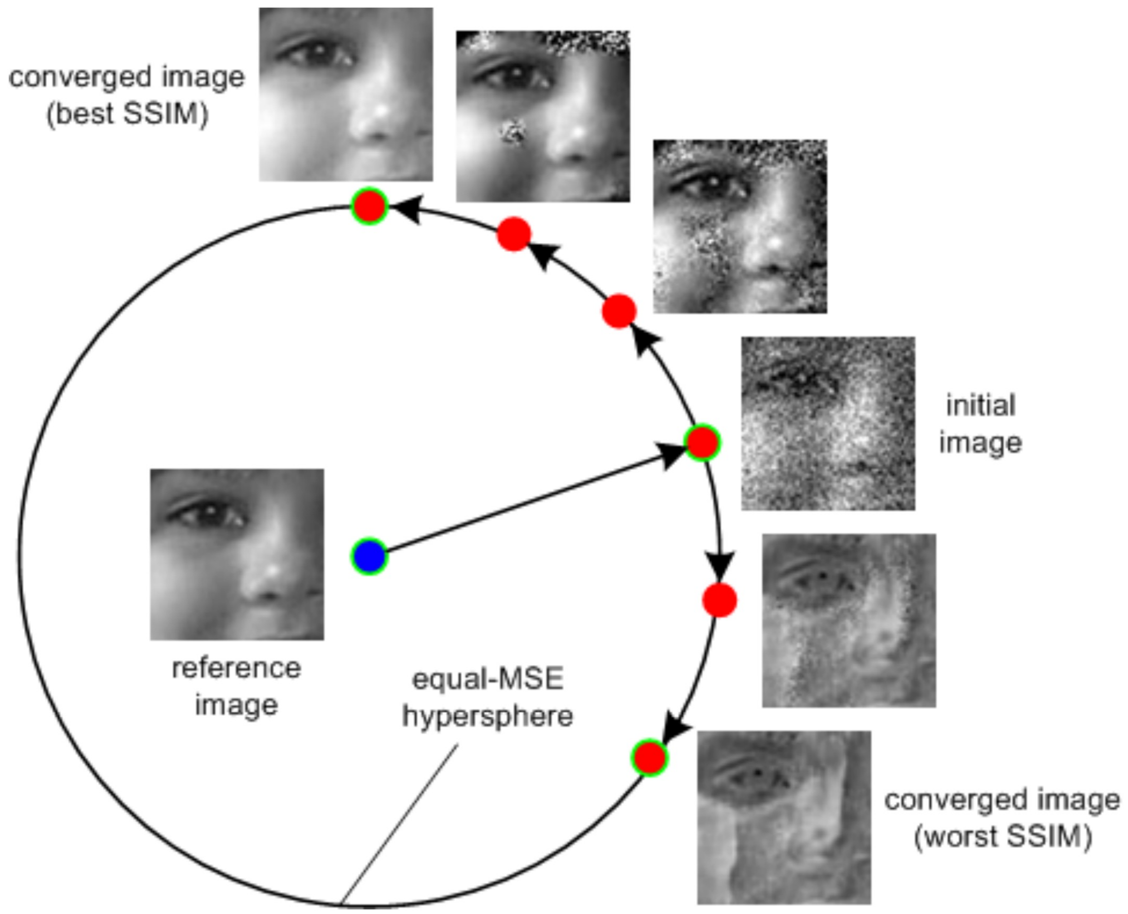
$$R = \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})\right)$$

Future Research Direction

- Extension to other perturbation norms besides ℓ_2
  - Laplace noise for ℓ_1 norm certified robustness
- Improve certified accuracy
  - May be achieved by utilizing base classifier properties

Is Certified Robustness All We Need?

Human vision is far from ℓ_p distance



[Wang & Bovik et al. IEEE signal processing magazine'09]

Small Changes Can Be Semantically Meaningful

[Qin et al. arxiv 2020]

“Breaking” Certified Defenses

[Ghiasi et al. ICLR 2020]

(a) Natural image (x) (b) Adversarial perturbation (δ) (c) Adversarial example ($x + \delta$)

Figure 2: An adversarial example built using our Shadow Attack for the smoothed ImageNet classifier for which the certifiable classifier produces a large certified radii. The adversarial perturbation is smooth and natural looking even-though it is large when measured using ℓ_p-metrics. Also see Figure 16 in the appendix.