Reliable and Trustworthy Artificial Intelligence

Lecture 6: Randomized Smoothing for Robustness

Marc Fischer & Martin Vechev
ETH Zurich

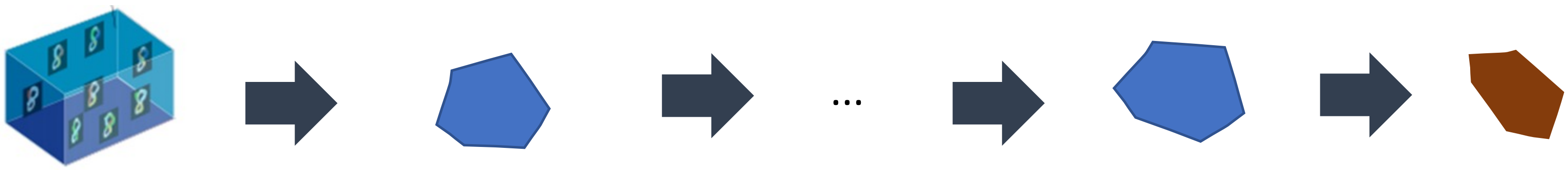
Fall 2022

SRILAB

http://www.sri.inf.ethz.ch

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Deterministic Certification: reminder



1. Infers (typically convex) shapes capturing intermediate invariants. Usually, relaxations are variants of Polyhedra to balance analysis scalability and precision.
2. The method is general. It can handle any safety property (not just robustness).
3. Deterministic guarantees are provided.
4. Very active research area constantly pushing the size of networks.

Key challenge: scaling to large networks

Randomized Smoothing

Key idea: construct a classifier **g** out of an existing classifier **f**, in a way which ensures that **g** has certain statistical robustness guarantees.

The construction does not assume knowledge of **f** and can scale to large networks. The method focuses on restricted robustness-like properties, and requires sampling at inference time, not required by convex methods. The usual standard accuracy vs. robustness trade-off is present here as well.

Certified Adversarial Robustness via Randomized Smoothing, ICML 2019
Cohen, Rosenfeld, Kolter https://arxiv.org/pdf/1902.02918.pdf

Constructing classifier g

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

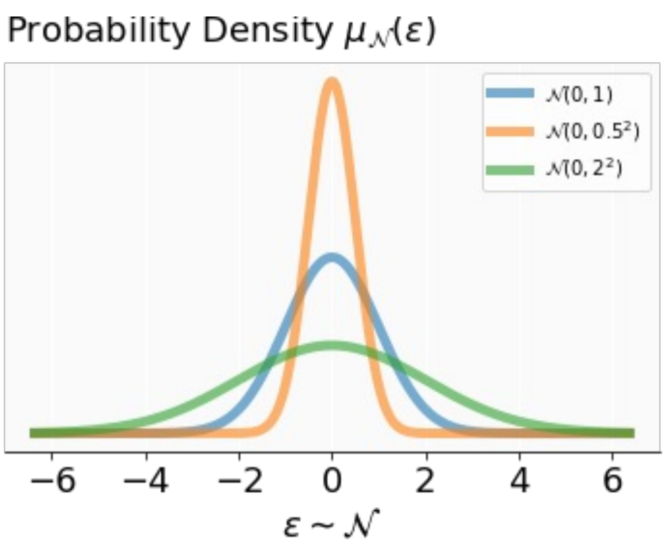
σ controls the amount of noise

Isotropic Gaussian: restricted co-variance matrix (joint distribution product of independent Gaussians).

Reminder: Gaussian Noise

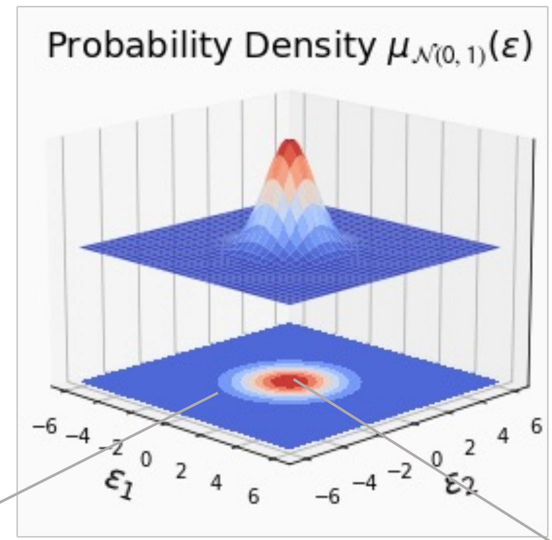
1d Gaussian

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



2d Gaussian

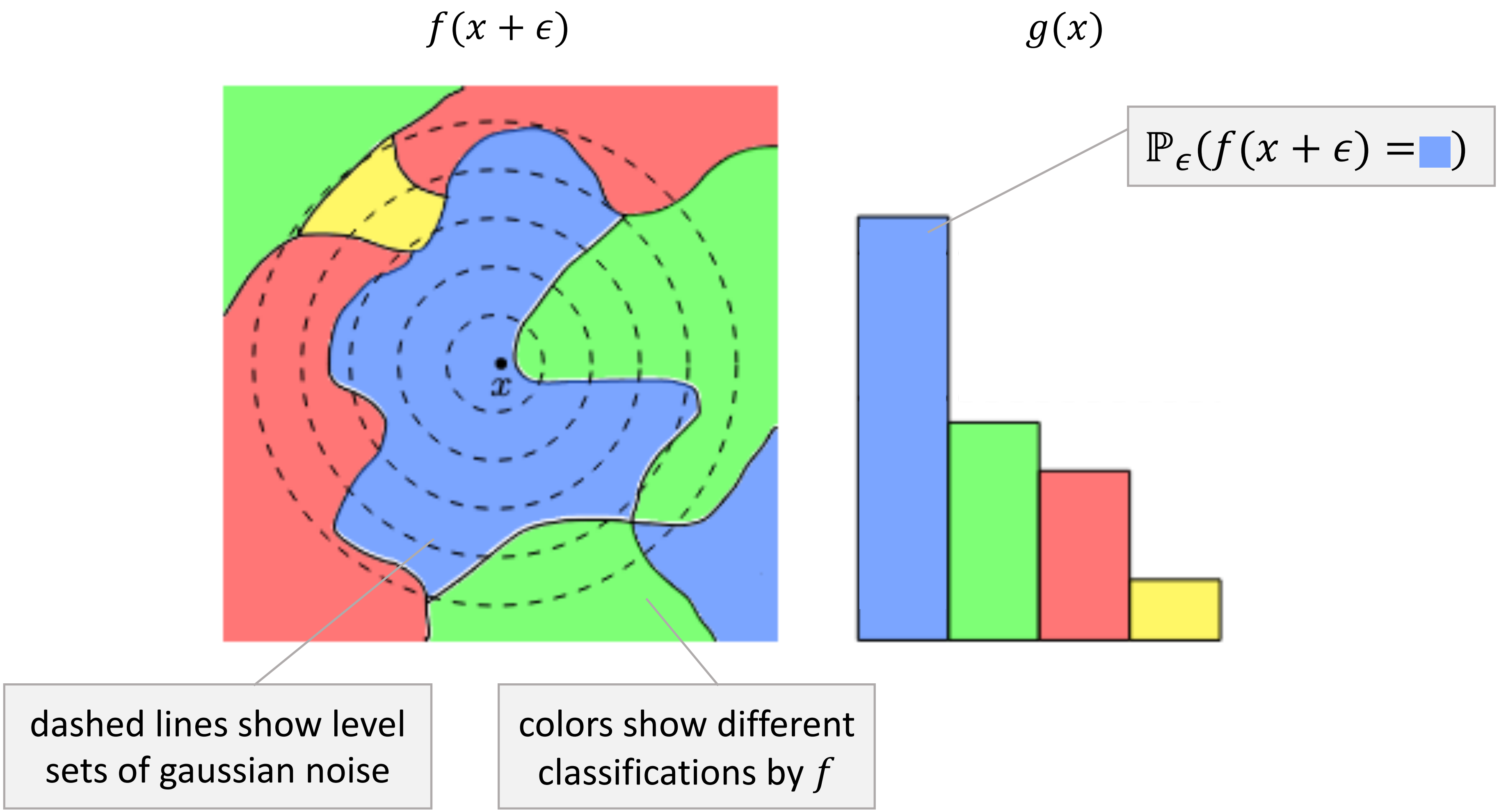
$$\epsilon \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \sigma^2\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right)$$



level-sets of the Gaussian distribution: lines of equal probability (density)

areas close to the center have higher density

Constructing classifier g from f: an intuition



Theorem: Robustness Guarantee

Suppose that: $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$, $c_A \in \mathcal{Y}$ and $\underline{p_{A,x}}$, $\overline{p_{B,x}} \in [0,1]$ satisfy:

$$\mathbb{P}_\epsilon(f(x+\epsilon) = c_A) =: p_A(x) \geq \underline{p_{A,x}} \geq \overline{p_{B,x}} \geq \boldsymbol{\max_{c \neq c_A}} \ \mathbb{P}_\epsilon(f(x+\epsilon) = c)$$

Then:

$$g(x+\delta) = c_A \text{ for all } \| \delta \|_{\mathbf{2}} < R_x \qquad \text{where:}$$

certification radius $R_x := \frac{\sigma}{2}(\Phi^{-1}(\underline{p_{A,x}}) - \Phi^{-1}(\overline{p_{B,x}}))$ for sample x

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

Proof sketch coming later

Theorem: Robustness Guarantee

Suppose that: $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$, $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:

c_A is the most likely class

dropping subscript x for readability

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

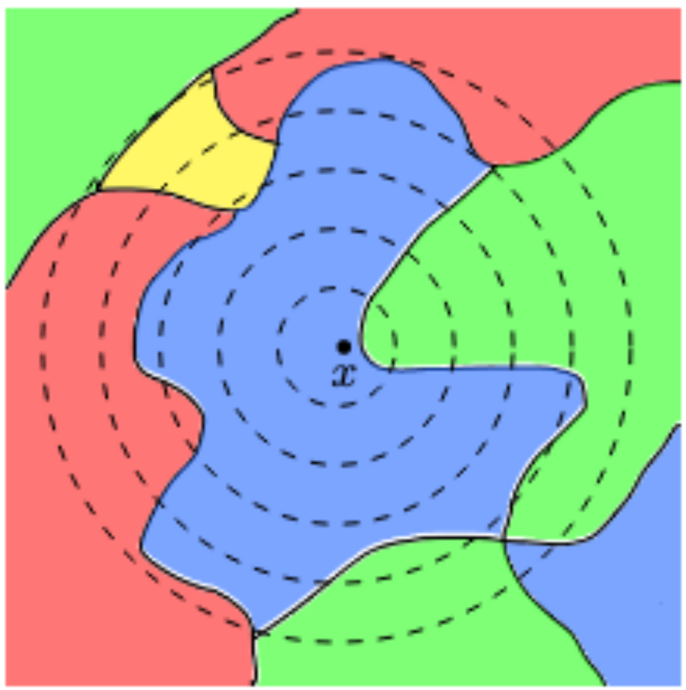
A **lower bound** on the true highest probability $p_A(x)$

An **upper bound** on the true second-highest probability $p_B(x)$

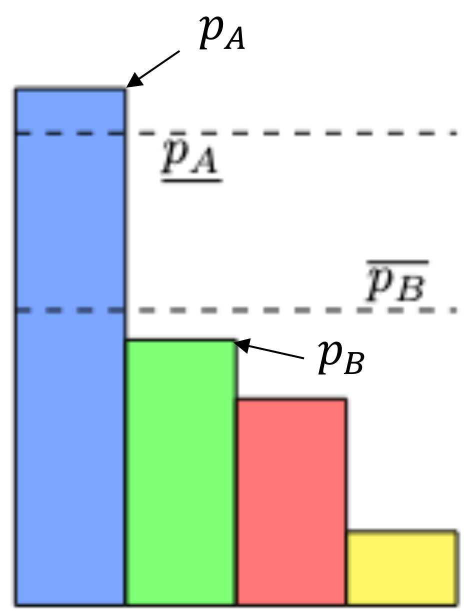
In theory, we could potentially compute the true exact probabilities p_A, p_B using for instance exact probabilistic inference solvers such as PSI [https://github.com/eth-sri/psi]. However, exact inference solvers do not scale to realistic networks and we will approximate the probabilities (with certain statistical guarantees).

Constructing classifier g: illustration with lower/upper bounds

$f(x+\epsilon)$



$g(x)$

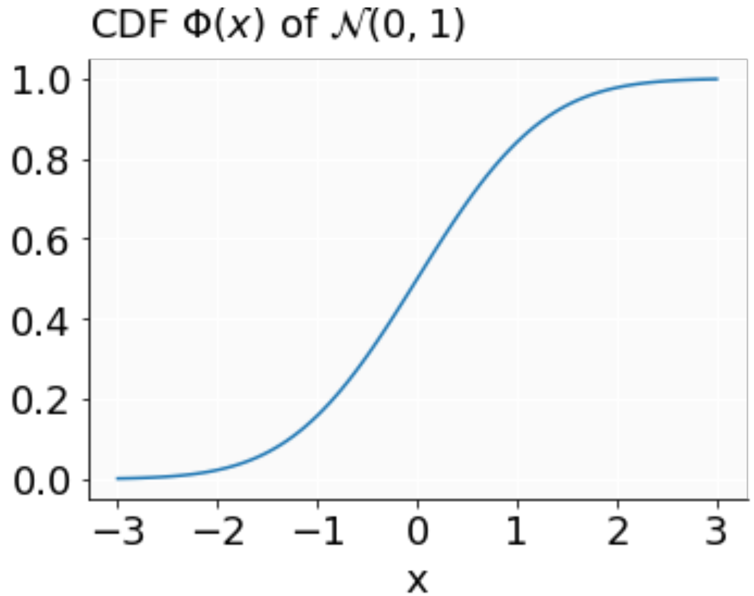


*Dashed lines are NOT radius!!

Robustness Guarantee: intuition

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

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



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 $\underline{p_A} \geq \overline{p_B}$

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

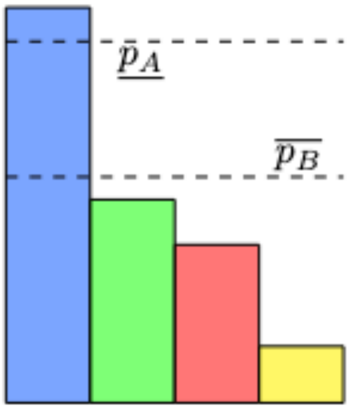
Robustness Guarantee: intuition

For a given σ in $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$, to increase radius R, we want higher p_A and lower p_B. Thus, while not theoretically required it is practically important to train f on noisy samples $x + \epsilon$. Increasing σ, and thus the level of noise, will reduce classifier accuracy.

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

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.



Note: Increasing the gap between $\underline{p_A}$ and $\overline{p_B}$ increases R.

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.

Reminder
Theorem: Robustness Guarantee

Suppose that: $c_A \in \mathcal{Y}$ and $\underline{p_{A,x}}$, $\overline{p_{B,x}} \in [0,1]$ satisfy:

$$\mathbb{P}_\epsilon(f(x+\epsilon) = c_A) =: p_A(x) \geq \underline{p_{A,x}} \geq \overline{p_{B,x}} \geq \max_{c \neq c_A} \mathbb{P}_\epsilon(f(x+\epsilon) = c)$$

Then:

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

certification radius $R_x := \frac{\sigma}{2}(\Phi^{-1}(\underline{p_{A,x}}) - \Phi^{-1}(\overline{p_{B,x}}))$ for sample x

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

Robustness Guarantee: Proof

Proofs via multiple mathematical persepctives:

- In the exercise: via Neymann-Pearson-Lemma.
- Last slide: via Lipschitzness.
- Others exist, e.g. optimization, information theory.

Proof sketch on last slide, different proof in exercise

Key challenge:

To compute **certified accuracy** of g, we need to get the probabilities p_A and p_B or their bounded versions **soundly** and **efficiently**. However, doing so analytically is not possible due to inherent costs.

Analytical Solution possible in some settings (so, certification and inference steps are the same):
(De-)Randomized Smoothing for Decision Stump Ensembles, NeurIPS'2022
Horváth, Müller, Fischer, Vechev https://www.sri.inf.ethz.ch/publications/horvath2022derand

Efficient Certification

Assumption: Assume $\underline{p_A} > \frac{1}{2}$ and let $\overline{p_B} = 1 - \underline{p_A}$.

We observe, that $\overline{p_B} \leq \frac{1}{2}$ and therefore $\underline{p_A} \geq \overline{p_B}$.

To get the radius:

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

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

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

Assumption above enables **efficient** certification: Now only $\underline{p_A}$ must be obtained, rather than **all** $\overline{p_{C \neq A}}$ to find $\overline{p_B}$.

Certification Procedure

given x determine the output class $\widehat{c_A}$ and certification radius R

```
function CERTIFY(f, σ, x, n₀, n, α):
  ĉ_A ← guess_top_class(f, σ, x, n₀)
  p_A ← lower_bound_p(ĉ_A, f, σ, x, n, α)
  if p_A > ½:
    R ← σΦ⁻¹(p_A)
    return ĉ_A, R
  else:
    return ABSTAIN
```

Certification Procedure

```
function CERTIFY(f, σ, x, n0, n, α):
 ĉA ← guess_top_class(f, σ, x, n0)
 pA ← lower_bound_p(ĉA, f, σ, x, n, α)
 if pA > ½:
  R ← σΦ^{-1}(pA)
  return ĉA, R
 else:
  return ABSTAIN
```

use n_0 samples of $f(x+\epsilon)$ to make a guess at the top class $\hat{c}_A$

Certification Procedure

```
function CERTIFY(f, σ, x, n₀, n, α):
 ĉ_A ← guess_top_class(f, σ, x, n₀)
 p_A ← lower_bound_p(ĉ_A, f, σ, x, n, α)
 if p_A > ½:
  R ← σΦ⁻¹(p_A)
  return ĉ_A, R
 else:
  return ABSTAIN
```

lower bound the probability for class $\hat{c}_A$

Monte Carlo Integration

$$p_A(x) = \mathbb{P}_\epsilon(f(x+\epsilon) = c_A) = \int_\epsilon [f(x+\epsilon) = c_A]\, \mu_{\mathcal{N}(0,\sigma^2\mathbf{1})}(\epsilon)\, d\epsilon$$

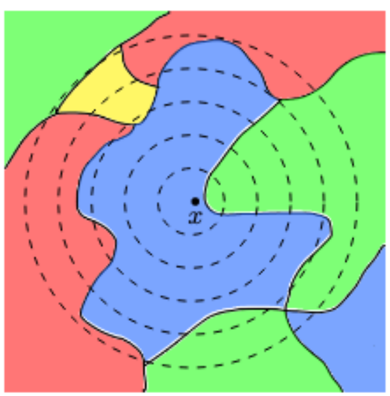
$\downarrow$ Monte Carlo Integration

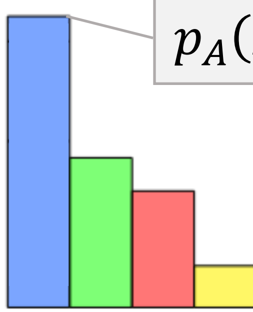
$$\approx \frac{1}{n}\sum_{i=1}^{n}[f(x+\epsilon_i) = c_A] =: \widehat{p_A}$$

for samples $\epsilon_1, \ldots, \epsilon_n \sim \mathcal{N}(0, \sigma^2\mathbf{1})$

$\mu_{\mathcal{N}(0,\sigma^2\mathbf{1})}(\cdot)$ denotes the probability density function of $\mathcal{N}(0, \sigma^2\mathbf{1})$

we don't know whether $\widehat{p_A}$ is smaller or larger than $p_A(x)$





$$p_A(x) = \mathbb{P}_\epsilon(f(x+\epsilon = \square)$$

The integral computes the probability by integrating over the blue region – the region is captured by the indicator function.

Monte Carlo Integration Bounds

$$p_A(x) = \mathbb{P}_\epsilon(f(x+\epsilon) = c_A) = \int_\epsilon [f(x+\epsilon) = c_A]\, \mu_{\mathcal{N}(0,\sigma^2\mathbf{1})}(\epsilon)\, d\epsilon$$

Monte Carlo Integration

$$\approx \frac{1}{n}\sum_{i=1}^{n}[f(x+\epsilon_i) = c_A] =: \widehat{p_A}$$

we don't know whether $\widehat{p_A}$ is smaller or larger than $p_A(x)$

for samples $\epsilon_1, \ldots, \epsilon_n \sim \mathcal{N}(0, \sigma^2\mathbf{1})$

Statistical Bound

find $\underline{p_A}$ and $\overline{p_A}$ such that

$$\mathbb{P}(\underline{p_A} \leq p_A \leq \overline{p_A}) \geq 1 - \alpha.$$

Several methods for this exist:

- Chebyshev's inequality
- Central Limit Theorem
- Binomial confidence bound (Clopper-Pearson intervals, next)

Usually such bounds take the form $\widehat{p_A} \pm b$ for some b depending on the sample variance n and α.

Binomial Proportion confidence bound

In our case $\mathrm{p}_A = p_A(x) = \mathbb{P}_\epsilon(f(x+\epsilon = c_A)$ is the true probability we want to estimate, but it is also the chance that for a sample $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$ the expression $[f(x+\epsilon = c_A)]$ evaluates to 1, i.e., $[f(x+\epsilon = c_A)] \sim Ber(p_A)$.

Thus for n samples we expect $n_A \sim \text{Binomial}(n, p_A)$ blue samples.

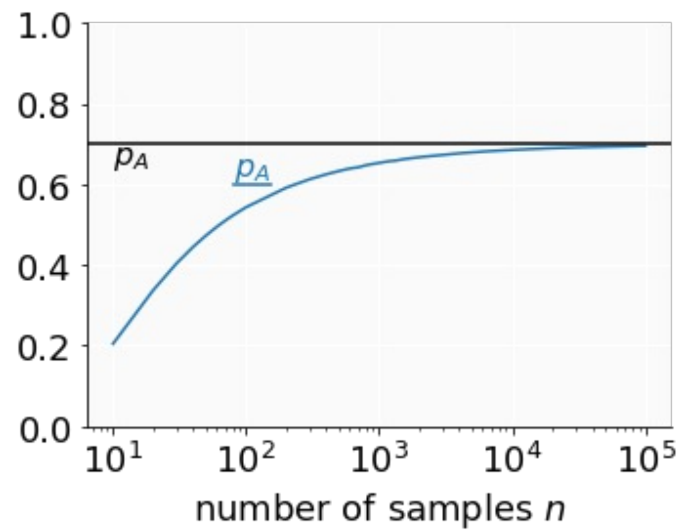
$$\widehat{p_A} \coloneqq \frac{1}{n} \underbrace{\sum_{i=1}^{n} [f(x+\epsilon_i) = c_A]}_{n_A} = \frac{n_A}{n}$$

A binomial confidence bound determines probability $\underline{p_A}, \overline{p_A}$, such that

$$\mathbb{P}(\underline{p_A} \leq p_A \leq \overline{p_A}) \geq 1 - \alpha$$

for the unknown success p_A.

There are many ways to compute this confidence interval, e.g. Clopper-Pearson (on the right).



$p_A = 0.7$ and $\alpha = 0.01$, i.e., 1 per cent failure probability using the lower bound from the Clopper Pearson confidence interval.

Certification Procedure

```
function CERTIFY(f, σ, x, n_0, n, α):
 ĉ_A ← guess_top_class(f, σ, x, n_0)
 p_A ← lower_bound_p(ĉ_A, f, σ, x, n, α)
 if p_A > ½:
  R ← σΦ^{-1}(p_A)
  return ĉ_A, R
 else:
  return ABSTAIN
```

given $\underline{p_A}$ we compute the radius R if the side-condition $\underline{p_A} > \frac{1}{2}$

Certification Procedure

```
function CERTIFY(f, σ, x, n_0, n, α):
  ĉ_A ← guess_top_class(f, σ, x, n_0)
  p_A ← lower_bound_p(ĉ_A, f, σ, x, n, α)
  if p_A > ½:
    R ← σΦ^{-1}(p_A)
    return ĉ_A, R
  else:
    return ABSTAIN
```

We sample twice, first with n_0, then $n >> n_0$ samples to prevent selection bias when estimating $\underline{p_A}$

Certification Procedure

```
function CERTIFY(f, σ, x, n_0, n, α):
 ĉ_A ← guess_top_class(f, σ, x, n_0)
 p_A ← lower_bound_p(ĉ_A, f, σ, x, n, α)
 if p_A > ½:
  R ← σΦ^{-1}(p_A)
  return ĉ_A, R
 else:
  return ABSTAIN
```

If $\widehat{c_A}$, R is returned by CERTIFY, then by the theorem, with probability of at least $1-\alpha$, $g(x) = g(x+\delta) = \widehat{c_A}$ for all δ with $\|\delta\|_2 \leq R$.

Certification Procedure

```
function CERTIFY(f, σ, x, n0, n, α):
 ĉA ← guess_top_class(f, σ, x, n0)
 pA ← lower_bound_p(ĉA, f, σ, x, n, α)
 if pA > ½:
  R ← σΦ^{-1}(pA)
  return ĉA, R
 else:
  return ABSTAIN
```

If CERTIFY returns ABSTAIN, then there are 3 possibilities:

1. our guess of $\widehat{c_A}$ was wrong, and we thus estimate the corresponding p_A under 0.5 (can be remedied by increasing n_0),

2. certification is not efficiently possible, i.e. the true $p_A \leq 0.5$ (can be remedied by training with noise to bias it toward the target level) or

3. the true $p_A > 0.5$ but, the lower bound is too lose (can be remedied by increasing n).

Certification Procedure

function CERTIFY(f, σ, x, n_0, n, α):
 $\widehat{c_A}$ ← guess_top_class(f, σ, x, n_0)
 $\underline{p_A}$ ← lower_bound_p($\widehat{c_A}$, f, σ, x, n, α)
 if $\underline{p_A}$ > ½:
  $\mathrm{R} \leftarrow \sigma\Phi^{-1}(\underline{p_A})$
  return $\widehat{c_A}$, R
 else:
  return ABSTAIN

As Φ^{-1} is monotone, increasing $\underline{p_A}$ will increase the radius. To increase $\underline{p_A}$ we need to get the base classifier f to classify more points as $\widehat{c_A}$.

Effect of noise σ on Certified Robustness vs. 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. Using $n = 100000$ samples and $\alpha = 0.001$ for certification, inference (discussed next) typically uses 100 to 1000 samples. ACR is the average certified Radius over correctly classified images. The table compares different methods for training the base classifier f (Gaussian is training with added Noise). The methods present trade-offs over lower and higher radii r.

σ	Training Method	ACR	r=0.0	r=0.5	r=1.0	r=1.5	r=2.0	r=2.5	r=3.0	r=3.5
0.50	Gaussian	0.733	**57**	46	37	29	0	0	0	0
	Consistency	0.822	55	**50**	**44**	34	0	0	0	0
	SmoothAdv	0.825	54	49	43	37	0	0	0	0
	SmoothMix	**0.846**	55	**50**	43	**38**	0	0	0	0
1.00	Gaussian	0.875	**44**	**38**	33	26	19	15	12	9
	Consistency	0.982	41	37	32	28	24	21	17	14
	SmoothAdv	1.040	40	37	**34**	**30**	**27**	**25**	**20**	15
	SmoothMix	**1.047**	40	37	**34**	**30**	26	24	**20**	**17**

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 and certified training.

Results from:
SmoothMix: Training Confidence-calibrated Smoothed Classifiers for Certified Robustness, Jeong et al., NeurIPS'2021

Once a classifier is certified on the test set (via sampling as discussed so far), we need to actually use this classifier at inference time, and again we resort to sampling (with statistical guarantees).

However, here we do not need to compute the radius anymore, we just need to find the top class, which leads to a cheaper procedure.

Inference Procedure I

```
function PREDICT1(f, σ, x, n, α):
 ĉ_A, n_A ← top_class(f, σ, x, n)
 if BinomPValue(n_A, n, <=, 0.5) ≤ α
   return ĉ_A
 else return ABSTAIN
```

Additional work is needed at inference time, which can be expensive for high number of samples, typically n is small though

`top_class` returns the top $\widehat{c_A}$ and the count n_A

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

`BinomPValue`$(i, n, \leq, q)$: 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 $q \leq 0.5$.

Inference Procedure II

function PREDICT2(f, σ, x, n, α):
  $\widehat{c_A}$, n_A, $\widehat{c_B}$, n_B ← top_two_classes(f, σ, x, n)
  if BinomPValue(n_A, $n_A + n_B$, =, 0.5) $\leq \alpha$
    return $\widehat{c_A}$
  else return ABSTAIN

More sample efficient.

Can return top class even if probability < ½.

top_two_classes returns the top $\widehat{c_A}, \widehat{c_B}$ and their counts n_A, n_B out of n samples.

In our case, the null hypothesis: the true probability of f returning $\widehat{c_A}$ is $q = 0.5$ (meaning the classes are indistinguishable).
BinomPValue$(i, n, =, q)$ returns the p-value for this case.

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

Inference Procedure

```
function PREDICT1(f, σ, x, n, α):
 ĉ_A, n_A ← top_class(f, σ, x, n)
 if BinomPValue(n_A, n, <=, 0.5) ≤ α
   return ĉ_A
 else return ABSTAIN

function PREDICT2(f, σ, x, n, α):
 ĉ_A, n_A, ĉ_B, n_B ← top_two_classes(f, σ, x, n)
 if BinomPValue(n_A, n_A + n_B, =, 0.5) ≤ α
   return ĉ_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.

We can prove that: both return $\widehat{c_A} \neq c_A$ with probability at most α

Extensions to Randomized Smoothing

Generalization of properties

Geometric Perturbations: Certified Defense to Image Transformations via Randomized Smoothing
Fischer, Baader, Vechev; NeurIPS'2020 https://arxiv.org/abs/2002.12463

Convex Inputs; various norms: Randomized Smoothing of All Shapes and Sizes
Yang et al., ICML'2020 https://arxiv.org/abs/2002.08118

Specialization of models and distribution (allowing determinism)

Restrict model: (De-)Randomized Smoothing for Decision Stump Ensembles
Horváth, Müller, Fischer, Vechev; NeurIPS'2022 https://arxiv.org/abs/2205.13909

Restrict perturbation: Improved, Deterministic Smoothing for l1 Certified Robustness
Alexander Levine, Soheil Feizi; ICML'2021 https://arxiv.org/abs/2103.10834

Combining with Fully Homomorphic Encryption

Private and Reliable Neural Network Inference, Jovanović, Fischer, Steffen, Vechev
CCS'2022 **(next week)** https://files.sri.inf.ethz.ch/website/papers/ccs22-phoenix.pdf

Better Base models

Use ensembles: Boosting Randomized Smoothing with Variance Reduced Classifiers
Horváth, Müller, Fischer, Vechev; ICLR'2022 (Spotlight) https://arxiv.org/abs/2106.06946

Use SOTA diffusion models: (Certified!!) Adversarial Robustness for Free!
Carlini et al.; arXiv'2022 https://arxiv.org/abs/2206.10550

Better training: SmoothMix: Training Confidence-calibrated Smoothed Classifiers for Certified Robustness
Jeong et al.; NeurIPS'2021 https://arxiv.org/abs/2111.09277

Summary of Randomized Smoothing

- We introduced randomized smoothing, a method which constructs robust classifiers by introducing Gaussian noise which induces a robustness radius. A benefit of smoothing is that it scales to large networks.

- Smoothing relaxes the standard deterministic guarantees into statistical guarantees on the robustness of the classifier.

- To obtain higher certified radius, one may need many samples. It also requires sampling at inference time which convex methods do not. The classic trade-off of accuracy vs. robustness is also present here and is controlled by the amount of noise.

- Both finding specific instantiations of smoothing and generalizing it is an active area of research.

Certification: comparing methods

	Robustness Certificate	Adaption to new model class f	Adaption to new Specification	Suitable for neural network scale
Deterministic Vertification e.g. DeepPoly	Through sound analysis	Requries new transfromers	Encode Perturbation as convex region	small to mid size
Randomized Smoothing	By construction	Model Agnostic	Requires new mathematical insights	All sizes, but added latency might be prohibitive when small nets are used

Summary of Part 1: Certification

Robustness	Privacy	Fairness/Bias
attacks and defenses, certification (relaxations, branch and bound, certified training, smoothing)	attacks, differential privacy, secure synthetic data, data minimization, federated learning vulnerabilities	individual fairness, group fairness, methods for building fair systems for tabular, NLP and visual data

Lecture appendix: Robustness Guarantee: Proof Sketch

Def: A function $h: R^m \mapsto [0,1]$ is ***K*-Lipschitz** if $|h(x_1) - h(x_2)| \leq K\|x_1 - x_2\|_2$.
Lemma: For some function $h: R^m \mapsto [0,1]$,
$\Phi^{-1}(\mathbb{E}_{\epsilon\sim\mathcal{N}(0,1)}[h(x+\epsilon)])$ is 1-Lipschitz in x.

probability of indicator function is an expectation

Assume the top class is c_A and $p_A(x) \coloneqq \mathbb{P}_\epsilon(f(x+\epsilon) = c_A) = \mathbb{E}_\epsilon([f(x+\epsilon) = c_A])$ and $p_A(x) > p_B(x)$. An adversary picks δ to flip classification, i.e. $p_A(x+\delta) \leq p_B(x+\delta)$.

By Lipschitzness: $\left| - \Phi^{-1}\big(p_A(x+\delta)\big)\right| \leq \|\delta\|_2$. This shows $\Phi^{-1}\big(p_A(x)\big) - \Phi^{-1}\big(p_A(x+\delta)\big)$ and when plugging in the adversary's goal yields $\Phi^{-1}\big(p_A(x)\big) - \Phi^{-1}\big(p_B(x+\delta)\big) \leq \|\delta\|_2$.

Applying the same reasoning to p_B, we arrive at $\Phi^{-1}\big(p_B(x+\delta)\big) - \Phi^{-1}\big(p_B(x)\big) \leq \|\delta\|_2$.

Adding these inequalities yields $\frac{1}{2}\Big(\Phi^{-1}\big(p_A(x)\big) - \Phi^{-1}\big(p_B(x)\big)\Big) \leq \|\delta\|_2$, which tells us that due to the Lipschitzness of $\Phi^{-1}\big(p_A(\cdot)\big)$ and $\Phi^{-1}\big(p_B(\cdot)\big)$ $\|\delta\|_2$ needs to be at least the LHS above.

By the monotonicity of $\Phi^{-1}(\cdot)$ we can also plug in p_A and p_B and recover the original theorem. This shows the case for $\sigma = 1.0$, can be extended to arbitrary σ by extending the above Lemma.

Provably Robust Deep Learning via Adversarially Trained Smoothed Classifier, NeurIPS 2019
Salman et al. https://arxiv.org/pdf/1906.04584.pdf