Foundations of Deep Learning

Lecture 10: Provable and Generalizable Adversarial Defenses

Soheil Feizi

Course Webpage:
http://www.cs.umd.edu/class/fall2020/cmsc828W/

@FeiziSoheil

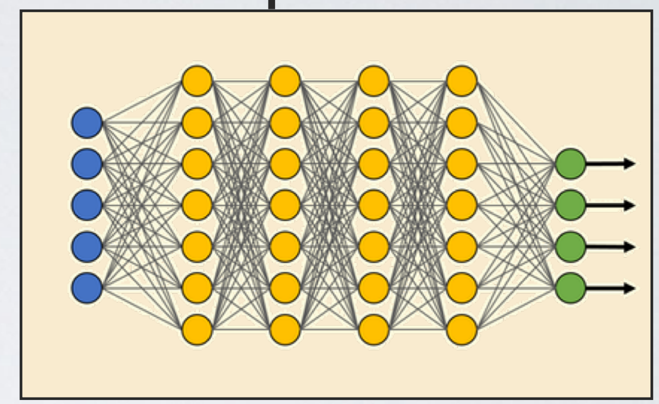
COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Deep Learning Pipeline

Training data

Optimization

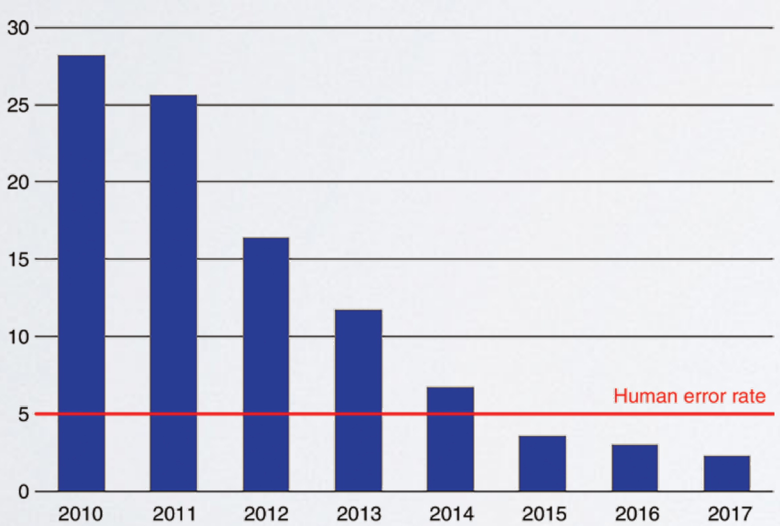
Deep model



Test data

Evaluation

Classification error



Robustness against inference time adversarial attacks

Adversarial Examples

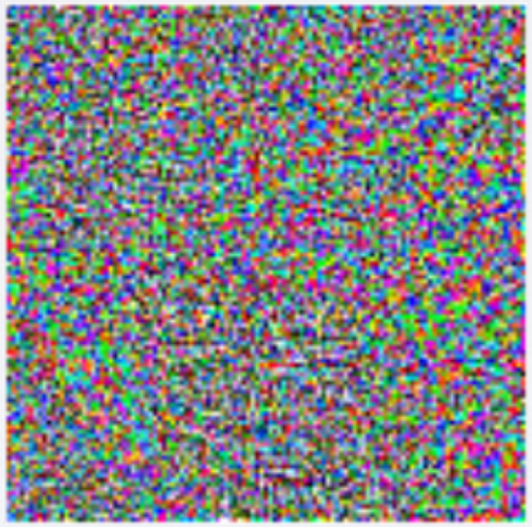
- $\mathbf{x}'$ is an adversarial examples for a ML classifier $f_{\mathrm{ML}}(\cdot)$ if

$$f_{\mathrm{ML}}(\mathbf{x}) \neq f_{\mathrm{ML}}(\mathbf{x}') \quad \text{and} \quad f_{\mathrm{human}}(\mathbf{x}) = f_{\mathrm{human}}(\mathbf{x}')$$

"Egyptian Cat"

$\mathbf{x}$

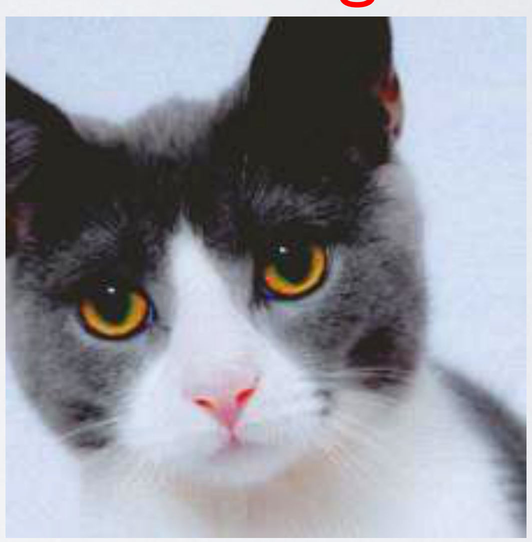
$+$



δ

$=$

"Traffic Light"



$\mathbf{x}'$

Challenge: Lack of a mathematical characterization of human perception

Adversarial Attack Problem

- Goal: create adversarial examples to mislead a classifier $f(.)$

$$\max_{\mathbf{x}'} \ \ell_{cls}(f(\mathbf{x}'), y)$$
$$\mathbf{x}' \in \mathcal{T}(\mathbf{x}, \rho)$$

threat model

- Often leads to non-convex opt → Solve using Projected Gradient Descent (Madry et al.' 17)

- Threat model:
  - L_p attacks:

$$\mathcal{T}(\mathbf{x}, \rho) = \{\mathbf{x}' : \|\mathbf{x} - \mathbf{x}'\|_p \leq \rho\}$$

Robustness against L_p attacks is necessary but not sufficient

  - Non-L_p attacks:
    - Spatial attacks (Wasserstein attacks, Wong et al.'19)
    - Semantic-level attacks (RecolorAdv, Laidlaw, F. NeurIPS'19)

Sparse Adversarial Attacks

- Adversary can change up to ρ pixels

Input Image

Classification label: 3

$\rho = 25$ →

Adv Example

Classification label: 5

Wasserstein Adversarial Attacks

- Introduced by Wong et al.'19
- Adversarial perturbation is measured by Wasserstein distance on normalized images

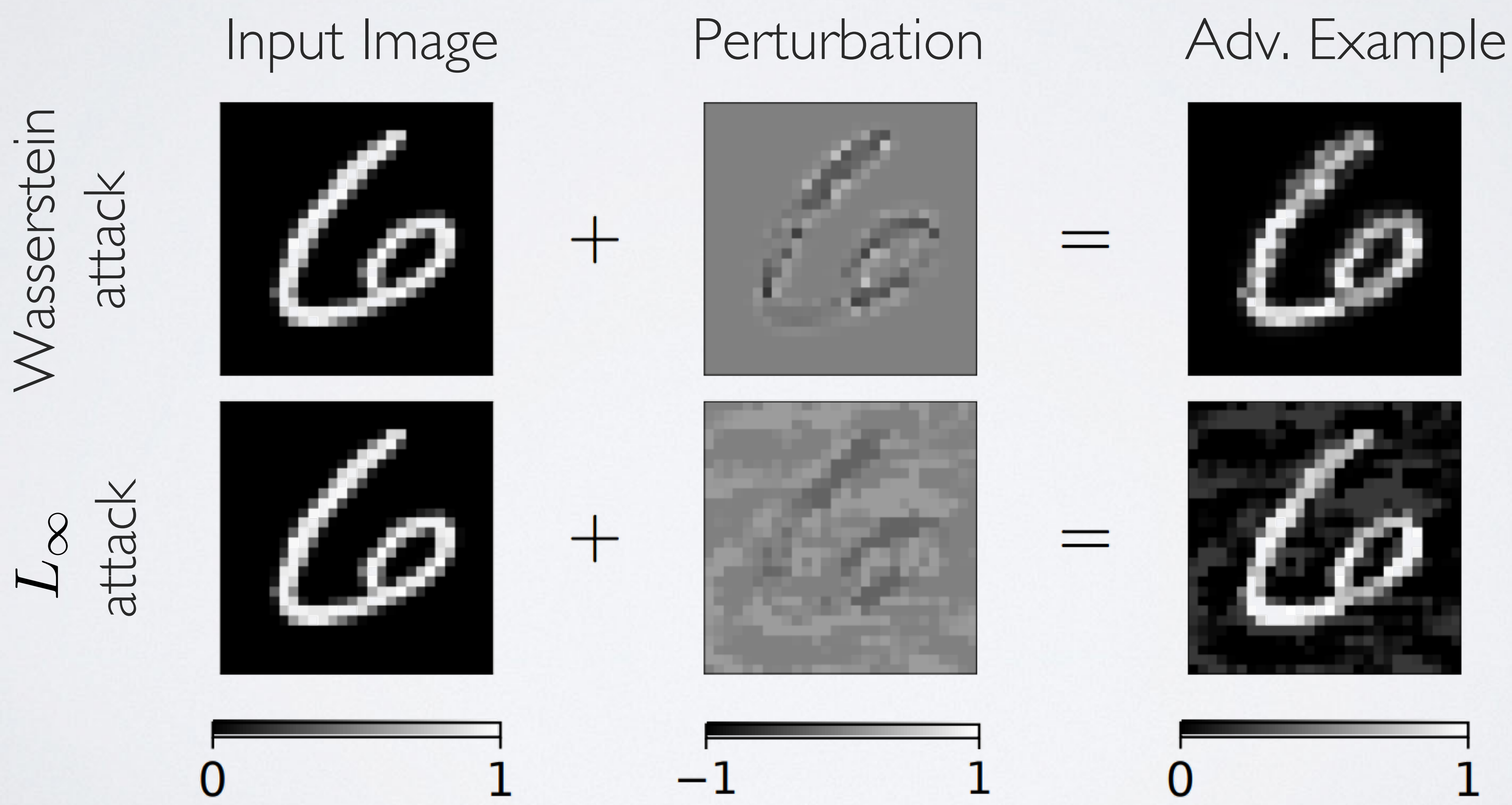
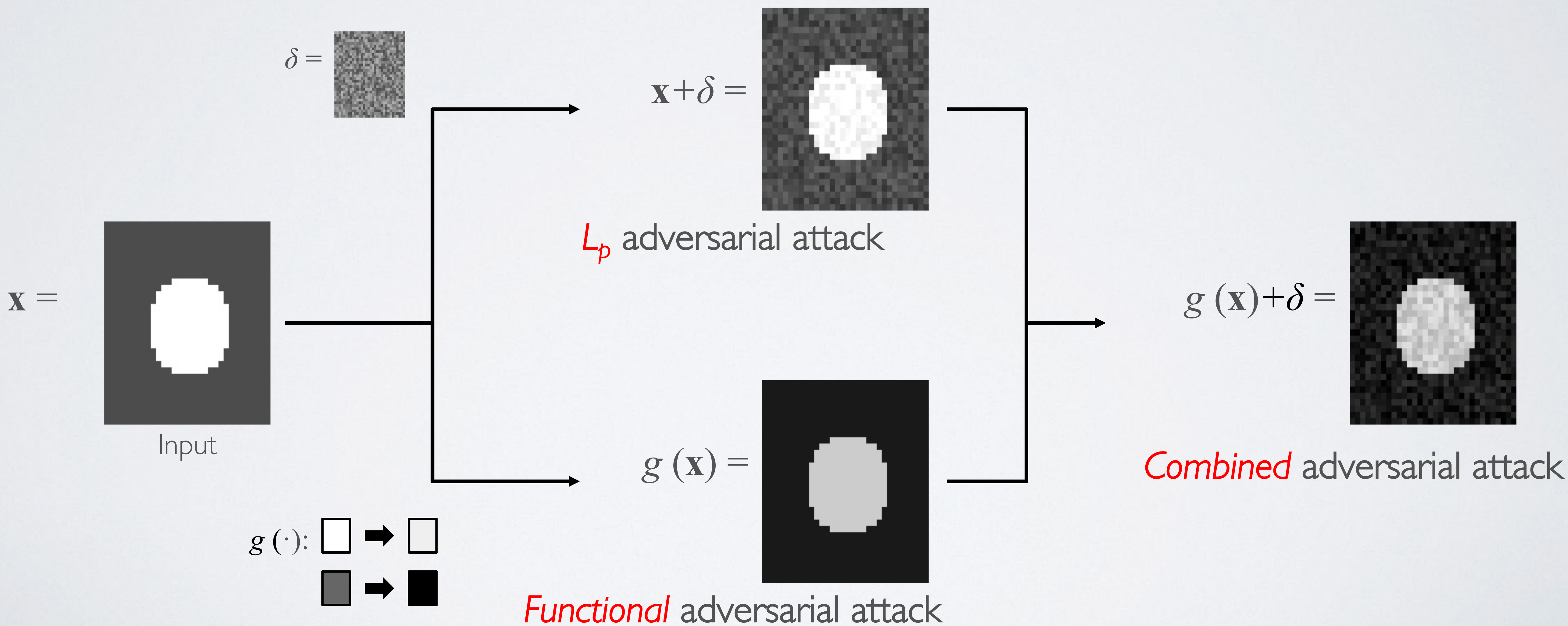


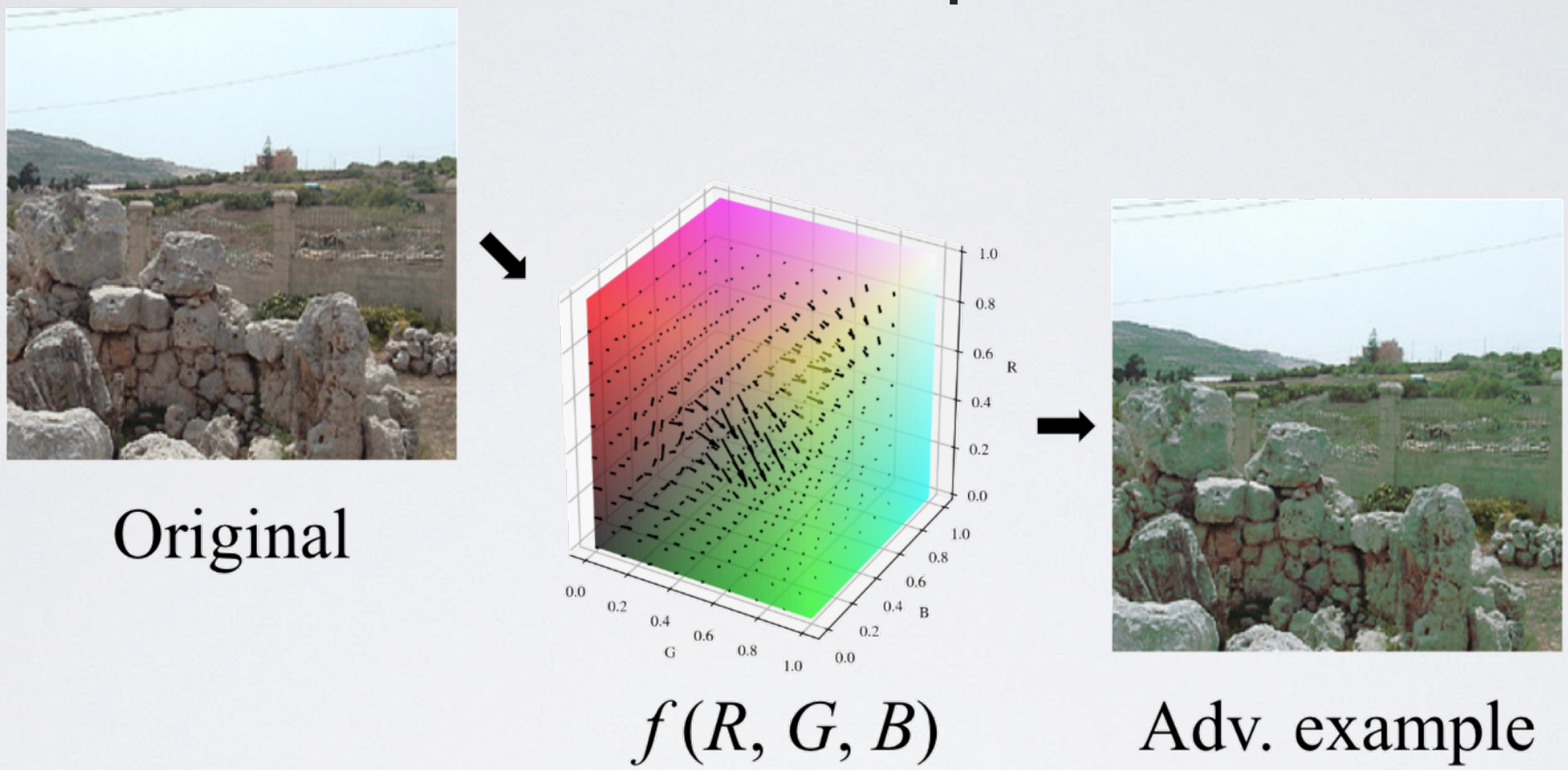
fig. from Wong et al.'19

Functional Adversarial Attacks

- Introduced by Laidlaw & F., NeurIPS'19
- Adversarial perturbation is a function of input features



RecolorAdv: Functional Attacks in Color Space



	Attack				
Defense	**C**	**C + D**	**C + S**	**S + D**	**C + S + D**
None	**3.3%**	**0.0%**	**0.9%**	**0.0%**	**0.0%**
Adv. training	**45.8%**	**5.2%**	**8.7%**	**7.6%**	**3.6%**
TRADES	**59.2%**	**22.0%**	**17.5%**	**8.7%**	**5.7%**

Accuracy under attack on CIFAR-10. C is Functional attack, D is additive (ℓ_∞) attack with ε=8/255, S is StAdv attack (Xiao et al., 2018)


Attack
Defense

Defenses against Adversarial Attacks

- Standard ERM training:

$$\min_{\theta} \ \mathbb{E}_{(\mathbf{x},y)} \left[\ell_{cls} \left(f_\theta(\mathbf{x}), y\right)\right]$$

- Adversarial training (AT) for L_p attacks (Madry et al.'17):

$$\min_{\theta} \ \mathbb{E}_{(\mathbf{x},y)} \left[\max_{\delta} \ell_{cls} \left(f_\theta(\mathbf{x}+\delta), y\right)\right]$$

$$\delta \in \mathbf{\Delta} := \{\delta \in \mathbb{R}^n : \|\delta\|_p \leq \rho\}$$

- Solve using alternative SGD+PGD

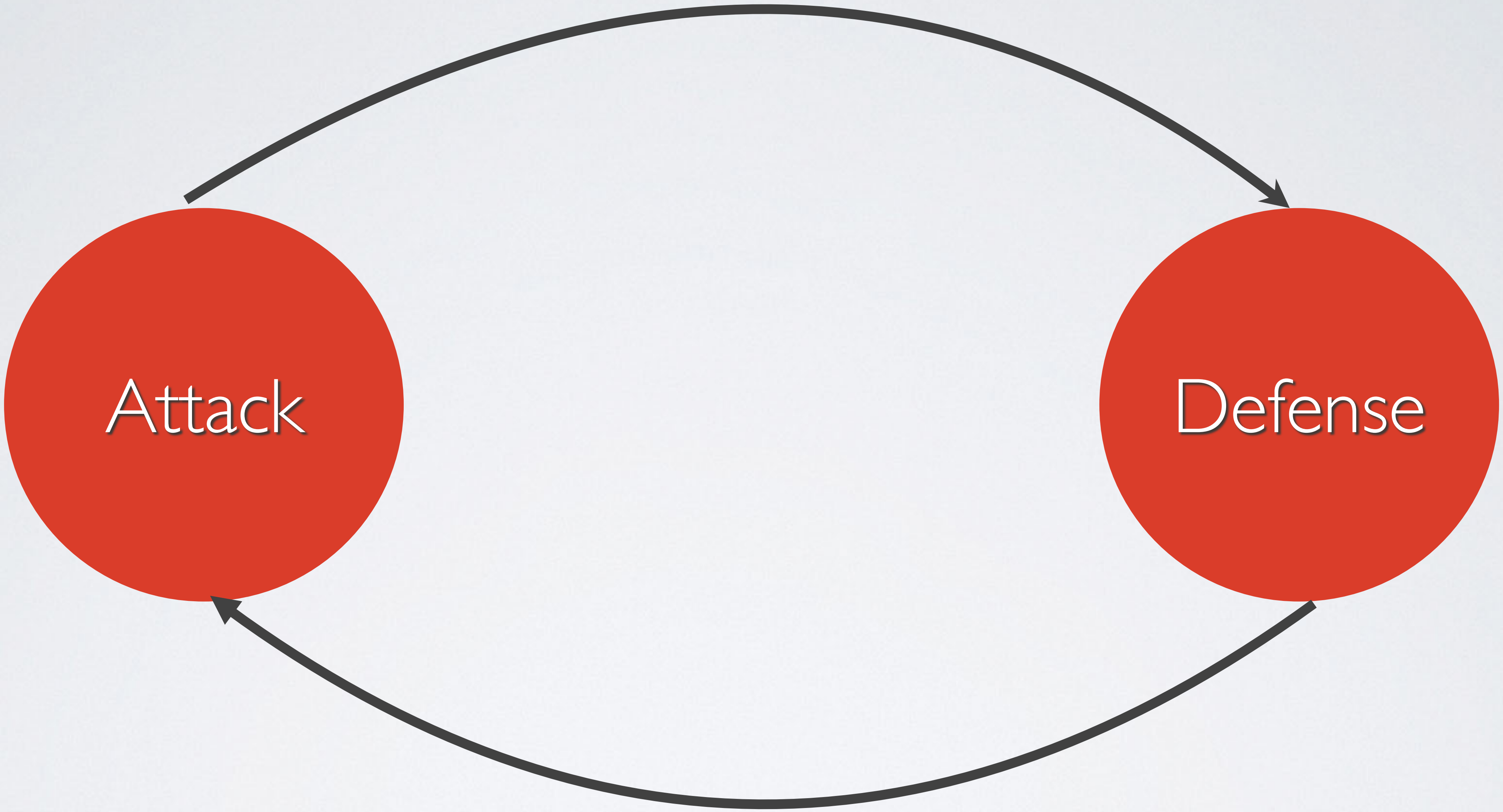
Several Heuristic Defenses

- New defenses introduced in ICLR 2018

Defense	Dataset	Distance
Buckman et al. (2018)	CIFAR	0.031 (ℓ_∞)
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)

Attack = (algorithm, threat model)

variable — fixed



Several Heuristic Defenses

- New defenses introduced in ICLR 2018

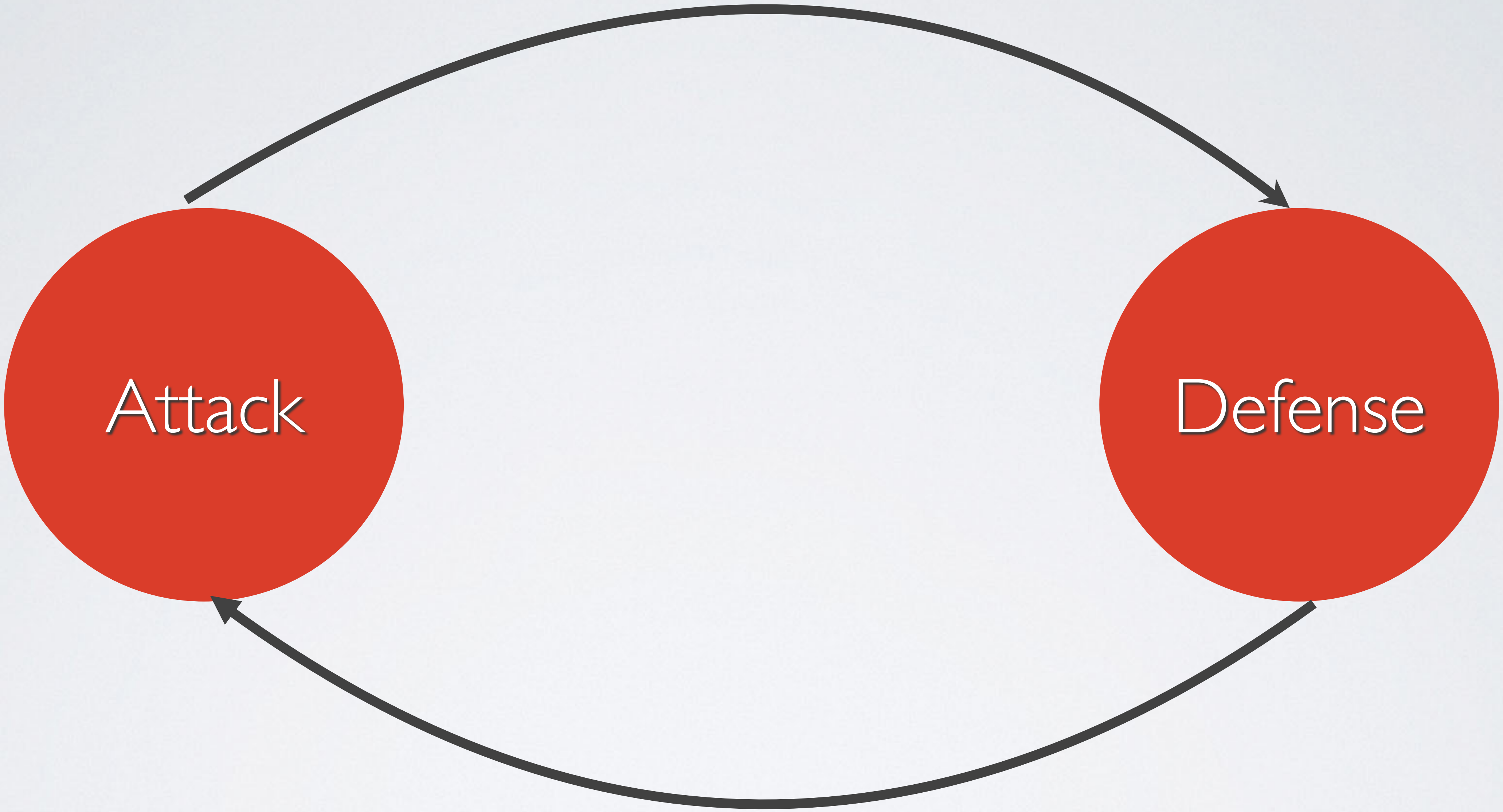
Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)	~~55%**~~ 0% (Ilyas et al. 2019)
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	15%

Empirical defenses are vulnerable against adaptive attacks (within the same threat model)

Athalye et al. ICML 2019

Attack = (algorithm, threat model)

variable variable



Generalization to Unforeseen Attacks

- Attackers may *not* obey the threat model used in the defense
- Standard defenses have poor generalization to unforeseen attacks (Kang et al. 2018)
- Unforeseen Attack Robustness of AT-based defenses on CIFAR-10

AT-based defenses show poor generalization against unforeseen attacks (the ones not used in training)

Normal	**95.2**	0.0	0.0	0.0	0.6
AT L_∞	87.0	**52.4**	25.1	6.3	59.7
AT L_2	81.6	45.3	51.8	14.9	60.5
AT StAdv	83.9	0.3	0.8	**76.1**	13.9
AT ReColorAdv	92.0	15.5	10.5	0.3	**81.2**

Laidlaw, Singla, F. '20

Today's Lecture

- Part I: Attack = (algorithm, threat model)
  - algorithm: variable; threat model: fixed

- Part II: Attack = (algorithm, threat model)
  - algorithm: variable; threat model: variable

Certifiable/Provable Defenses

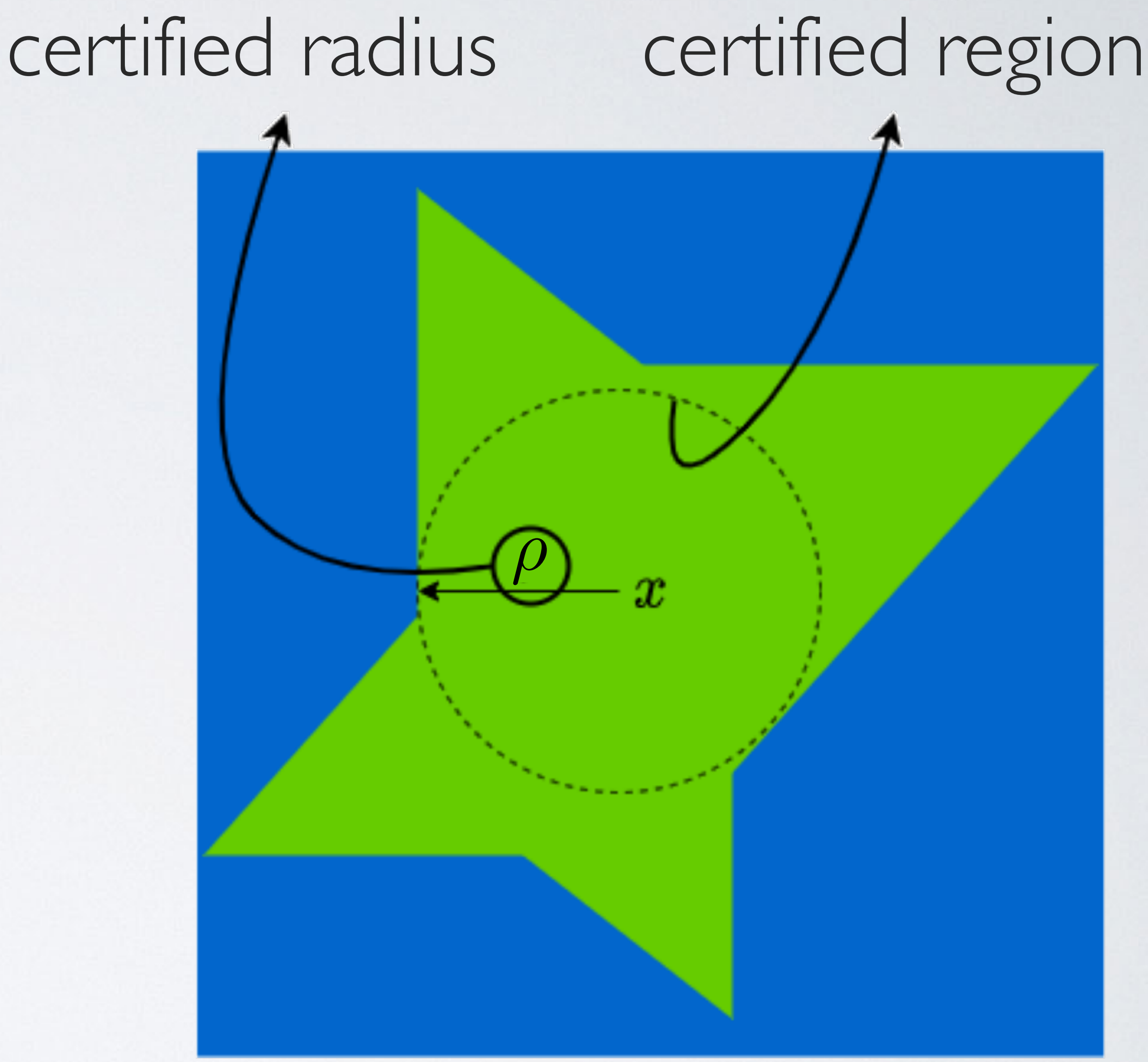
- A classifier f_θ is certifiably robust at $\mathbf{x}$ if for any

$$\mathbf{x}' \in \mathcal{T}(\mathbf{x}, \rho)$$

  we have:

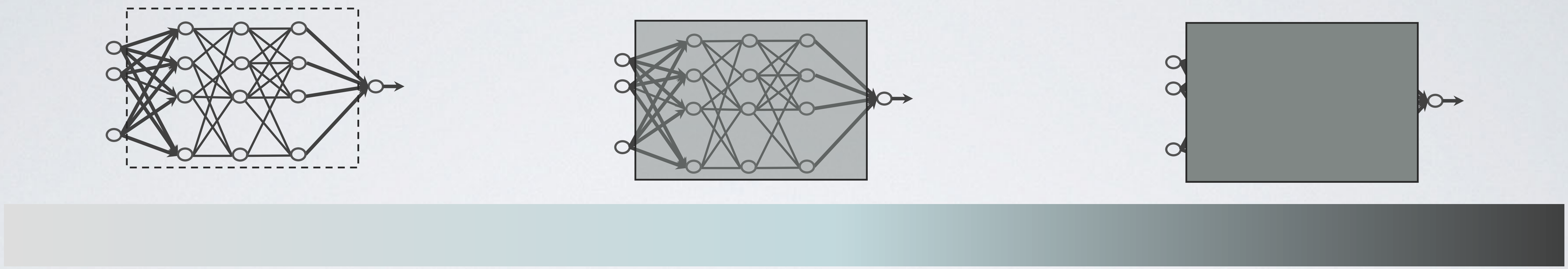
$$f_\theta(\mathbf{x}) = f_\theta(\mathbf{x}')$$

- ρ is the certification level



Landscape of Provable Defenses

Amount of the network information used in the defense



L_p:

Lipschitz/Curvature Bounds
Singla & F., ICML'20
Singla & F., ICML'21

IBP/Convex
Wong & Kolter, '18
Gowal, et al., '18, Mirman 2018, Zhang 2019

Randomized Smoothing
Cohen et al. '19, Li et al. '18, Salman et al. '19, Lecuyer et al. '19, Teng et al. '20, Lee et al. '19, Yang et al. '20, KLGF., ICML 20, KLFG, NeurIPS 20, Levine, F. ICML'21

Non-L_p:

Patch Threat
Chaing et al.'20

Sparse Threat
Lee et al. '19, Levine, F. AAAI'20

Wasserstein Threat
Levine, F. AISTATS '20

Patch Threat
Levine, F. NeurIPS'20, Xiang et al.'20

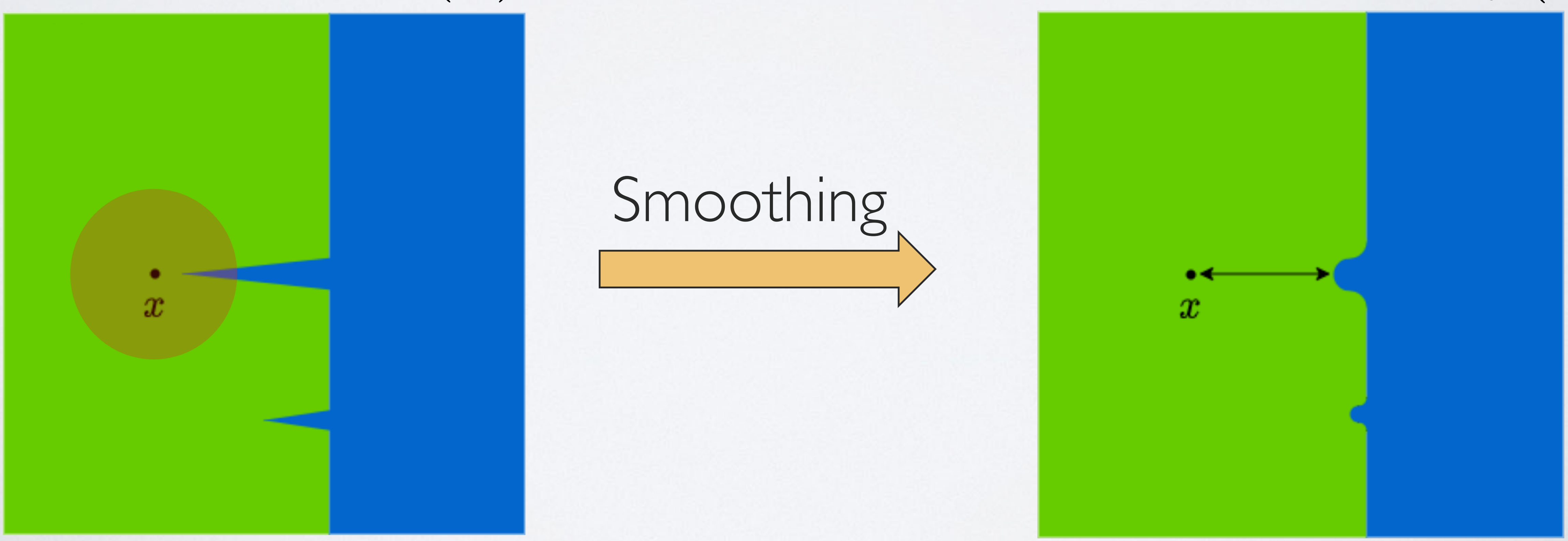
Randomized Smoothing

- A smoothed classifier:

$$\bar{f}(\mathbf{x}) := \mathbb{E}_{\epsilon}\left[f(\mathbf{x}+\epsilon)\right]$$

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

Base classifier $f(\mathbf{x})$ Smoothed classifier $\bar{f}(\mathbf{x})$



Gaussian Smoothing for L_2 attacks

- **Theorem** (Cohen et al.'19)

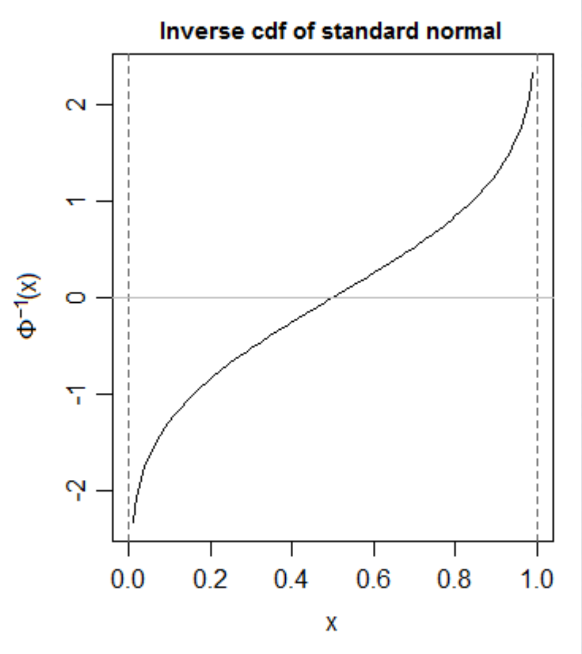
  No adv. example exists within the radius

$$\frac{\sigma}{2}\left(\Phi^{-1}\left(p_1(\mathbf{x})\right) - \Phi^{-1}\left(p_2(\mathbf{x})\right)\right)$$

majority class probability

runner-up class probability

$\Phi^{-1}(.)$

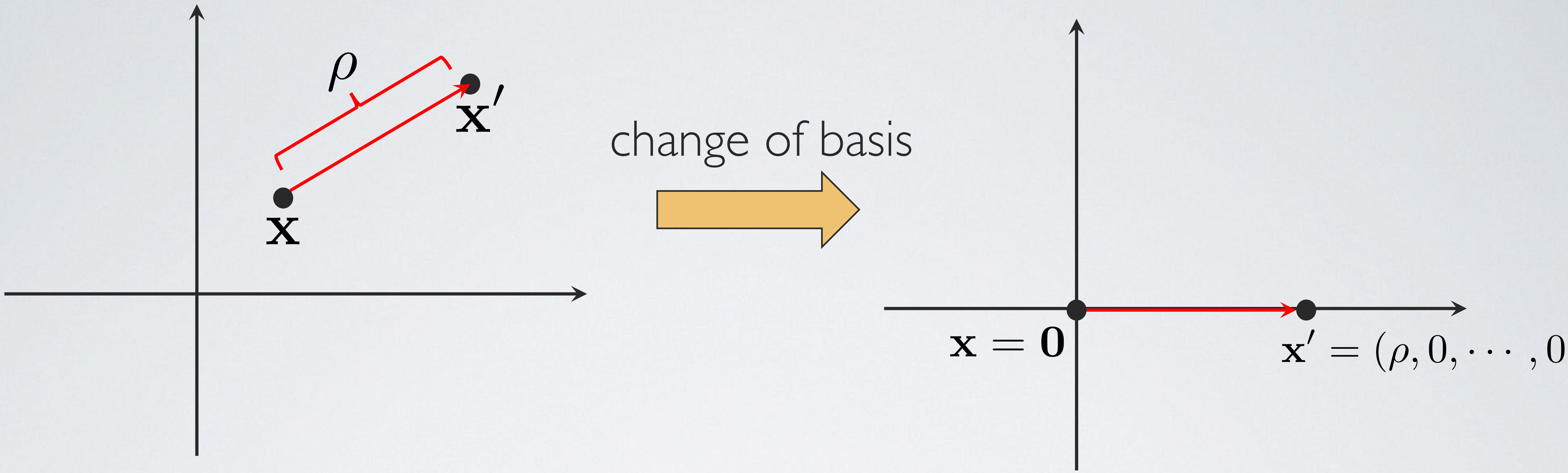


- Proof based on Neyman & Pearson lemma 1933
- Empirical bounds on probabilities
- **Theorem** (Levine, Singla, F.'19, Salman et al.'19)

  $\Phi^{-1}(\bar{f}(\mathbf{x}))$ is Lipschitz with constant $1/\sigma$

A simple one dimensional proof for Gaussian Smoothing

A Simple Proof for Gaussian Smoothing



- Define $g(z) := \mathbb{E}\left[f(z, \underbrace{\epsilon_2, \cdots, \epsilon_d}_{\text{smoothed out}})\right]$ $\quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$

Scalar function!

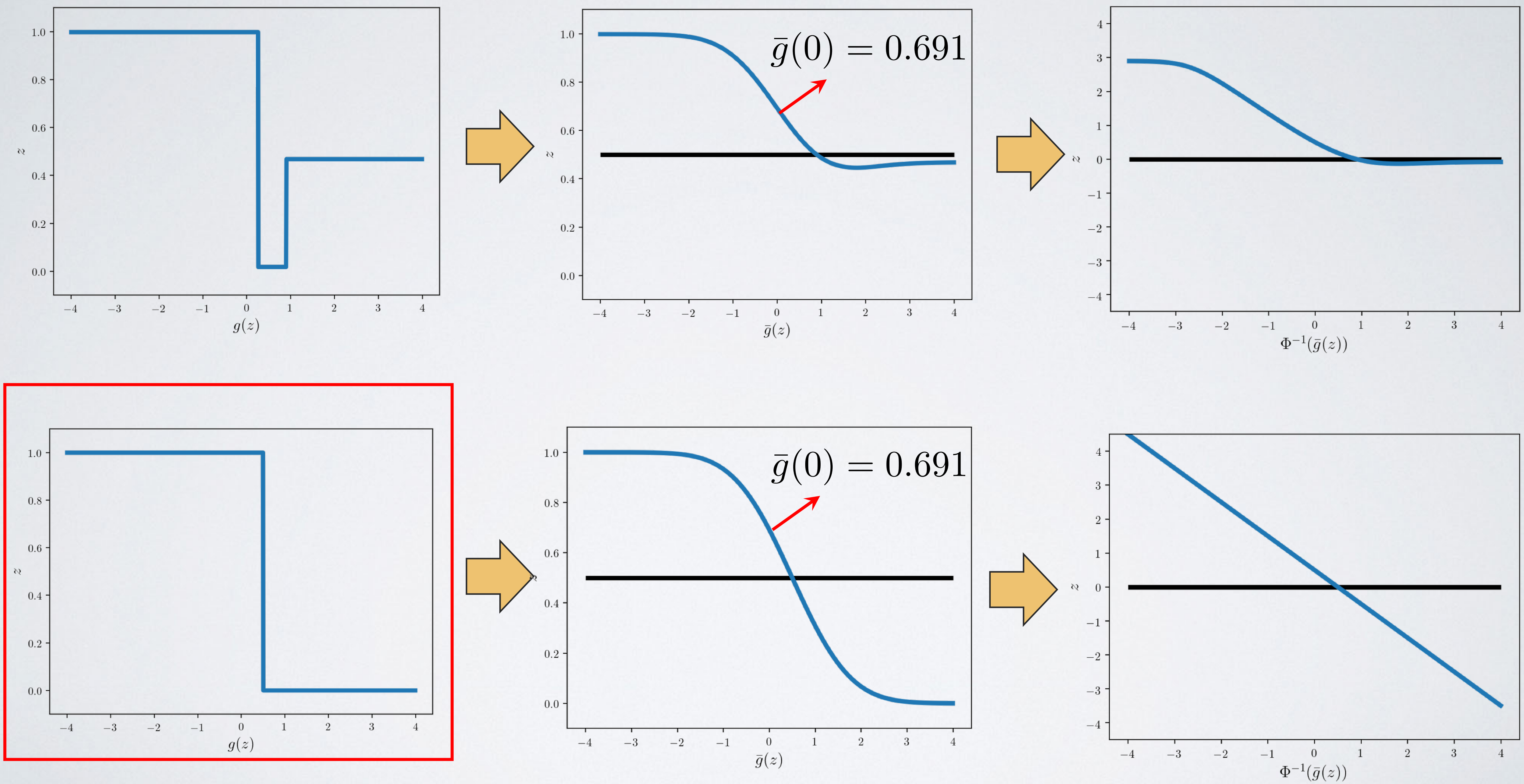
$$\bar{g}(z) := \mathbb{E}[g(z + \epsilon_1)]$$

- Need to show $\Phi^{-1} \circ \bar{g}$ is Lipschitz

Proof by Levine & F.

What is the worst g(.)?

$g(z)$ | $\bar{g}(z)$ | $\Phi^{-1} \circ \bar{g}$



What is the worst g(.)?

- Define $g_\Phi(y) := g(\sigma\Phi^{-1}(y))$
- Using straightforward one-dim calculus:

monotonically increasing

$$\bar{g}(\rho) \geq \min_{g_\Phi \in [0,1]\to[0,1]} \int_0^1 g_\Phi(y) e^{\Phi^{-1}(y) - \frac{\rho^2}{2\sigma^2}} dy$$

$$\text{s.t.} \quad \int_0^1 g_\Phi(y) dy = \bar{g}(0)$$

$$\Rightarrow \quad g^{\text{worst}}(z) = \begin{cases} 1 & \text{if } z \leq \sigma\Phi^{-1}(\bar{g}(0)) \\ 0 & \text{if } z > \sigma\Phi^{-1}(\bar{g}(0)) \end{cases}$$

Generalizability of Randomized Smoothing

- **Theorem** (KLGF. ICML'20)

  Using **any** symmetric i.i.d. smoothing:

$$r_p^* \leq \frac{\sigma}{2\sqrt{2}d^{\frac{1}{2}-\frac{1}{p}}}\left(\frac{1}{\sqrt{1-p_1(\mathbf{x})}}+\frac{1}{\sqrt{p_2(\mathbf{x})}}\right)$$

Robustness radius against L_p attacks

Extra dependence on **d** for **p>2**

- Curse of dimensionality: For L_p attacks where p>2, the smoothing-based certificate upper bound decreases as d increases

Curse of Dimensionality on Randomized Smoothing for Certifiable Robustness,
Kumar, Levine, Goldstein, F., ICML 2020

Gaussian Smoothing for L_p Attacks

- If we use Gaussian smoothing against L_p attacks, we get:

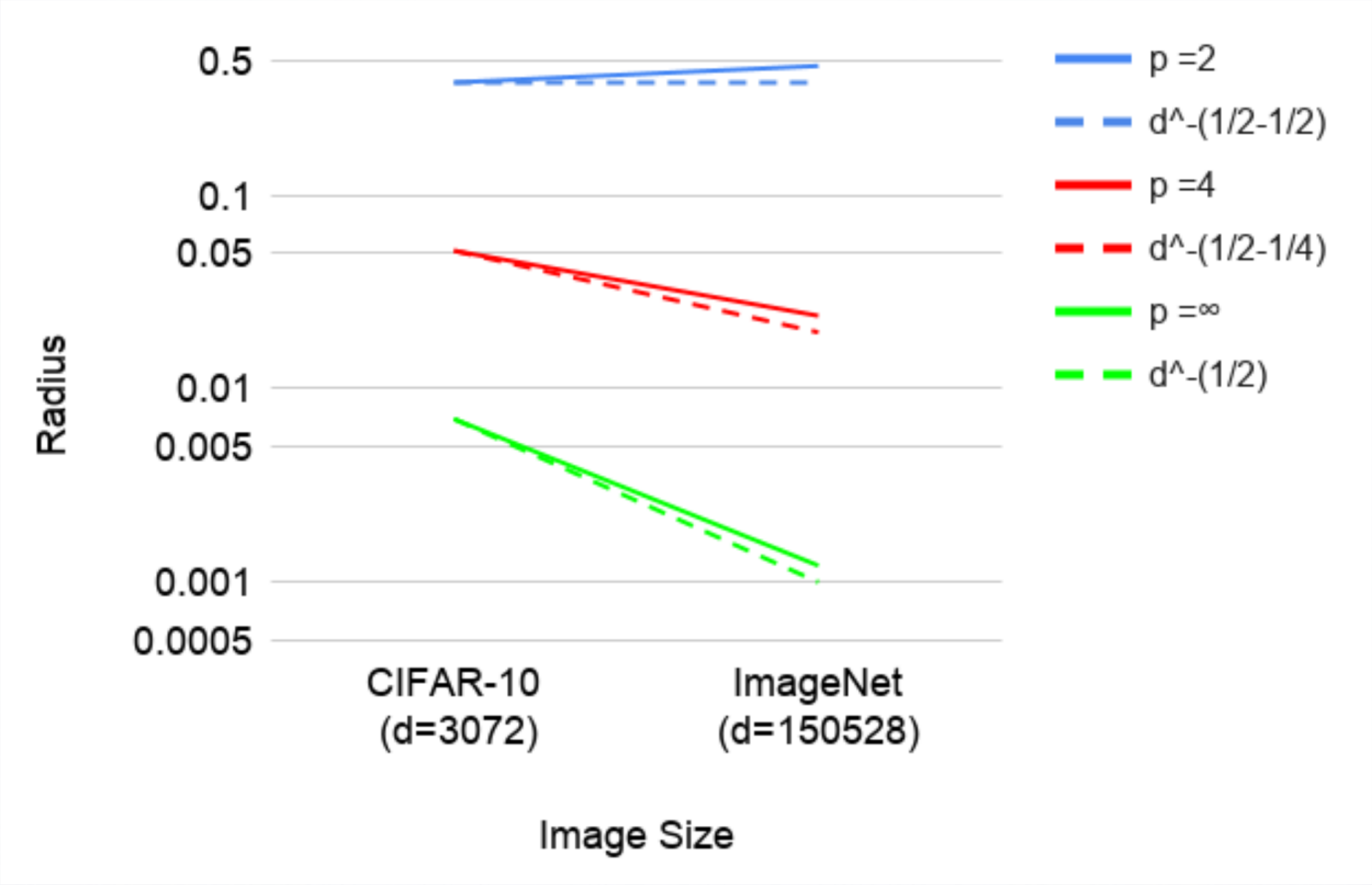
$$r_p = \frac{\sigma}{2d^{\frac{1}{2}-\frac{1}{p}}}\Big(\Phi^{-1}(p_1(\mathbf{x})) - \Phi^{-1}(p_2(\mathbf{x}))\Big)$$

- Using any symmetric i.i.d. smoothing:

$$r_p^* \leq \frac{\sigma}{2\sqrt{2}d^{\frac{1}{2}-\frac{1}{p}}}\Big(\frac{1}{\sqrt{1-p_1(\mathbf{x})}} + \frac{1}{\sqrt{p_2(\mathbf{x})}}\Big)$$

Up to some constants, Gaussian smoothing is optimal within i.i.d. smoothing distributions against L_p attacks

CIFAR-10 vs. ImageNet



- Gaussian smoothing with $\sigma = 0.25$
- The certified radius decreases with dimension with a scaling $\sim d^{1/2-1/p}$

Uniform Smoothing for L_1 attacks

- A smoothed classifier: $$\bar{f}(\mathbf{x}) := \mathbb{E}_\epsilon \left[f(\mathbf{x} + \epsilon) \right]$$ $$\epsilon \sim \mathcal{U}^d(-\lambda, \lambda)$$

- **Theorem** (Lee et al.'19)

  $\bar{f}(\mathbf{x})$ is $1/(2\lambda)$-Lipschitz with respect to L_1 norm

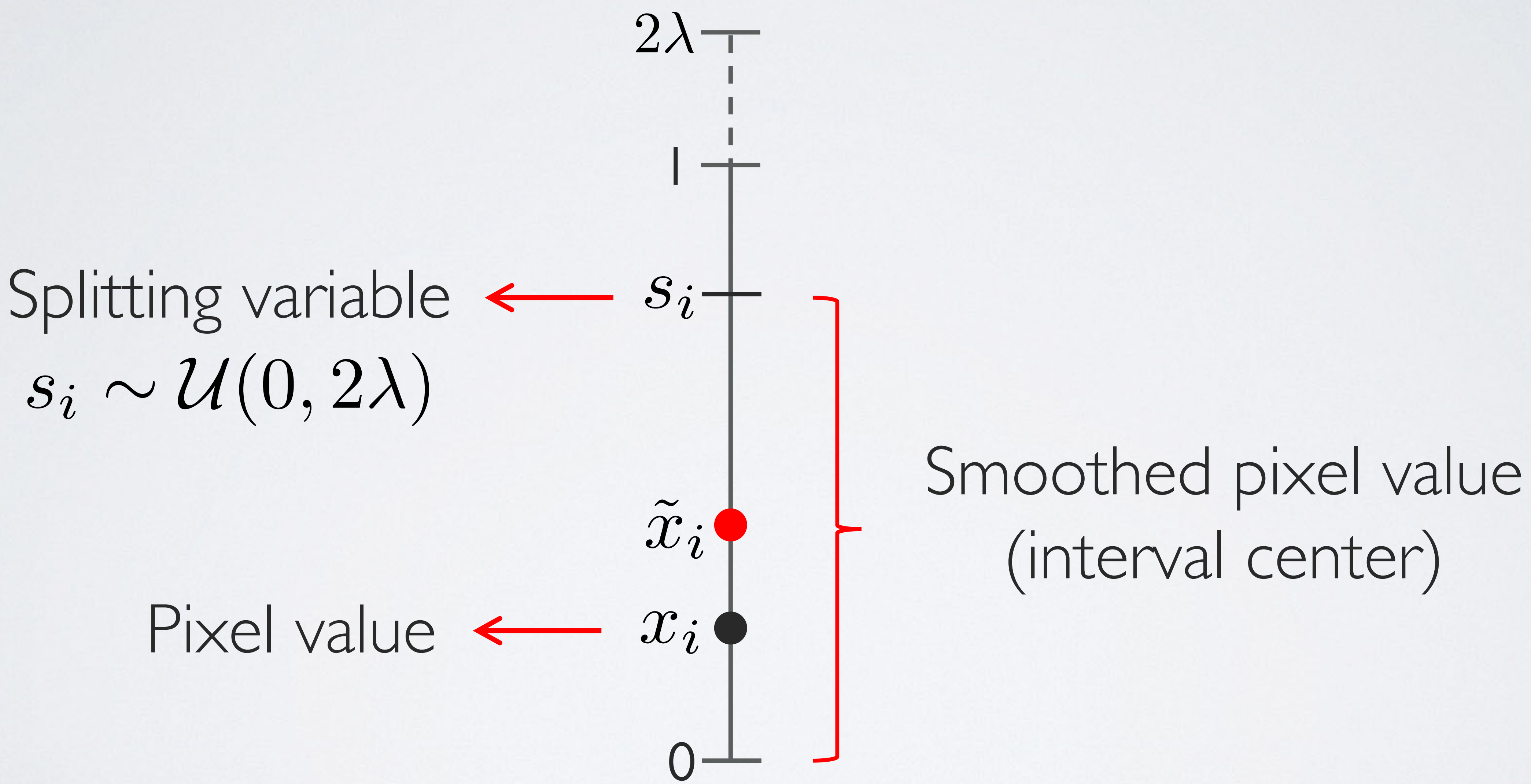
- Yang et al. (2020) shows that this is (in a sense) optimal for the L_1 norm (among additive smoothing distributions)

- Uniform additive noise requires independence → smoothing is done in d-dimensional space

Curse of Dimensionality on Randomized Smoothing for Certifiable Robustness, Kumar, Levine, Goldstein, F., ICML 2020

Non-additive Smoothing with Splitting Noise

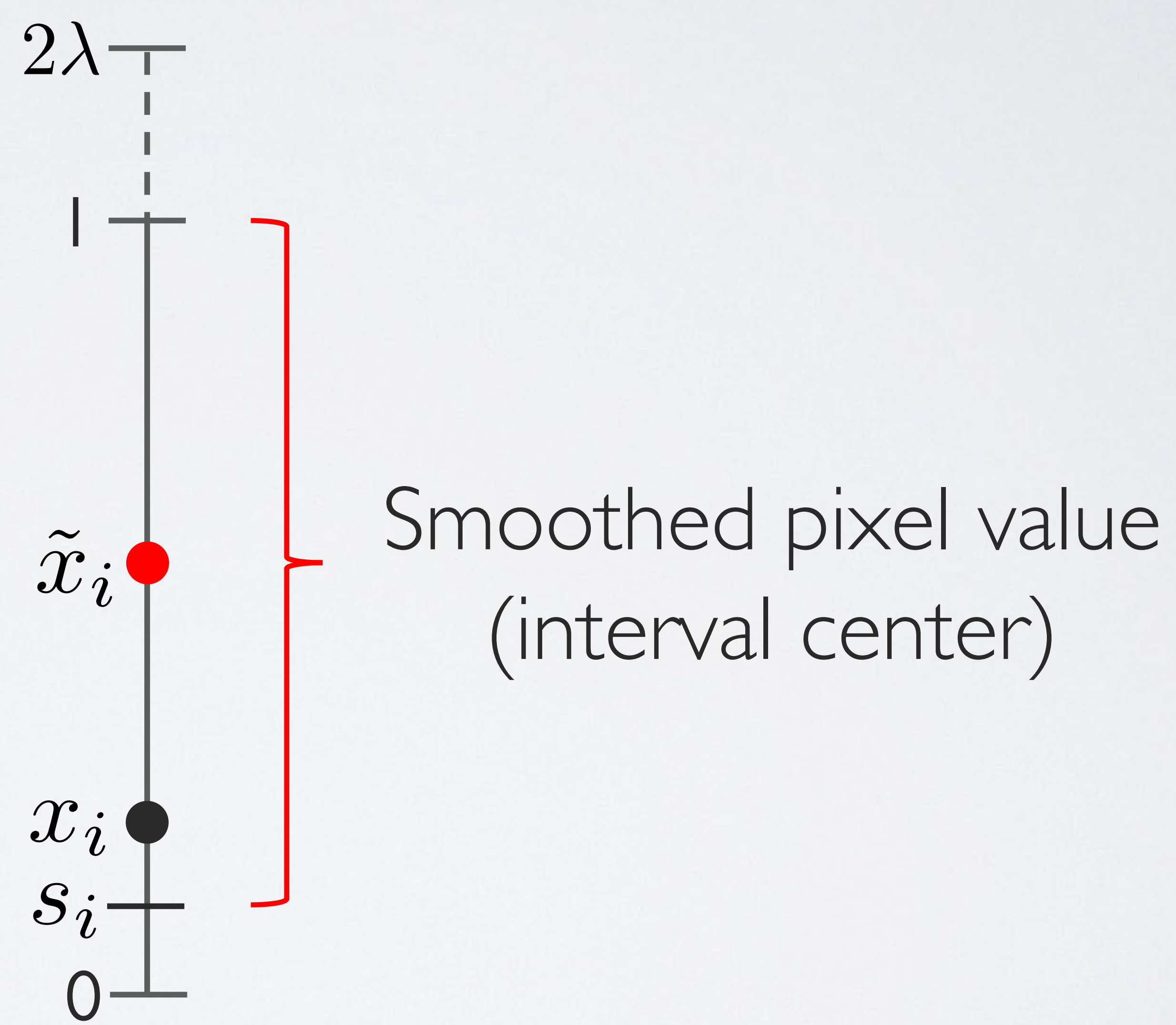
- SSN: a smoothed classifier with splitting noise



Levine and F., improved, Deterministic Smoothing for L_1 Certified Robustness, ICML 2021

Non-additive Smoothing with Splitting Noise

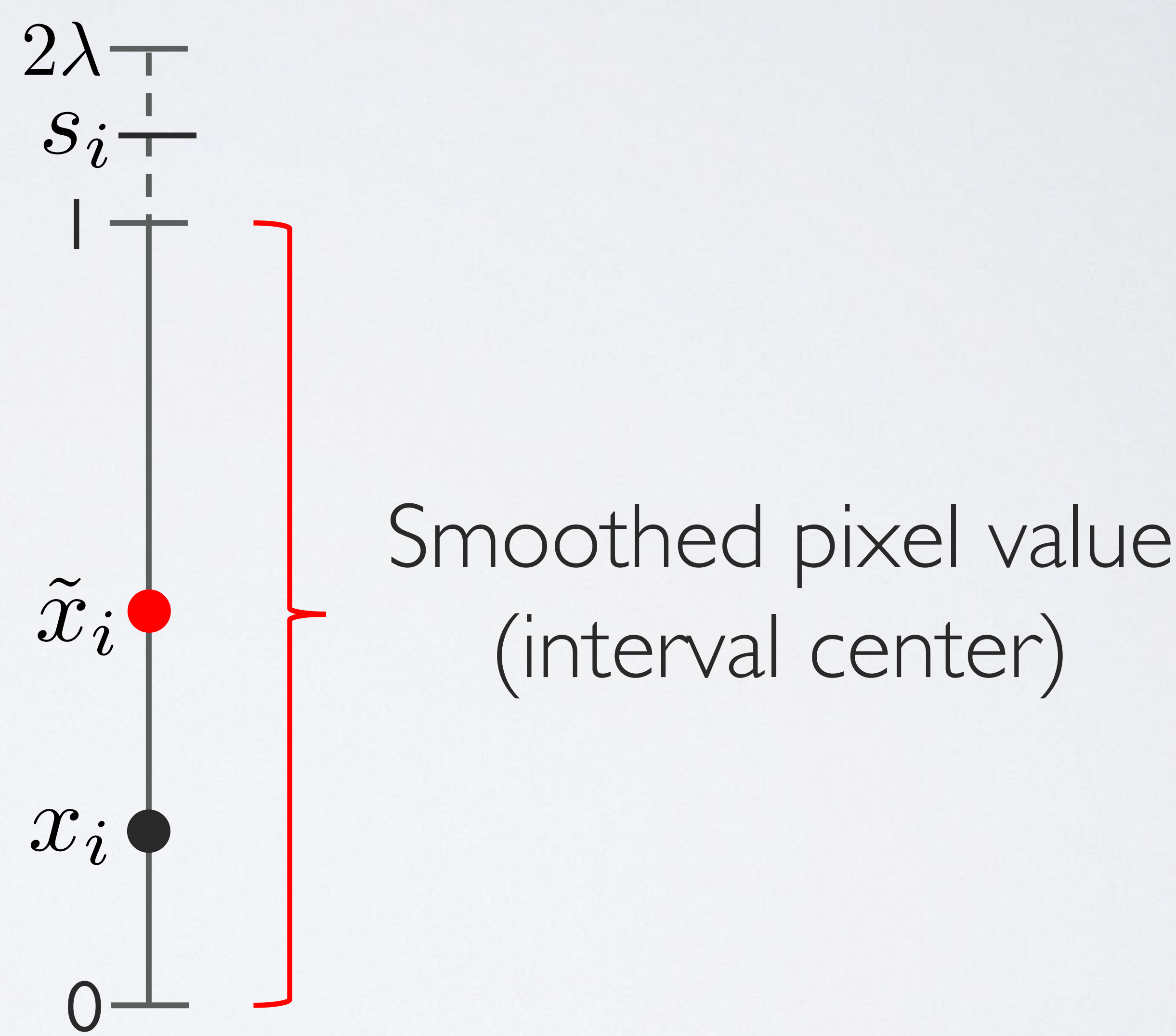
- SSN: a smoothed classifier with splitting noise



Levine and F., improved, Deterministic Smoothing for L_1 Certified Robustness, ICML 2021

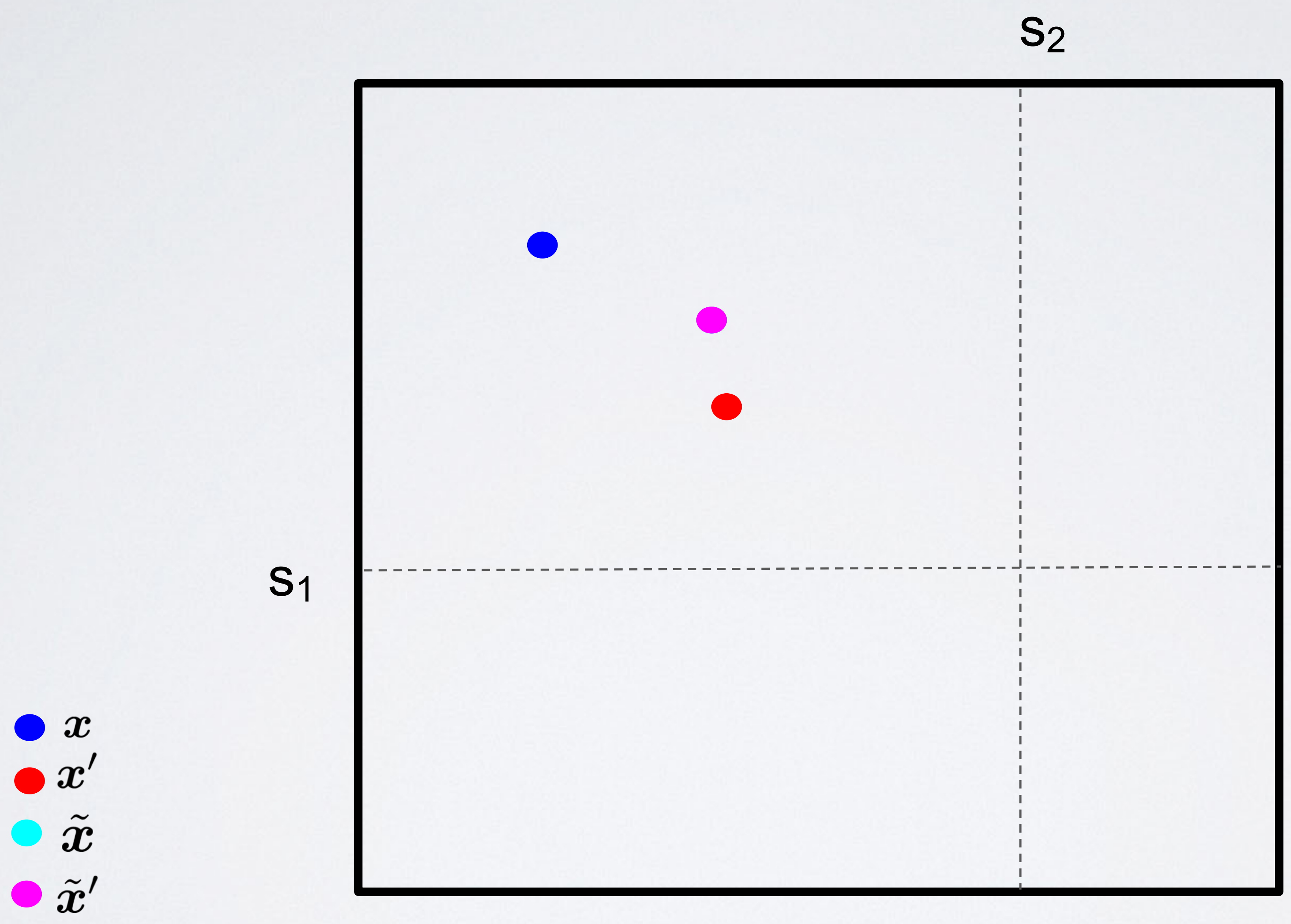
Non-additive Smoothing with Splitting Noise

- SSN: a smoothed classifier with splitting noise

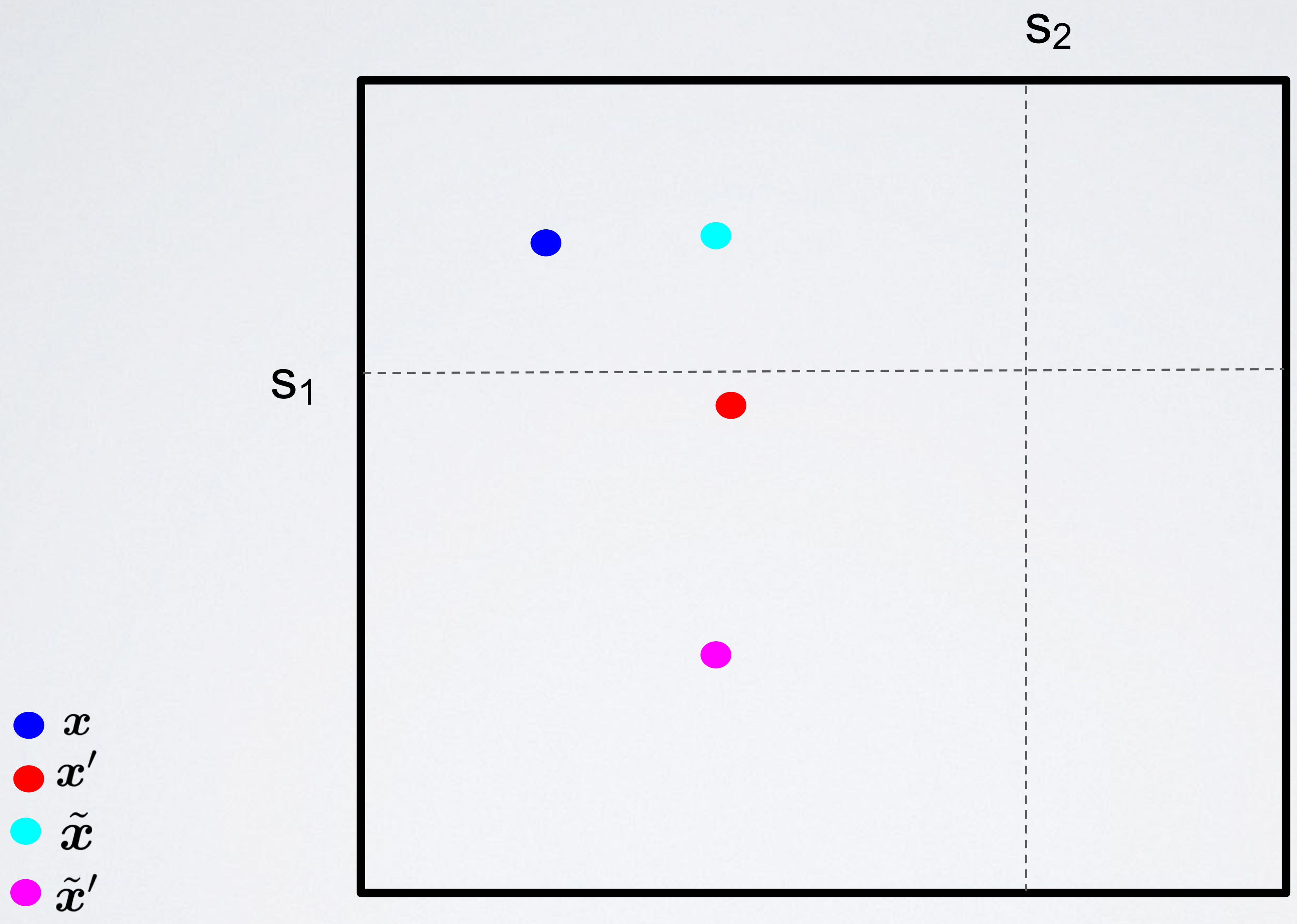


Levine and F., improved, Deterministic Smoothing for L_1 Certified Robustness, ICML 2021

Smoothing with Splitting Noise



Smoothing with Splitting Noise

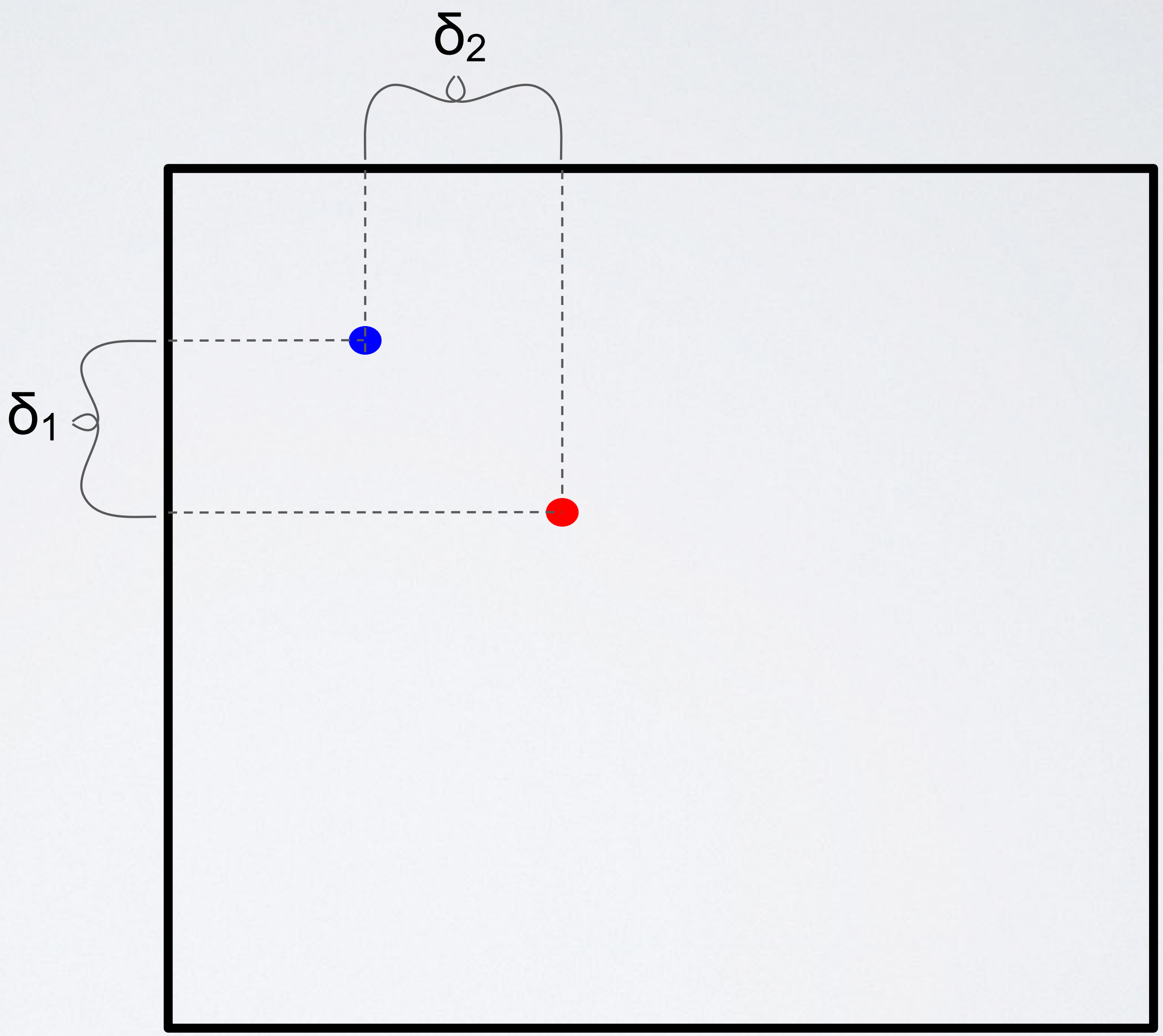


Smoothing with Splitting Noise

$$\Pr_{\boldsymbol{s}}[\tilde{\boldsymbol{x}} \neq \tilde{\boldsymbol{x}'}] = \Pr_{\boldsymbol{s}}\left[\bigcup_{i=1}^{d} \tilde{x}_i \neq \tilde{x}'_i\right]$$

$$\leq \sum_{i=1}^{d} \frac{|\delta_i|}{2\lambda} = \frac{\|\delta\|_1}{2\lambda}$$

Union Bound: holds regardless of joint distribution of $\boldsymbol{s_i}$'s



Non-additive Smoothing with Splitting Noise

- SSN: a smoothed classifier with splitting noise

  For **any** joint distribution **s** with each $s_i \sim \mathcal{U}(0, 2\lambda)$

  $$\bar{f}(\mathbf{x}) := \mathbb{E}_{\mathbf{s}}\left[f(\tilde{\mathbf{x}})\right]$$

- **Theorem** (Levine & F. ICML'21)

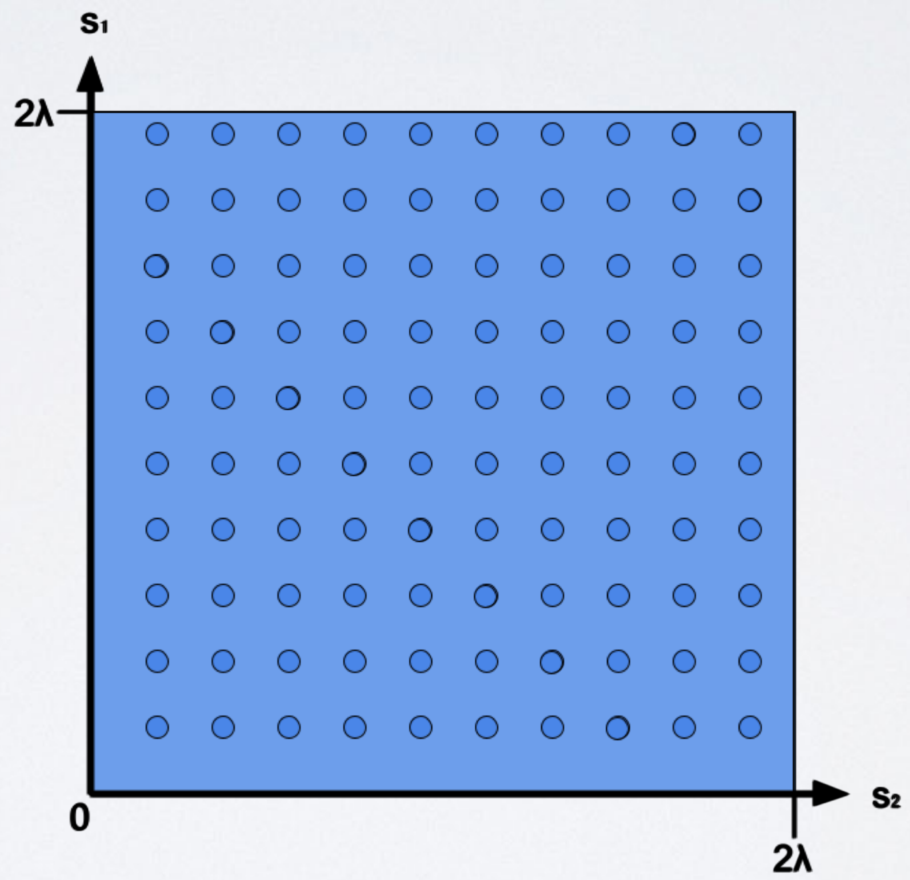
  $\bar{f}(\mathbf{x})$ is $1/(2\lambda)$-Lipschitz with respect to L_1 norm

- SSN is non-additive

- Splitting noise component does NOT require independence → smoothing is done in one-dimensional space and can be de-randomized

Derandomized Smoothing with Splitting Noise

- **Goal:** evaluate all possible noise realizations, to compute $\bar{f}(\mathbf{x})$ exactly.
- For quantized inputs (e.g. in images), s_i is uniform on a finite set
- Let q := number of quantizations (e.g. 256 for images)

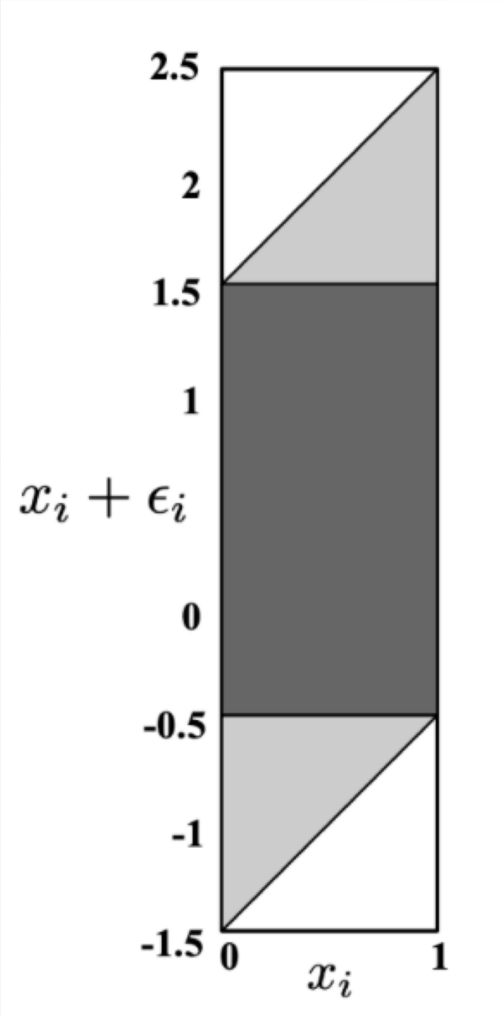


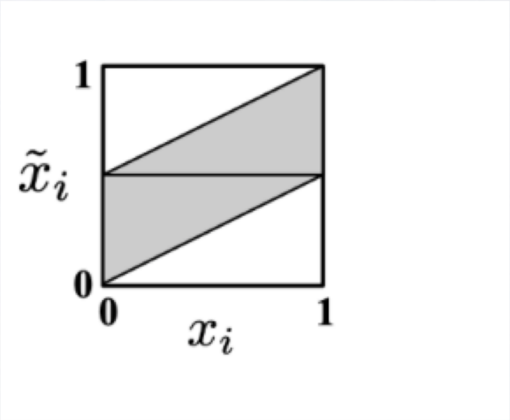
- If independence was required (i.e. in uniform smoothing), this would mean $(2\lambda q)^d$ evaluations → computationally expensive
- But with SSN, **independence is not required:** only need $2\lambda q$ evaluations.

SSN - Representation Differences

$$x_i + \epsilon_i \sim \begin{cases} \mathcal{U}(x_i - \lambda, 1 - \lambda) & \text{w. prob. } \frac{1-x_i}{2\lambda} \\ \mathcal{U}(1 - \lambda, \lambda) & \text{w. prob. } \frac{2\lambda-1}{2\lambda} \\ \mathcal{U}(\lambda, x_i + \lambda) & \text{w. prob. } \frac{x_i}{2\lambda} \end{cases}$$

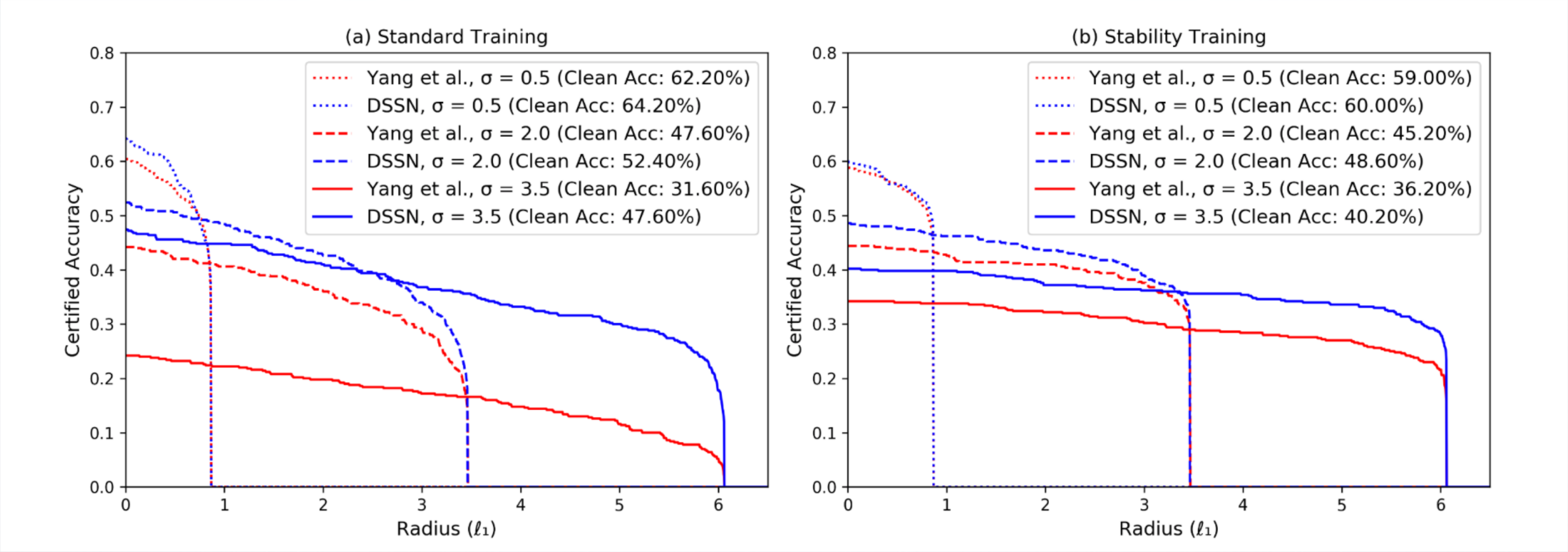
$$\tilde{x}_i \sim \begin{cases} \frac{\mathcal{U}(x_i,1)}{2} & \text{w. prob. } \frac{1-x_i}{2\lambda} \\ \frac{1}{2} & \text{w. prob. } \frac{2\lambda-1}{2\lambda} \\ \frac{\mathcal{U}(1,x_i+1)}{2} & \text{w. prob. } \frac{x_i}{2\lambda} \end{cases}$$





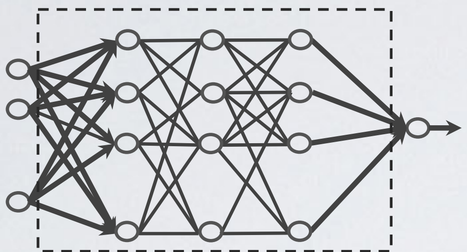
Empirical Results

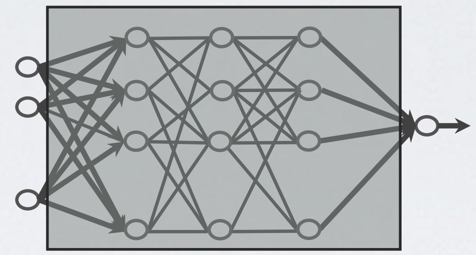
- Our method established new state-of-the-art results on ImageNet



Landscape of Provable Defenses

Amount of the network information used in the defense





L_p:

Lipschitz/Curvature Bounds
Singla & F., ICML'20
Singla & F., ICML'21

IBP/Convex
Wong & Kolter, '18
Gowal, et al., '18, Mirman 2018, Zhang 2019

Randomized Smoothing
Cohen et al. '19, Li et al. '18, Salman et al. '19, Lecuyer et al. '19, Teng et al. '20, Lee et al. '19, Yang et al. '20, KLGF., ICML 20, KLFG, NeurIPS 20, Levine, F. ICML'21

Non-L_p:

Patch Threat
Chaing et al.'20

Sparse Threat
Lee et al. '19, Levine, F. AAAI'20

Wasserstein Threat
Levine, F. AISTATS '20

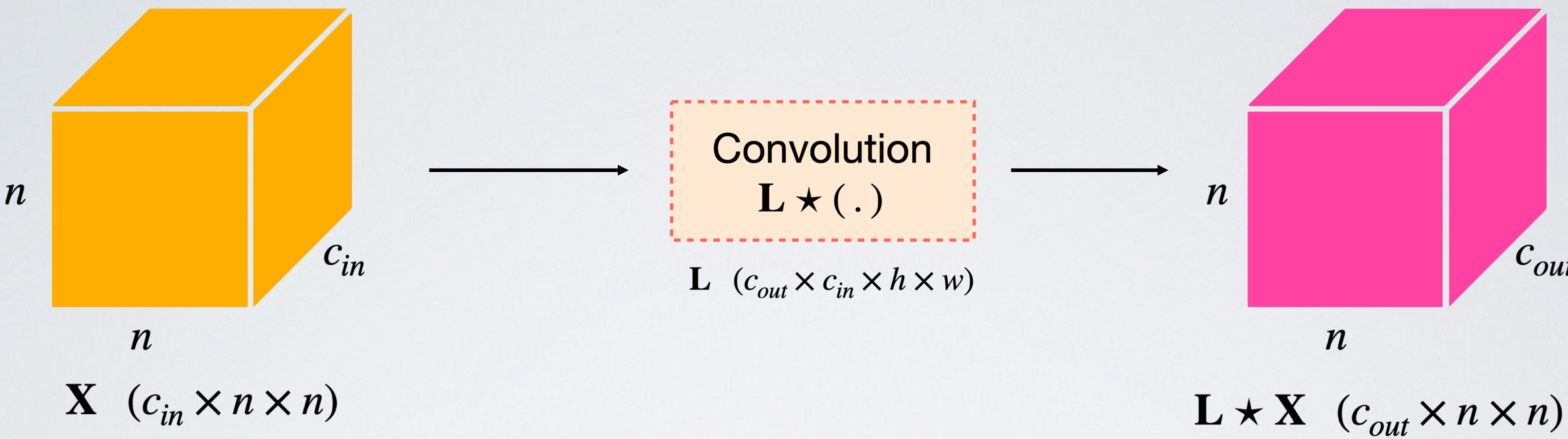
Patch Threat
Levine, F. NeurIPS'20, Xiang et al.'20

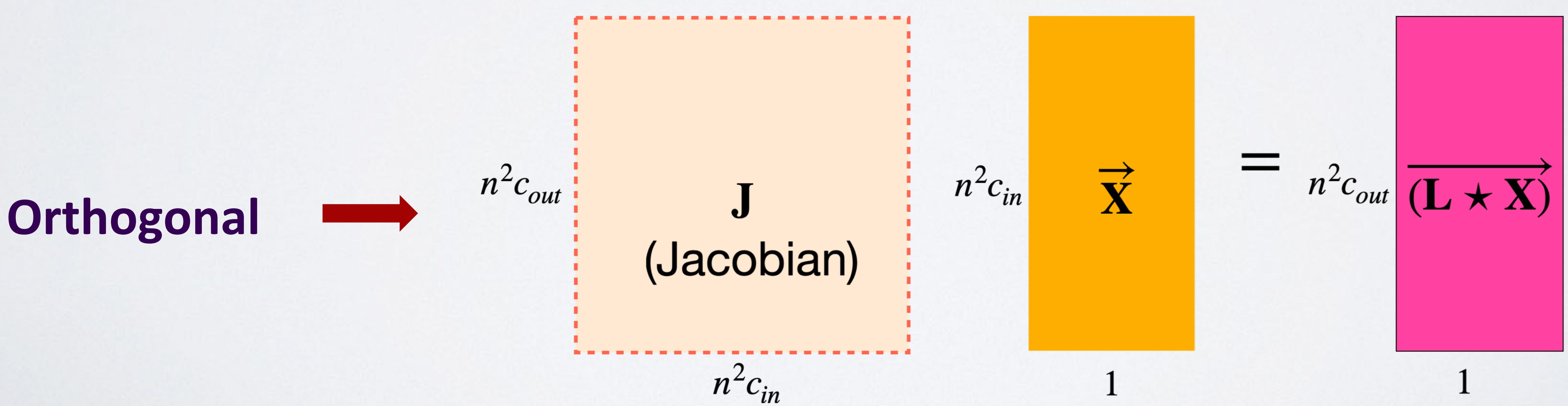
Orthogonal Convolutions

- Goal: develop convolution layers with orthogonal Jacobians → Lipschitz CNNs → provable robustness against L2 adversarial attacks

- Related works:
  - Orthogonal convolutions: BCOP (Li et al.'19); Cayley (Trockman, Kolter, 2021)
  - Spectral analysis of convolutions: Sedghi et al. (2018), Singla & F. (2021)

Orthogonal Convolutions

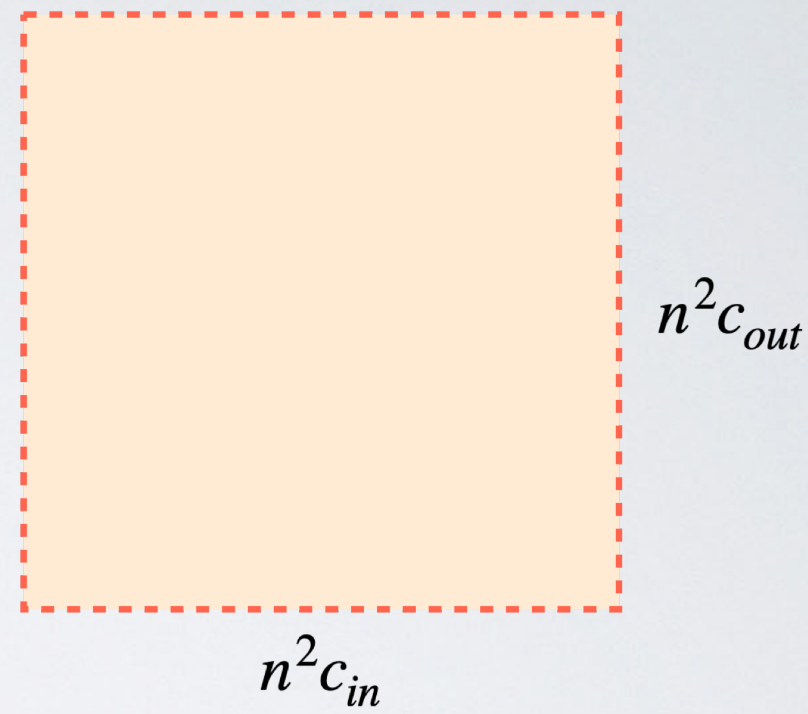




Why orthogonalize convolution layers?

$$\mathbf{J} = \nabla_{\vec{\mathbf{X}}} \overrightarrow{(\mathbf{L} \star \mathbf{X})} =$$

(Jacobian)



- Exploding and vanishing gradients [Pennington et al. 2017, Xiao et al. 2018]
- Robustness [Szegedy et al. 2014, Cisse et al. 2017]
- Generalization bounds [Long et al. 2019]
- Wasserstein distance estimation [Villani et al. 2008]

Key mathematical properties

- $\mathbf{A} = -\mathbf{A}^T \implies \exp(\mathbf{A})$ is orthogonal

- $\exp(\mathbf{A}) = \sum_{i=0}^{\infty} \frac{\mathbf{A}}{i!} = \mathbf{I} + \frac{\mathbf{A}}{1!} + \frac{\mathbf{A}^2}{2!} + \frac{\mathbf{A}^3}{3!} + \dots$

Skew-symmetric convolution filters

- **Theorem:** A convolution filter $\mathbf{L}$ is Skew-Symmetric if and only if

Skew Symmetric →

$$\mathbf{L} = \mathbf{M} - \text{conv\_transpose}(\mathbf{M})$$

Jacobian $(\mathbf{J})$ $(\mathbf{J}^T)$

Flip the height and width dimensions, transpose the two channel dimensions

Computing the exponential series

- Given an input X, convolution filter L of appropriate sizes

$$\mathbf{L} \star^1 \mathbf{X} = \mathbf{L} \star \mathbf{X}$$
$$\mathbf{L} \star^n \mathbf{X} = \mathbf{L} \star^{n-1} (\mathbf{L} \star \mathbf{X})$$

$$\implies \overrightarrow{\mathbf{L} \star^n \mathbf{X}} = \mathbf{J}^n \vec{\mathbf{X}} \quad \text{where } \overrightarrow{\mathbf{L} \star \mathbf{X}} = \mathbf{J}\vec{\mathbf{X}}$$

$$\mathbf{L} \star_e \mathbf{X} = \mathbf{X} + \frac{\mathbf{L} \star \mathbf{X}}{1!} + \frac{\mathbf{L} \star^2 \mathbf{X}}{2!} + \frac{\mathbf{L} \star^3 \mathbf{X}}{3!} + \cdots$$

$$\exp(\mathbf{J})\,\mathbf{X} = \overrightarrow{\mathbf{L} \star_e \mathbf{X}}$$

Convolution exponential
[Hoogeboom et al. 2020]

Approximation guarantee

- **Theorem**: If J is skew symmetric:

$$\left\| \exp(\mathbf{J}) - \sum_{i=0}^{k-1} \frac{\mathbf{J}^i}{i!} \right\|_2 \leq \frac{\|\mathbf{J}\|_2^k}{k!}$$

Orthogonal matrix

Our finite term approximation

Approximation Error (< **2.42** x **10**$^{-6}$ in our experiments)

- Approximation error **decays exponentially with the number of terms k** used for approximation

Results for provably robust training

~10% improvement for deeper (>25 layers) networks

2-3x decrease

Number of layers	Standard Accuracy		Provably Robust Accuracy		Train time/epoch (secs)	
	BCOP	SOC	BCOP	SOC	BCOP	SOC
5	74.35%	**75.78%**	58.01%	**59.16%**	96.153	**31.096**
10	74.47%	**76.48%**	58.48%	**60.82%**	122.115	**48.242**
15	73.86%	**76.68%**	57.39%	**61.30%**	145.944	**63.742**
20	69.84%	**76.43%**	52.10%	**61.92%**	170.009	**77.226**
25	68.26%	**75.19%**	49.92%	**60.18%**	207.359	**98.534**
30	64.11%	**74.47%**	43.39%	**59.04%**	227.916	**110.531**
35	63.05%	**73.70%**	41.72%	**58.44%**	267.272	**130.671**
40	60.17%	**71.63%**	38.87%	**54.36%**	295.350	**144.556**

Results for standard/adversarial training

Model	Standard Accuracy		
	Vanilla	BCOP	SOC
Resnet-18	95.10%	92.38%	**94.24%**
Resnet-34	95.54%	93.79%	**94.44%**
Resnet-50	95.47%	OOM Error	**94.68%**

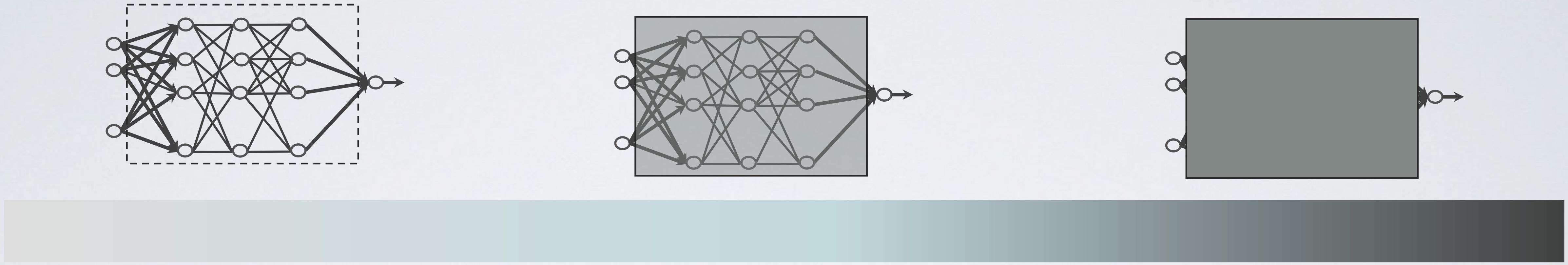
Results using **standard training**

Model	Standard Accuracy			Empirical Robust Accuracy		
	Vanilla	BCOP	SOC	Vanilla	BCOP	SOC
Resnet-18	83.05%	79.26%	**82.24%**	59.87%	54.80%	**58.95%**

Results using **adversarial training**

Landscape of Provable Defenses

Amount of the network information used in the defense



Lipschitz/Curvature Bounds

Singla & F., ICML'20
Singla & F., ICML'21

IBP/Convex

Wong & Kolter, '18
Gowal, et al., '18, Mirman 2018, Zhang 2019

Randomized Smoothing

Cohen et al. '19, Li et al. '18, Salman et al. '19, Lecuyer et al. '19, Teng et al. '20, Lee et al. '19, Yang et al. '20, KLGF., ICML 20, KLFG, NeurIPS 20, Levine, F. ICML'21

L_p:

Non-L_p:

Patch Threat

Chaing et al.'20

Sparse Threat

Lee et al. '19, Levine, F. AAAI'20

Wasserstein Threat

Levine, F. AISTATS '20

Patch Threat

Levine, F. NeurIPS'20, Xiang et al.'20

Sparse Adversarial Attacks

- Adversary can change up to ρ pixels

Input Image

Classification label: 3

$\rho = 25$ →

Adv. Example

Classification label: 5

Certifiable Defense against Sparse Adversarial Attacks

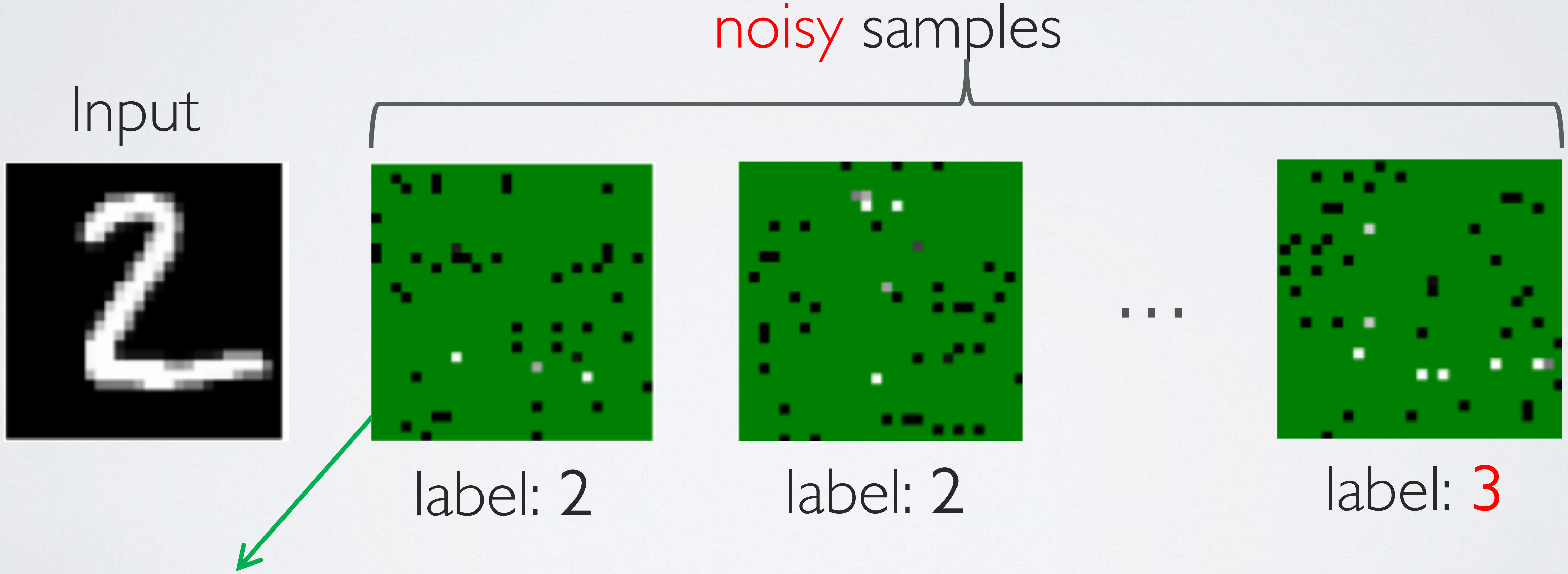
- **Lee et al '19:** With some probability, randomize the value of each pixel. Then, take the consensus among randomizations.

- Gives median certified robustness of **4** pixels on MNIST, **one** pixel on ImageNet-1000.

- Question: is there a better smoothing distribution for sparse attacks?

Our Approach: Randomized Ablation

- Use **k** randomly selected pixels (out of **d**) in classification
- $p_i(x)$: probability that **x** gets the label **i** using randomly ablated samples



NULL pixels: encoded far from the retained pixels

Robustness Certificate

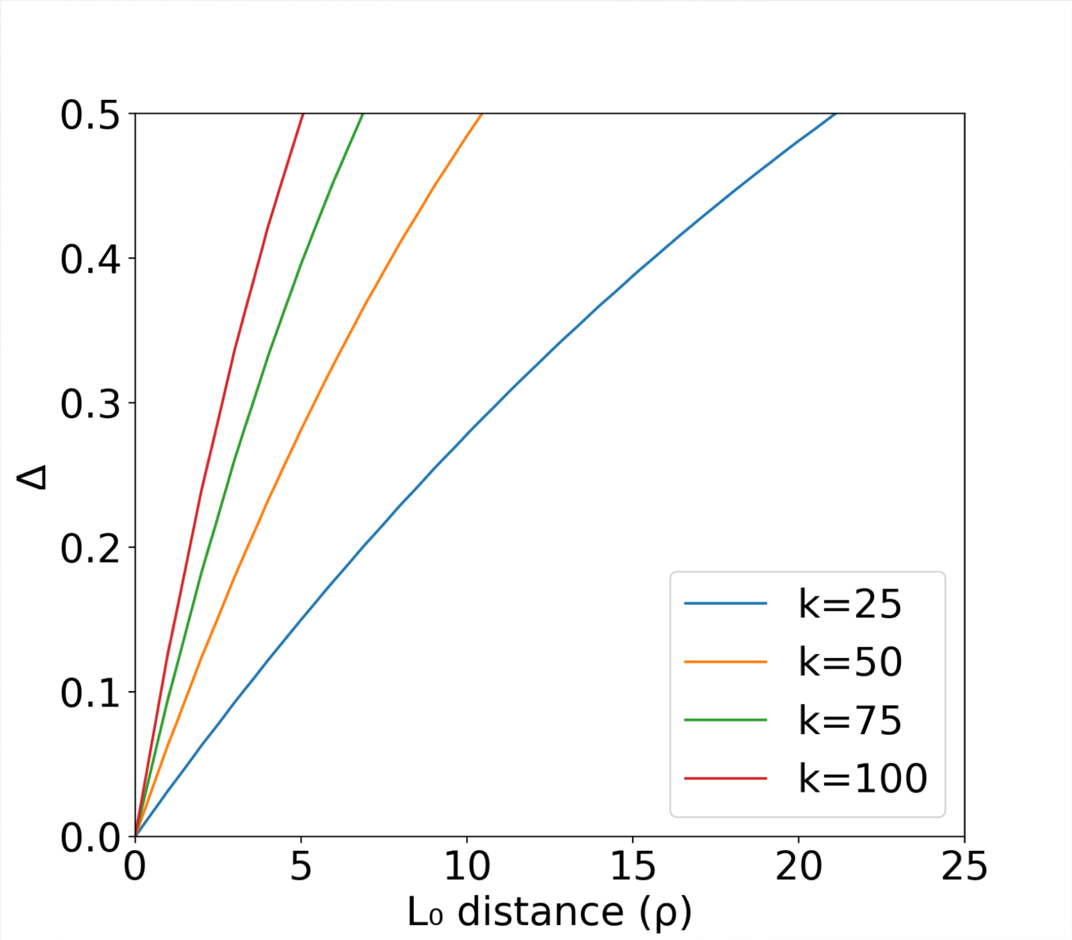
- Theorem (Levine, F. AAAI'20)

For inputs x and x' with $\|\mathbf{x} - \mathbf{x}'\|_{\ell_0} \leq \rho$, for all i

$$|p_i(\mathbf{x}) - p_i(\mathbf{x}')| \leq \Delta$$

where $\Delta = 1 - \frac{\binom{d-\rho}{k}}{\binom{d}{k}}$

probability that **any** of adv. perturbed pixels is used in classification



Robustness vs Accuracy Trade-off

- Increasing *k* boosts classification accuracy but also increases Δ
- Empirically, there exists a *k* that achieves maximum robustness

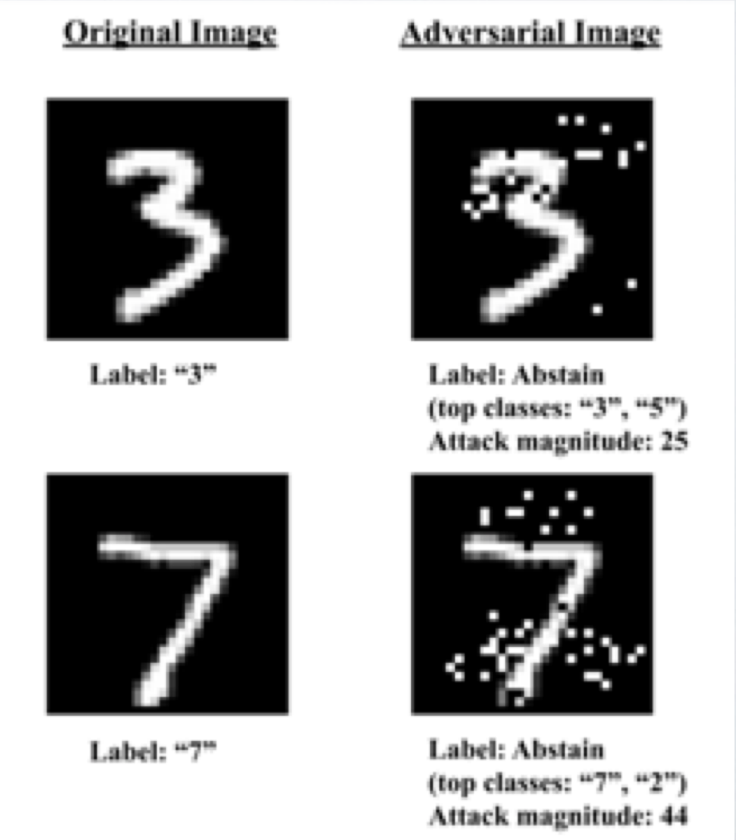
Retained pixels k	Classification accuracy (Percent abstained)	Median certified robustness
5	32.32% (5.65%)	N/A
10	74.90% (5.08%)	0
15	86.09% (2.82%)	0
20	90.29% (1.81%)	3
25	93.05% (1.02%)	5
30	94.68% (0.77%)	7
35	95.40% (0.66%)	7
40	96.27% (0.52%)	8
45	**96.72% (0.45%)**	**8**
50	97.16% (0.32%)	7
55	97.41% (0.34%)	7
60	97.78% (0.18%)	7
65	98.05% (0.15%)	6
70	98.18% (0.20%)	6
75	98.28% (0.20%)	6
80	98.37% (0.12%)	5
85	98.57% (0.12%)	5
90	98.58% (0.16%)	5
95	98.73% (0.11%)	5
100	98.75% (0.16%)	4

Empirical Results

- Median certified robustness:
  - MNIST: 8 pixels
  - ImageNet: 16 pixels

- Median empirical robustness on MNIST:

Model	Class. acc.	Median adv. attack mag.
CNN	99.1%	9.0
Binarized CNN	98.5%	11.0
Nearest Neighbor	96.9%%	10.0
L_∞-Robust (Madry et al. 2017)	98.8%	4.0
(Schott et al. 2019)	99.0%	22.0
Binarized (Schott et al. 2019)	99.0%	16.5
Our model ($k = 45$)	**96.7%**	**31.0**



Comparison with Lee et al. '19

Dataset	Median certified robustness (pixels) (Lee et al. 2019)	Median certified robustness (pixels) (our model)
MNIST	4	**8**
ImageNet	1	**16**

- Ablating pixels instead of randomizing them **preserves more information: we know which pixels** are from the original image and which are ablated.

- This can be quantified in terms of the **mutual information** between the original and ablated images.

Encoding Ablated Pixels

- **Approach one:** double the number of channels, encode NULL as (0,0)
- **Approach two:** Encoding NULL pixels as the mean value on the dataset works fine:

$\mathcal{S}_{\text{NULL}}$ encoding	Classification acc. (Pct. abstained)	Median certified robustness
MNIST		
Multichannel	**96.72% (0.45%)**	**8**
Mean	96.27% (0.43%)	7
CIFAR-10		
Multichannel	**78.25% (0.93%)**	**7**
Mean	77.71% (1.05%)	**7**

Today's Talk

- Part I: Attack = (algorithm, threat model)
  - algorithm: variable
  - threat model: fixed

- Part II: Attack = (algorithm, threat model)
  - algorithm: variable
  - threat model: variable

Perceptual Adversarial Robustness: Defense Against Unseen Threat Models,
Laidlaw, Singla, F., on arXiv

- *Key assumption:* the defender ~~knows~~ *does not know* the threat model used by the attacker

Example of Robustness Generalization

- Suppose we use the popular adversarial training to robustify a CIFAR-10 classification model against L_∞

$$\min_{\theta} \; \mathbb{E}_{(\mathbf{x},y)} \left[\max_{\substack{\delta \\ \|\delta\|_\infty \leq \rho}} \ell_{cls} \left(f_\theta(\mathbf{x} + \delta), y \right) \right]$$

→ Robust accuracy against L_∞ attacks is

$\backsimeq$ 50% 🙂

→ Robust accuracy against spatial attacks is

$\backsimeq$ 5% !!! 😱

Generalization to Unforeseen attacks

- Standard defenses have poor generalization to unforeseen adversarial attacks
- Unforeseen Attack Robustness of Adversarial Training-based defenses on CIFAR-10

Training	Union	Unseen mean	Narrow threat models: Clean	L_∞	L_2	StAdv	ReColor
Normal	0.0	0.1	94.8	0.0	0.0	0.0	0.4
AT L_∞	1.0	19.6	86.8	49.0	19.2	4.8	54.5
TRADES L_∞	4.6	23.3	84.9	52.5	23.3	9.2	60.6
AT L_2	4.0	25.3	85.0	39.5	47.8	7.8	53.5
AT StAdv	0.0	1.4	86.2	0.1	0.2	53.9	5.1
AT ReColorAdv	0.0	3.1	93.4	8.5	3.9	0.0	65.0

Laidlaw, Singla, F., ICLR' 21

- *Question:* Can we develop a defense with a generalizable robustness across various adversarial threat models?

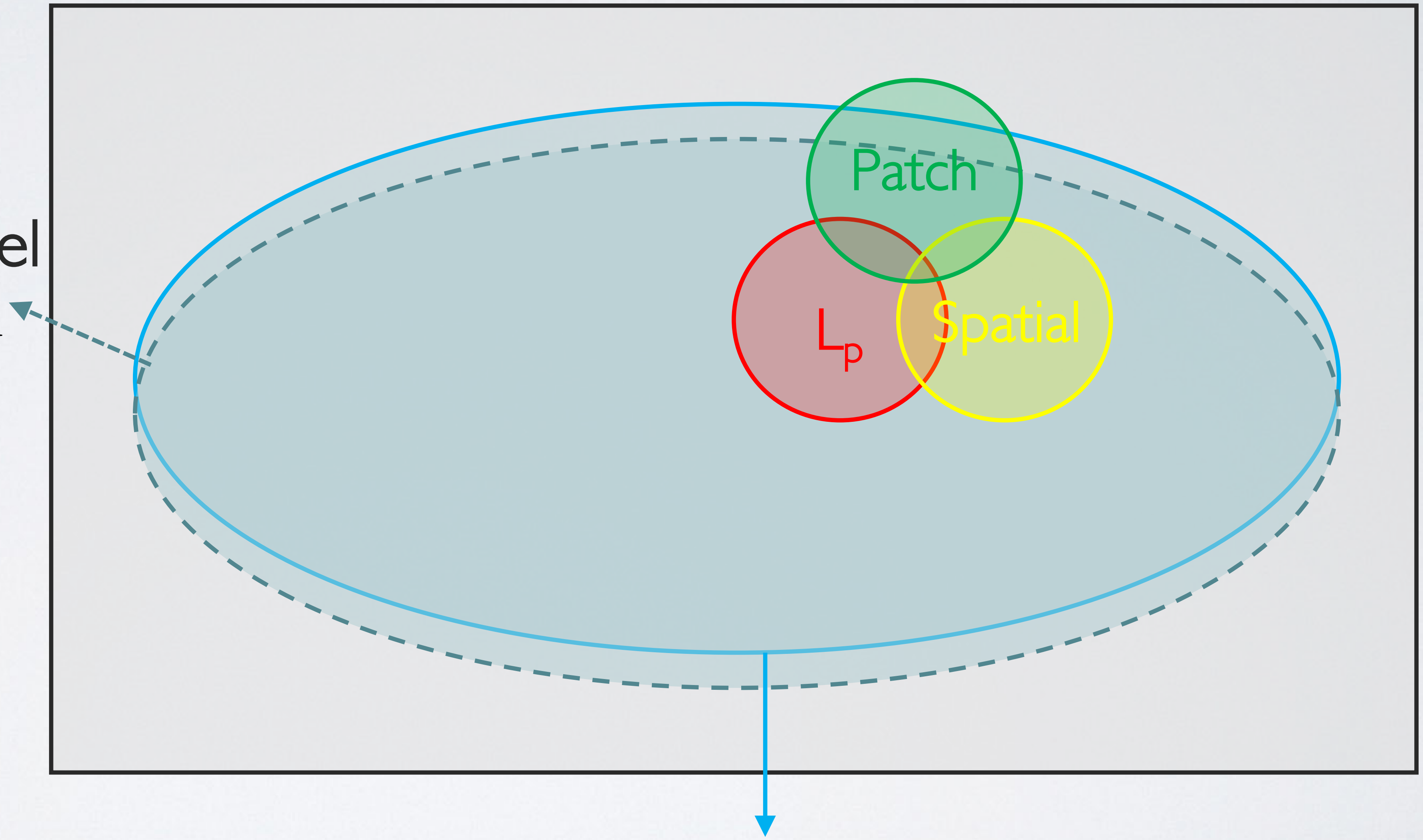
- *Yes, Perceptual Adversarial Training (PAT)*

Laidlaw, Singla, F., Perceptual Adversarial Robustness: Defense Against Unseen Threat Models, ICLR 2021

Relationship Between Threat Models

Unrestricted threat model
$\{\mathbf{x}' : f_{\text{human}}(\mathbf{x}') = f_{\text{human}}(\mathbf{x})\}$

Proposed: Neural Perceptual Threat Model
$\{\mathbf{x}' : d_{\text{neural}}(\mathbf{x}', \mathbf{x}) \leq \rho\}$



True Perceptual threat model
$\{\mathbf{x}' : d_{\text{perc}}(\mathbf{x}', \mathbf{x}) \leq \rho\}$

Proposed: Neural Perceptual Threat Model

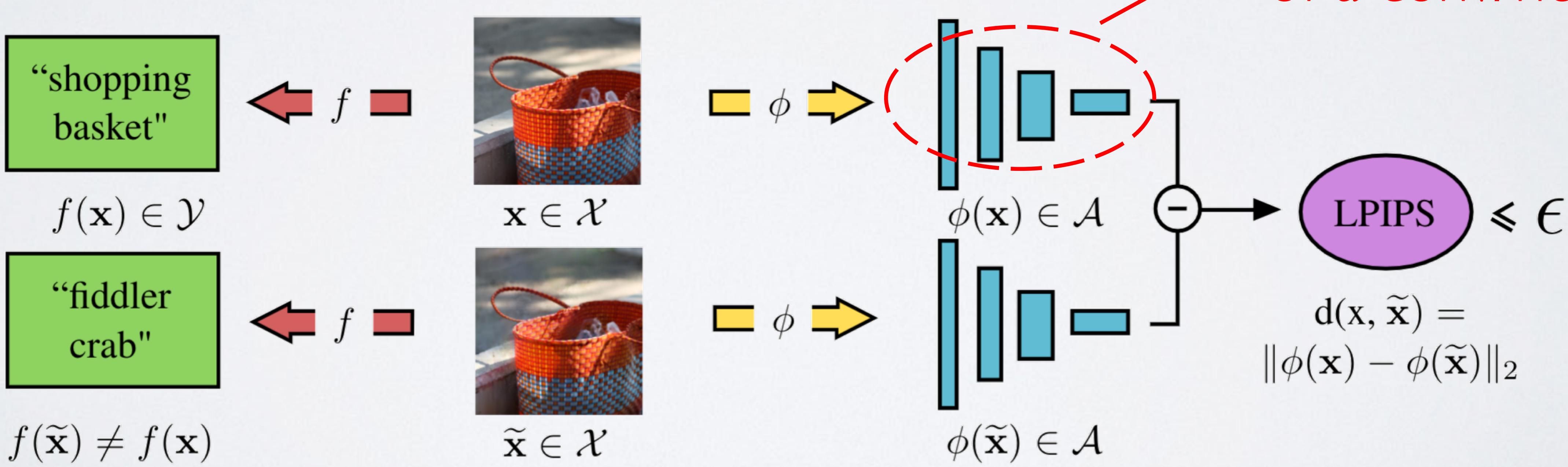
- Idea: use deep networks to approximate the true perceptual distance in the adversarial threat model

- Challenges:
  - What are proper neural perceptual distance functions?
  - The attack is a more complex optimization problem due to non-convexity of constraints
  - The defense has a new front of vulnerability: the threat model itself can be attacked

Neural Perceptual Distances

- An age-old problem in computer vision: several surrogate functions exist including SSIM (wang et al. '04) and LPIPS (Zhang et al.'18)
- We use the LPIPS (Learned Perceptual Image Patch Similarity) as $d_{\text{neural}}(\mathbf{x}, \mathbf{x}')$

Internal activations of a conv. net g(.)



Perceptual attacks

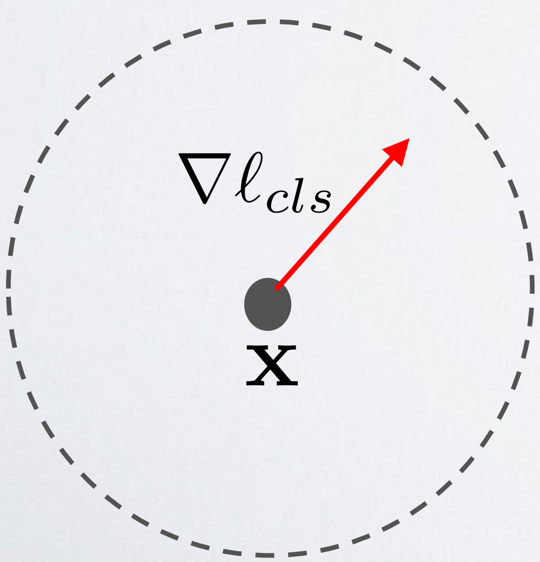
L2 Attacks

$$\max_{\mathbf{x}'} \ \ell_{cls}(f(\mathbf{x}'), y)$$
$$\|\mathbf{x} - \mathbf{x}'\| \leq \rho$$

1st order apx

$$\max_{\mathbf{x}'} \ \nabla \ell_{cls}(f(\mathbf{x}), y)^T \delta$$
$$\|\delta\| \leq \rho$$

$$\delta^* \propto \nabla \ell_{cls}$$



Perceptual Attacks

$$\max_{\mathbf{x}'} \ \ell_{cls}(f(\mathbf{x}'), y)$$
$$d_{\text{neural}}(\mathbf{x}, \mathbf{x}') = \|\phi(\mathbf{x}) - \phi(\mathbf{x}')\| \leq \rho$$

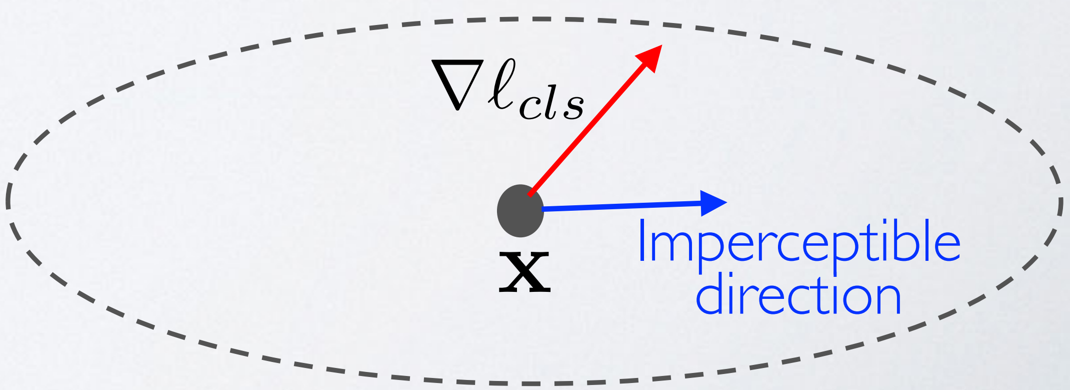
1st order apx

$$\max_{\mathbf{x}'} \ \nabla \ell_{cls}(f(\mathbf{x}), y)^T \delta$$
$$\|J\delta\| \leq \rho$$

Jacobian of ϕ

$$\delta^* \propto (J^\top J)^{-1}(\nabla \ell_{cls})$$

Efficient comp. via conjugate gradient



Perceptual Adversarial Attacks

- We introduce two perceptual attacks:
  - ✓ Perceptual Projected Gradient Descent (PPGD)
    → in par with L2 PGD attack
  - ✓ Lagrangian Perceptual Attacks (LPA)
    → in par with C&W attack

- Choices for the perceptual network g(.):
  - ✓ Same perceptual and classification networks → self-bounded attack
  - ✓ Different perceptual and classification networks → externally-bounded attack

PPGD: Perceptual Projected Gradient Descent

- PPGD Attack:
  - Solve the first-order approximation
  - Project back onto the feasible set
- Lemma (Laidlaw, Singla, F. '20):

  The first-order optimal adversarial perturbation under the perceptual threat model is:

$$\mathbf{x}' = \mathbf{x} + \eta \frac{(J^\top J)^{-1}(\nabla \hat{f})}{\|(J^{+})^\top(\nabla \hat{f})\|_2}$$

J : Jacobian of ϕ w.r.t. $\mathbf{x}$

$\hat{f} = \ell_{cls} \circ f$

- Efficient computation using conjugate gradient method
- Approximate projection using the bisection root finding method

LPA: Lagrangian Perceptual Attacks

- LPA Attack:

$$\max_{\mathbf{x}'} \ \ell_{cls}(f(\mathbf{x}'), y) - \lambda \max\Big(0, \|\phi(\mathbf{x}') - \phi(\mathbf{x})\| - \rho\Big)$$

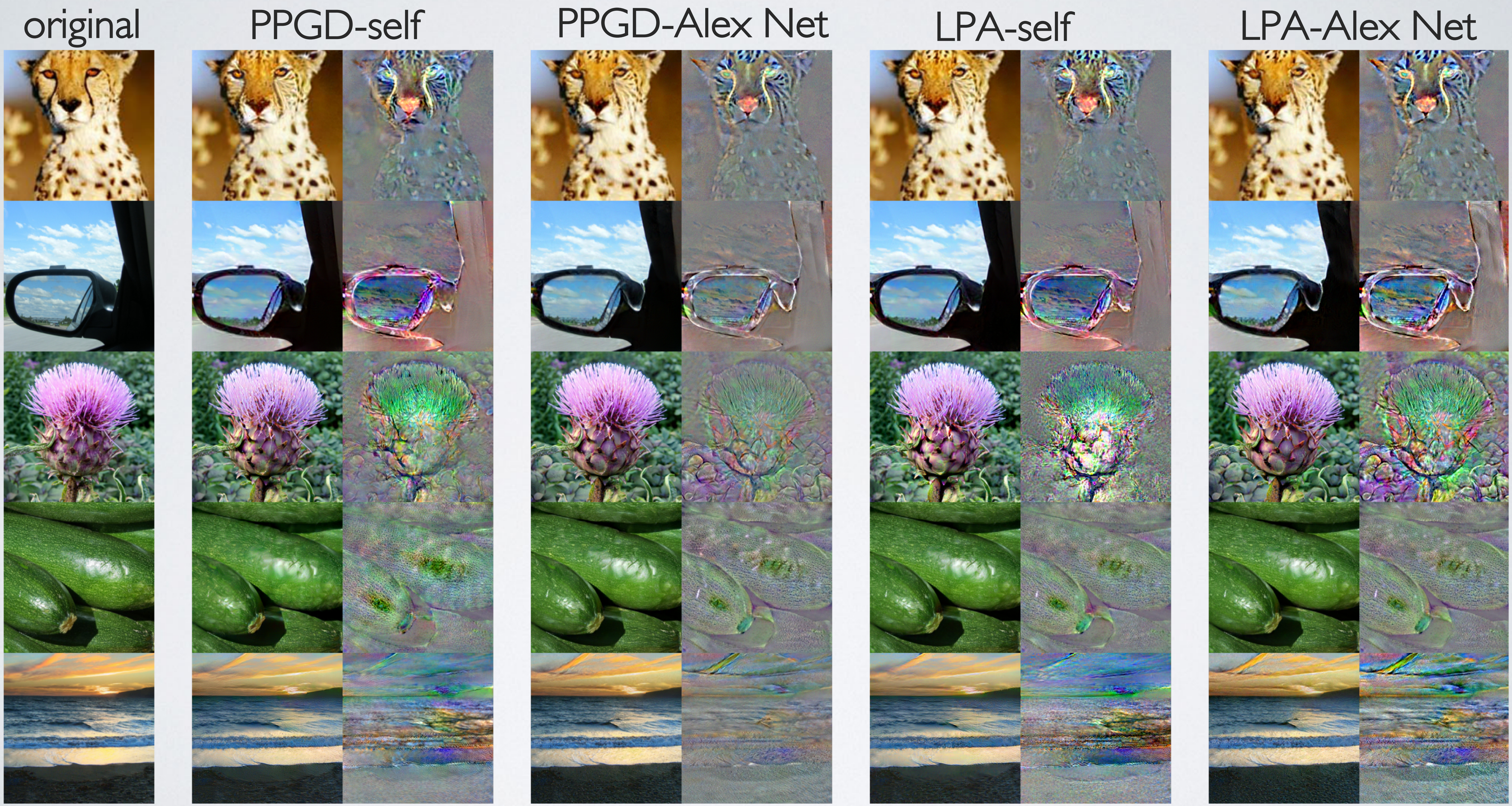
Lagrangian weight

- Similar in spirit to the Carlini & Wagner attack
- We perform a search on the Lagrangian weight λ : start with a small value of λ; if the solution is outside of the desired perceptual distance, increase λ.

LPA is the strongest adversarial attack against various types of AT-based defenses.

Example Attacks by LPA-self

Example Attacks



PAT: Perceptual Adversarial Training

- PAT optimization:

$$\min_{\theta} \; \mathbb{E}_{(\mathbf{x},y)} \left[\max_{\mathbf{x}'} \ell_{cls} \left(f_\theta(\mathbf{x}'), y \right) \right]$$

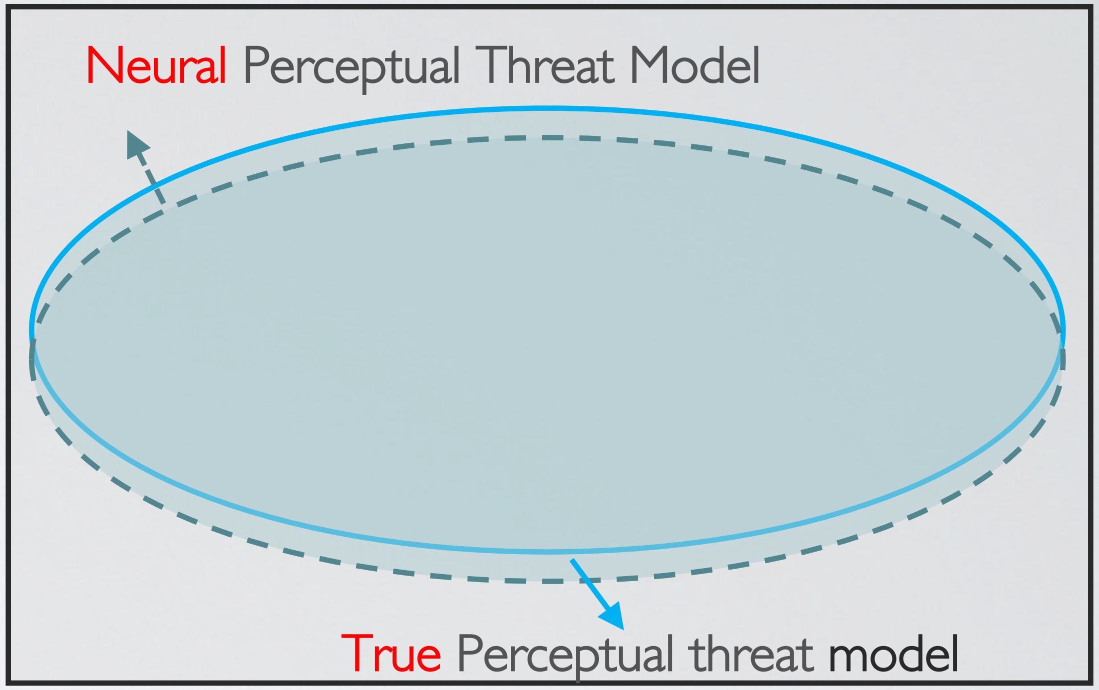
$$d_{\text{neural}}(\mathbf{x}, \mathbf{x}') = \|\phi(\mathbf{x}) - \phi(\mathbf{x}')\| \leq \rho$$

- Self-bounded PAT: perceptual and classification networks are the same ($f = g$) → neural perceptual distance changes during the training as the classifier is optimized
- Externally-bounded PAT: the neural perceptual network is pre-trained

- The inner maximization is solved using a fast variant of LPA attack (without search over the Lagrangian weight)

Perceptual Evaluation

- We study approximation power of neural perceptual distances via human evaluations



- Evaluation pipeline:
  - Adversarial examples generated using different attacks on ImageNet-100
  - Each pair is shown to an AMT participant for 2 secs
  - Perceptibility of the attack: the proportion of pairs for which participants are correct

Perceptual Evaluation

Instructions show/hide

Please carefully examine the two photos that will be displayed one after another. The photos may be the same or they may be slightly different.

Your task is to determine whether the images are the same or different.

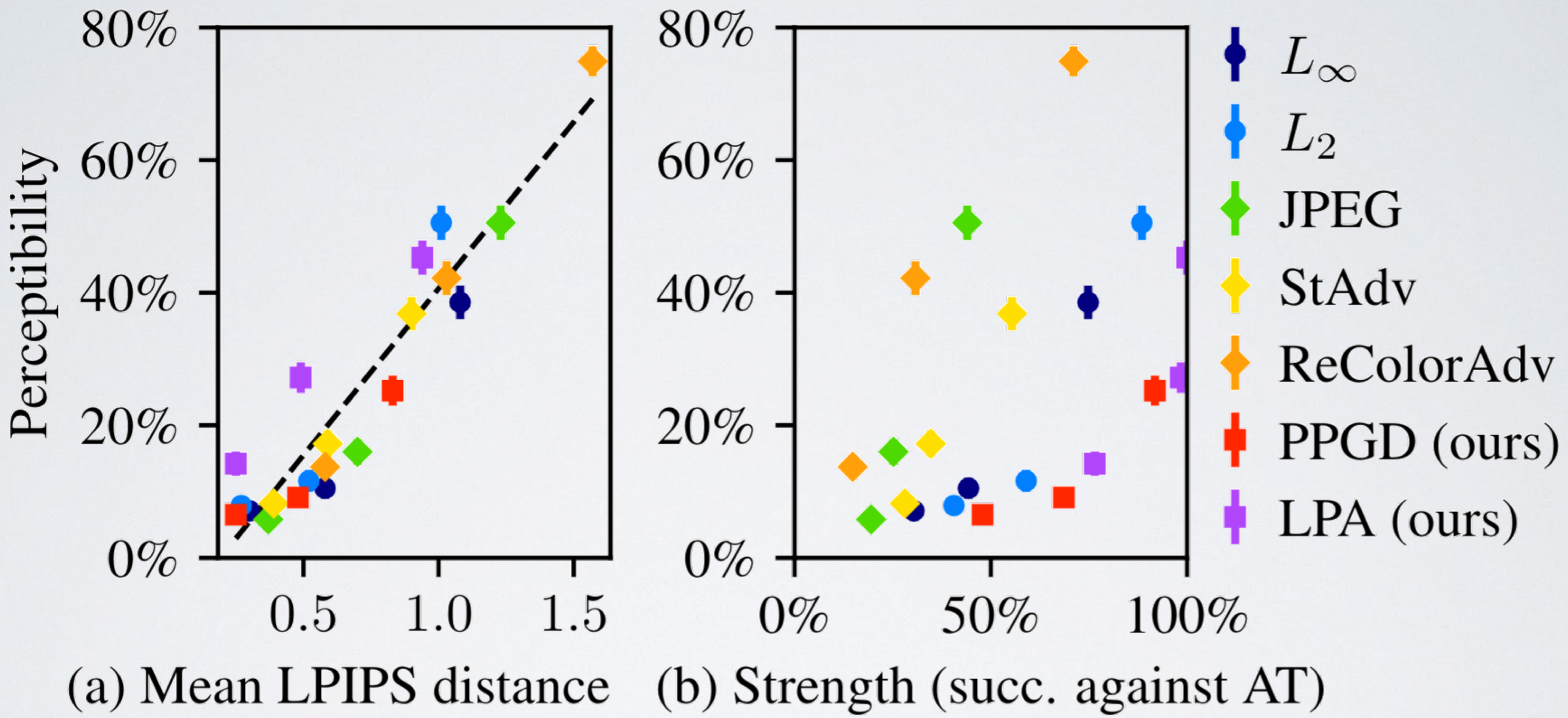
You will receive $0.01 per pair of images you examine.

Only submit up to 20 of these HITs. Any additional HITs after the first 20 will be rejected.

Image pair 1/25

Click continue to view the next pair of images.

Attack Perceptibility vs. LPIPS distance



The attack perceptibility correlates well with the neural perceptual distance

Results on CIFAR-10

- Attack bounds are 8/255 for L_∞, one for L_2, and the original bounds for StAdv/ReColorAdv.

Training	Union	Unseen mean	Narrow threat models					NPTM	
			Clean	L_∞	L_2	StAdv	ReColor	PPGD	LPA
Normal	0.0	0.1	94.8	0.0	0.0	0.0	0.4	0.0	0.0
AT L_∞	1.0	19.6	86.8	49.0	19.2	4.8	54.5	1.6	0.0
TRADES L_∞	4.6	23.3	84.9	52.5	23.3	9.2	60.6	2.0	0.0
AT L_2	4.0	25.3	85.0	39.5	47.8	7.8	53.5	6.3	0.3
AT StAdv	0.0	1.4	86.2	0.1	0.2	53.9	5.1	0.0	0.0
AT ReColorAdv	0.0	3.1	93.4	8.5	3.9	0.0	65.0	0.1	0.0
AT all (random)	0.7	—	85.2	22.0	23.4	1.2	46.9	1.8	0.1
AT all (average)	14.7	—	86.8	39.9	39.6	20.3	64.8	10.6	1.1
AT all (maximum)	21.4	—	84.0	25.7	30.5	40.0	63.8	8.6	1.1
PAT-self	21.9	45.6	82.4	30.2	34.9	46.4	71.0	13.1	2.1
PAT-AlexNet	**27.8**	**48.5**	71.6	28.7	33.3	64.5	67.5	**26.6**	**9.8**

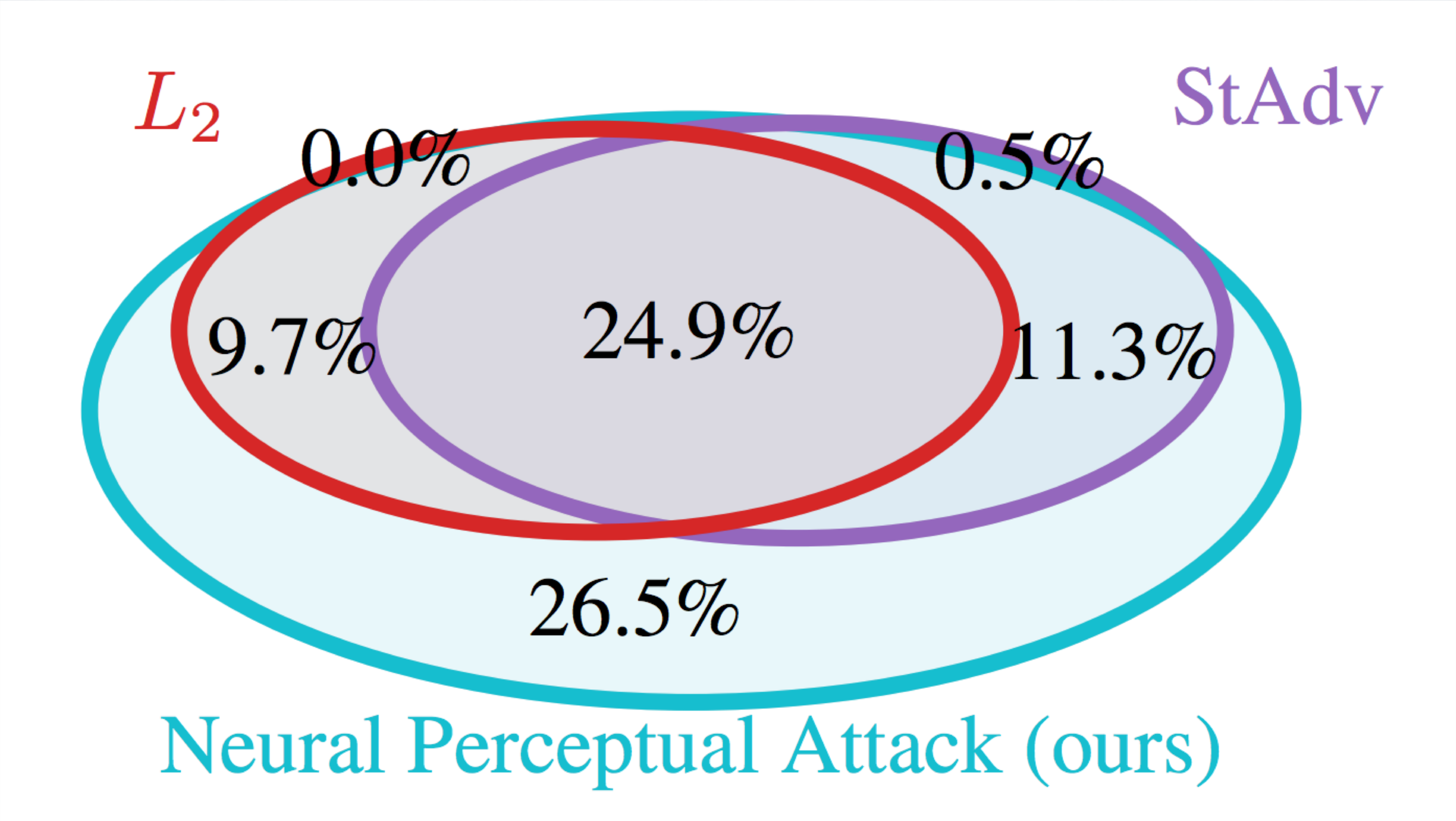
Our method has high Unforeseen Attack Robustness

Results on ImageNet-100

Training	Union	Unseen mean	Narrow threat models						NPTM	
			Clean	L_∞	L_2	JPEG	StAdv	ReColor	PPGD	LPA
Normal	0.0	0.1	89.1	0.0	0.0	0.0	0.0	2.4	0.0	0.0
L_∞	0.5	11.3	81.7	55.7	3.7	10.8	4.6	37.5	1.5	0.0
L_2	12.3	31.5	75.3	46.1	41.0	56.6	22.8	31.2	22.0	0.5
JPEG	0.1	7.4	84.8	13.7	1.8	74.8	0.3	21.0	0.5	0.0
StAdv	0.6	2.1	77.1	2.6	1.2	3.7	65.3	2.9	0.6	0.0
ReColorAdv	0.0	0.1	90.1	0.2	0.0	0.1	0.0	69.3	0.0	0.0
All (random)	0.9	—	78.6	38.3	26.4	61.3	1.4	32.5	16.1	0.2
PAT-self	**32.5**	**46.4**	72.6	45.0	37.7	53.0	51.3	45.1	29.2	**2.4**
PAT-AlexNet	25.5	44.7	75.7	46.8	41.0	55.9	39.0	40.8	**31.1**	1.6

Our method has high Unforeseen Attack Robustness

Results on ImageNet-100



- Each ellipse indicates a set of vulnerable examples to an attack
- The NPTM encompasses both other types of attacks and includes additional examples not vulnerable to either.

Today's Lecture

- Part I: Attack = (algorithm, threat model)
  - algorithm: variable
  - threat model: fixed

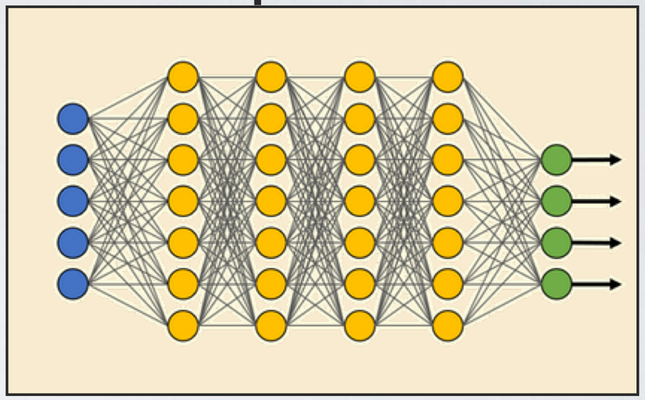
- Part II: Attack = (algorithm, threat model)
  - algorithm: variable
  - threat model: variable

Deep Learning Pipeline

Training data

Optimization

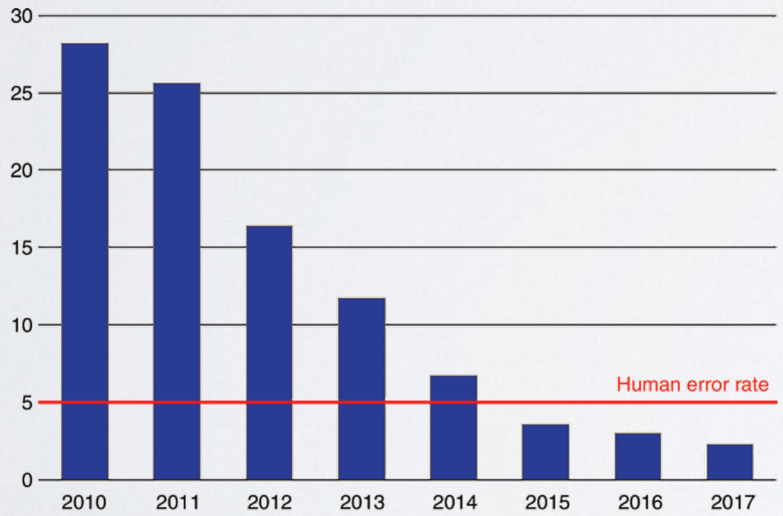
Deep model



Test data

Evaluation

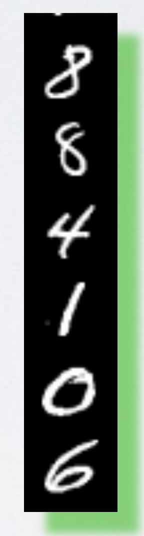
Classification error



Robustness against training time (poisoning) attacks

General Poisoning Threat Model

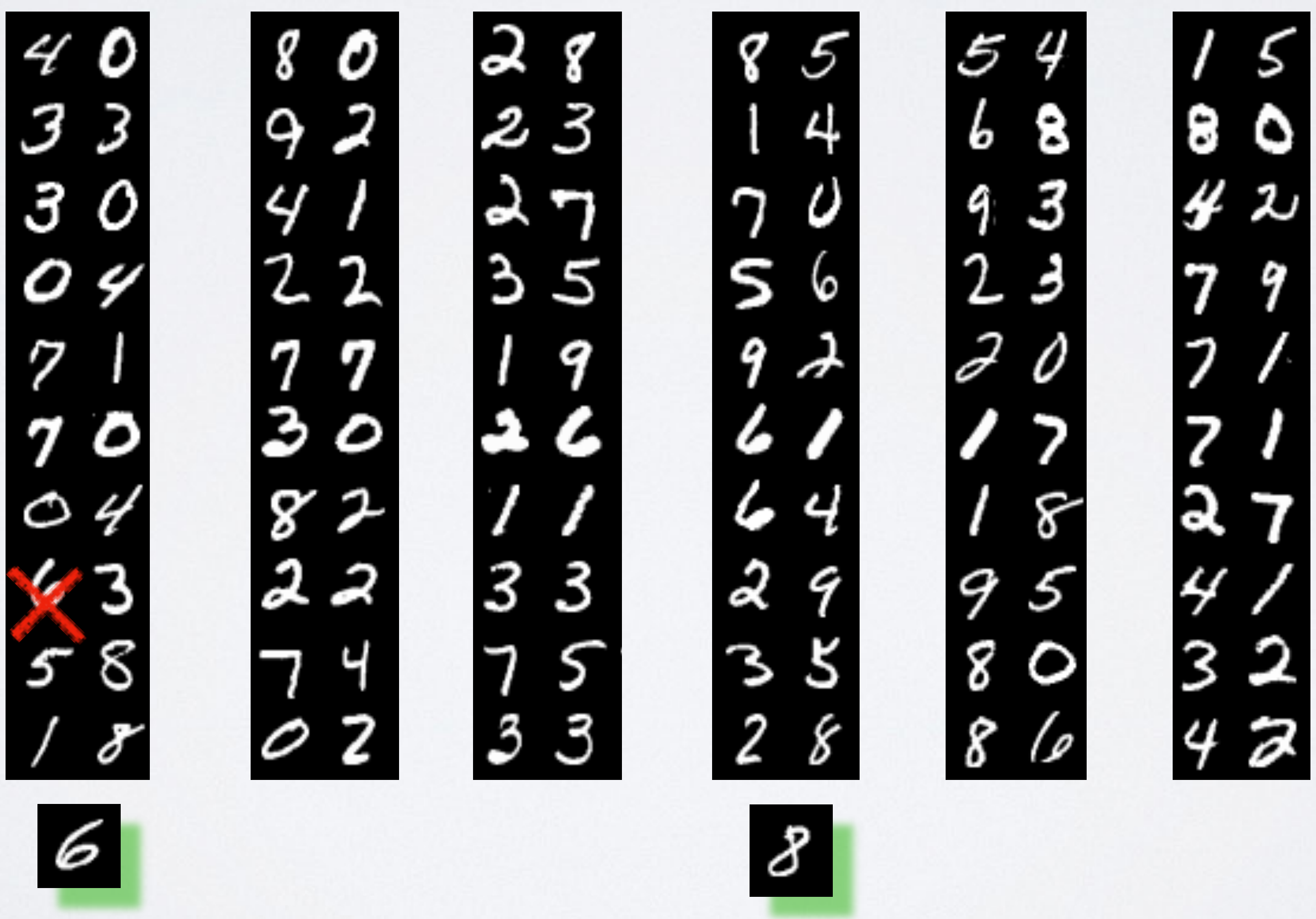
- We consider a general threat model: the attacker can insert or remove up to ρ training images
- Example (ρ = 10):



- This includes any distortion and/or label flip to a bounded number of samples

Deep Partition Aggregation (DPA)

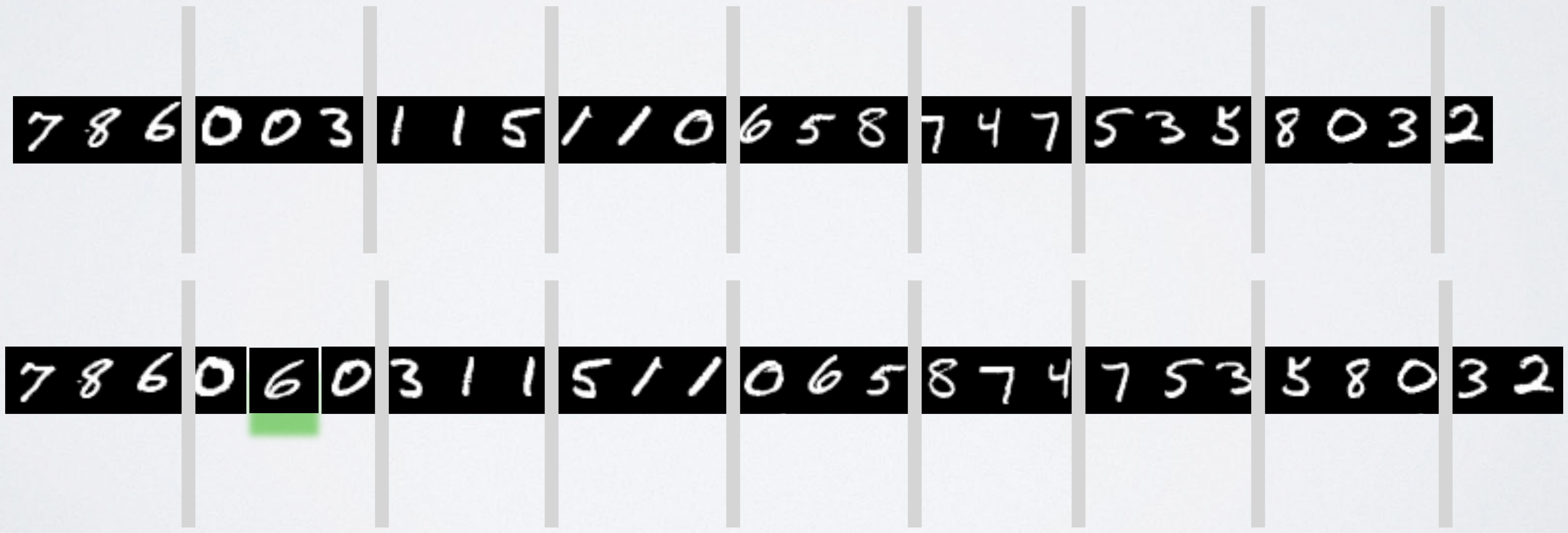
- DPA is a certified defense against general poisoning
- Idea: partition data, then train a CNN classifier on each partition. The number of partitions affected by poisoning is at most ρ → robustness certificate



Levine and F., Deep Partition Aggregation: Provable Defense against General Poisoning Attacks, ICLR 2021

Robust Partitioning for DPA

- Naive partitioning can allow for a single insertion or deletion to cause an unbounded number of base classifiers to change



Robust Partitioning for DPA

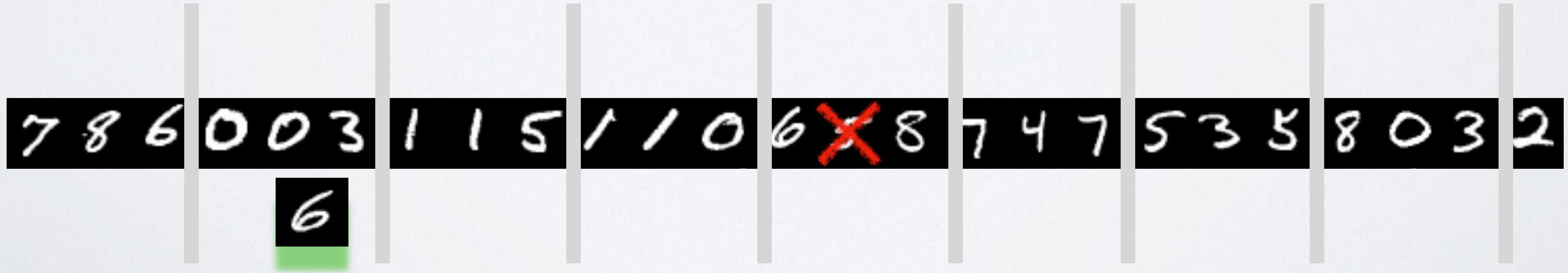
- Naive partitioning can allow for a single insertion or deletion to cause an unbounded number of base classifiers to change

- Solution: use deterministic hash functions

$$P_i := \{\boldsymbol{t} \in T \mid h(\boldsymbol{t}) \equiv i \pmod{k}\}$$

Partition i ↙ ↘ Deterministic hash

- Inserting or removing a sample only affects the one partition that it is assigned to

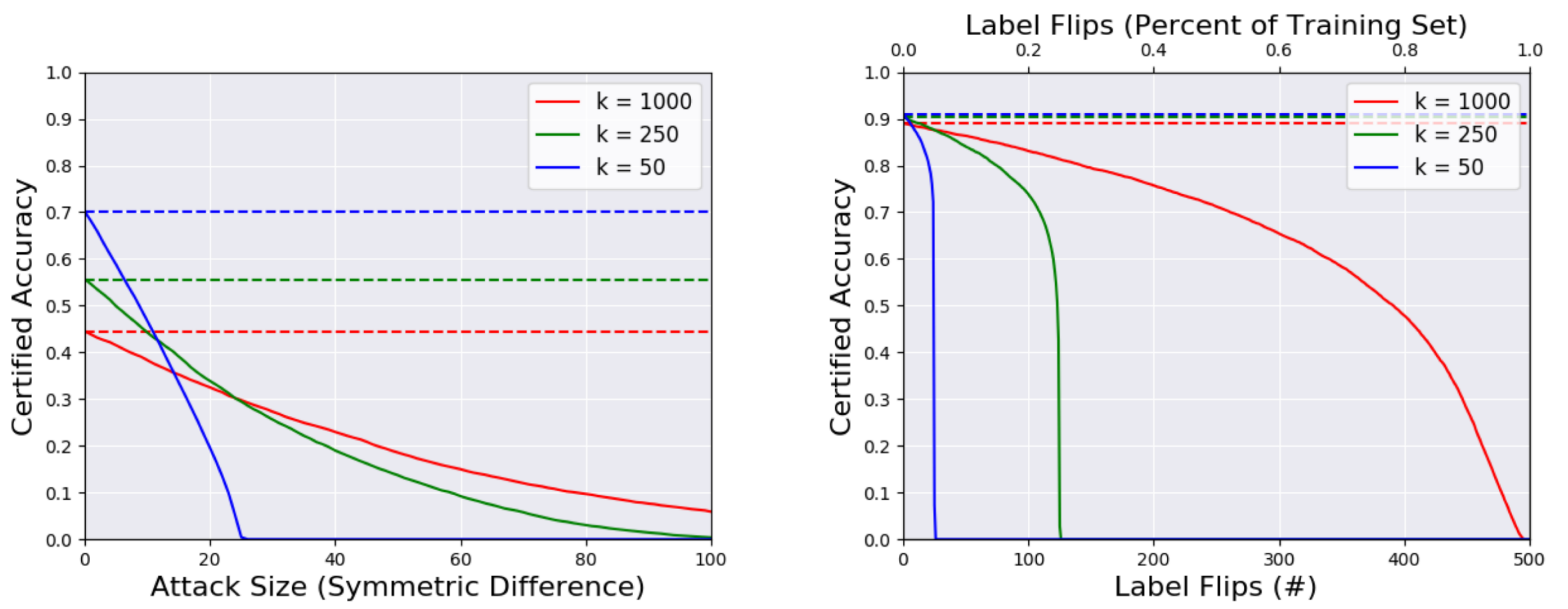


Comparison to Prior Work

- DPA is the first scheme for certified robustness for general poisoning attacks
- For label-flipping attacks, we have developed a semi-supervised DPA method that significantly outperforms the previous SOTA (Rosenfeld et al., 2020)

Empirical Results (CIFAR-10)

- Our method established new state-of-the-art results for both general and label-flipping poisoning attacks



(a) DPA (General poisoning attacks) (b) SS-DPA (Label-flipping poisoning attacks)