Adversarial Robustness against the Union of Multiple Perturbation Models

Pratyush Maini[1], Eric Wong[2], J. Zico Kolter[2,3]

https://github.com/locuslab/robust_union

[1]Indian Institute of Technology Delhi

New Delhi, India

[2]School of Computer Science

Carnegie Mellon University

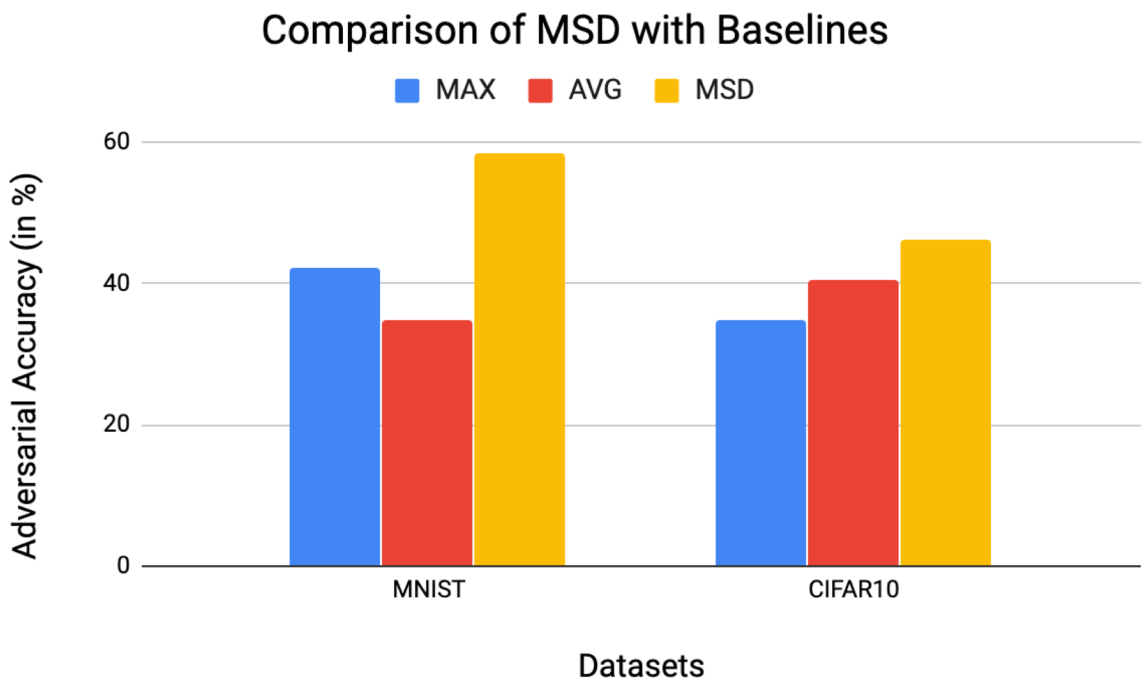
Pittsburgh, PA 15213, USA

[3]Bosch Center for Artificial Intelligence

Pittsburgh, PA 15222, USA

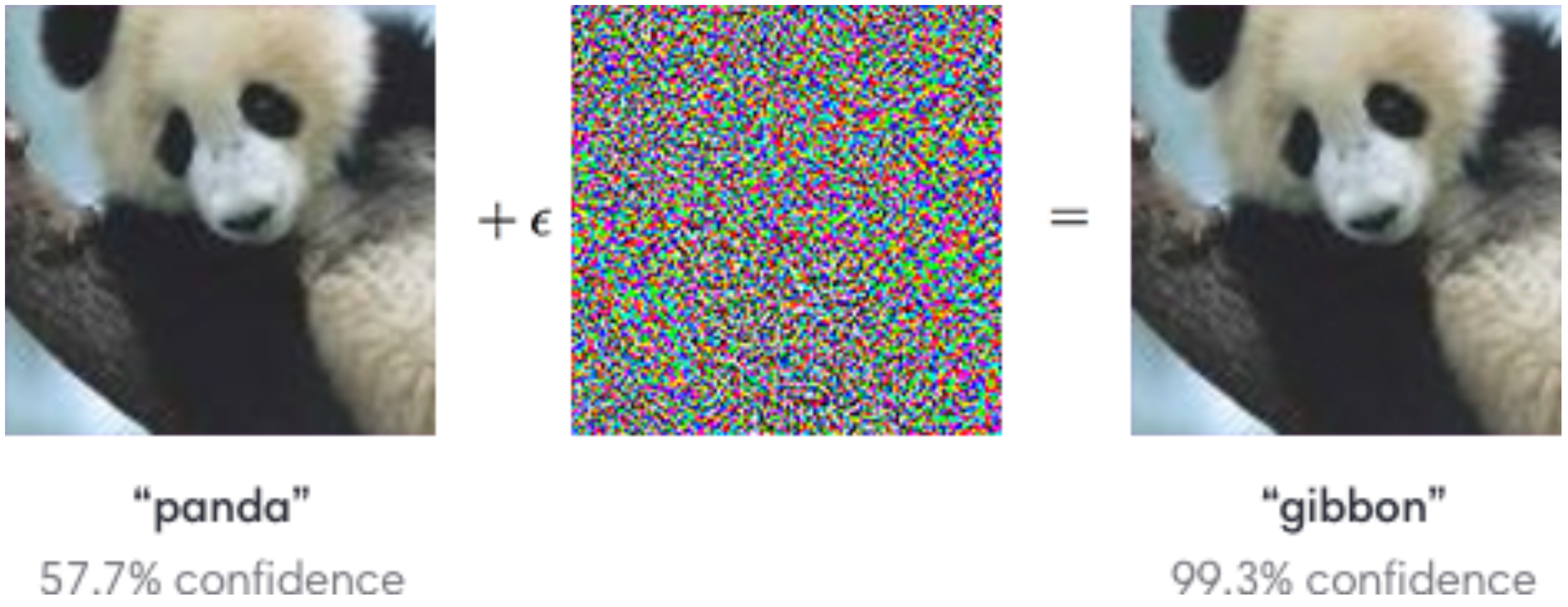
Overview

- Robustness to multiple perturbation types is non-trivial, yet important
- Prior baselines can be difficult to tune and have suboptimal trade-offs
- MSD offers consistent benefits on both MNIST and CIFAR10



Deep networks are vulnerable to adversarial attacks

Imperceptible Adversaries can fool deep networks

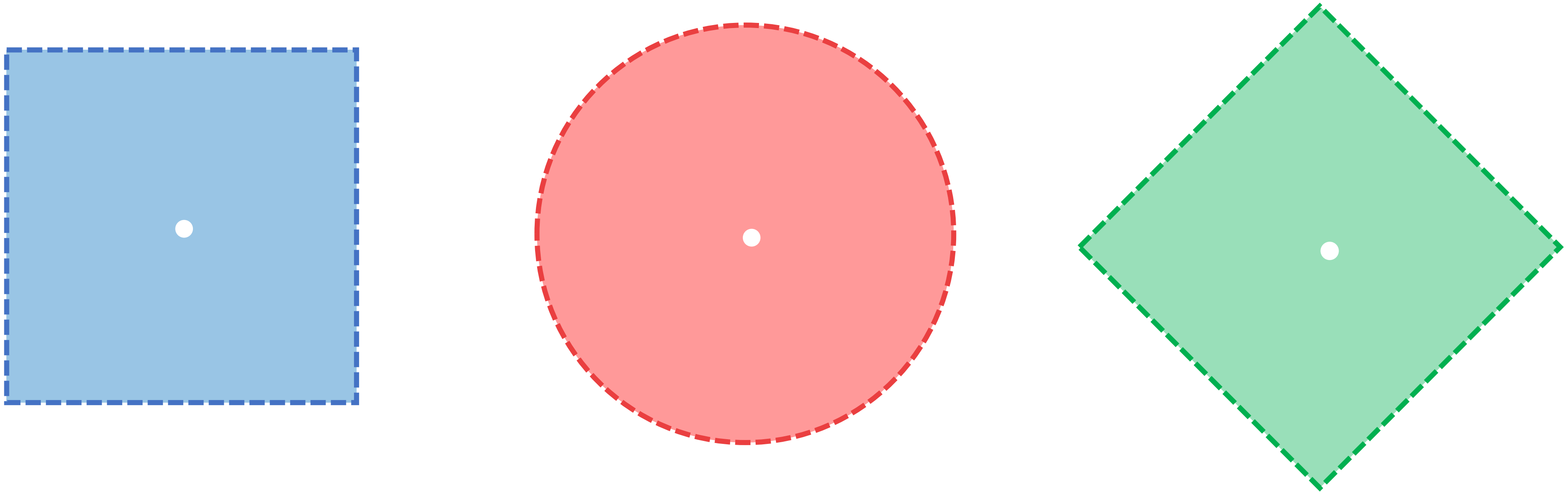


[Goodfellow et al., 2014]

The attack is staged using the *'Fast Gradient Sign Method'* which restricts an adversary within a small ℓ_∞ ball of radius ϵ_∞ around the original image

Exclusivity of different ℓ_p balls

Different perturbation types have non-overlapping regions



ℓ_∞ ball | ℓ_2 ball | ℓ_1 ball

$\max|\delta_i| \leq \epsilon_\infty$

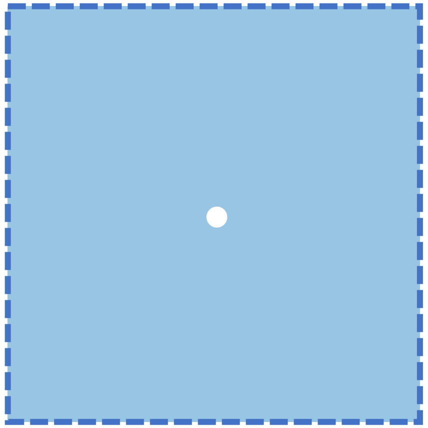
$\sqrt{\sum |\delta_i|^2} \leq \epsilon_2$

$\sum |\delta_i| \leq \epsilon_1$

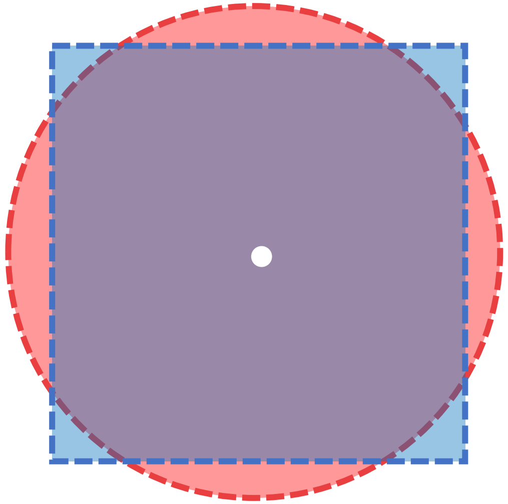
Exclusivity of different ℓ_p balls

Different perturbation types have non-overlapping regions

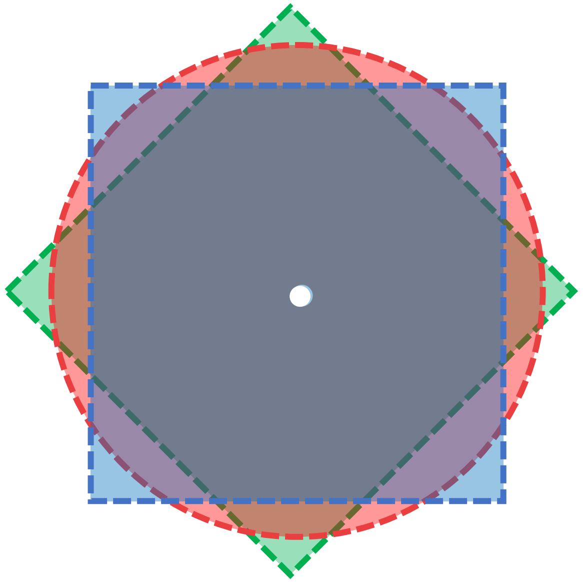
*The distinction is more significant in high-dimensional spaces



ℓ_∞ ball



ℓ_∞ ball
+
ℓ_2 ball

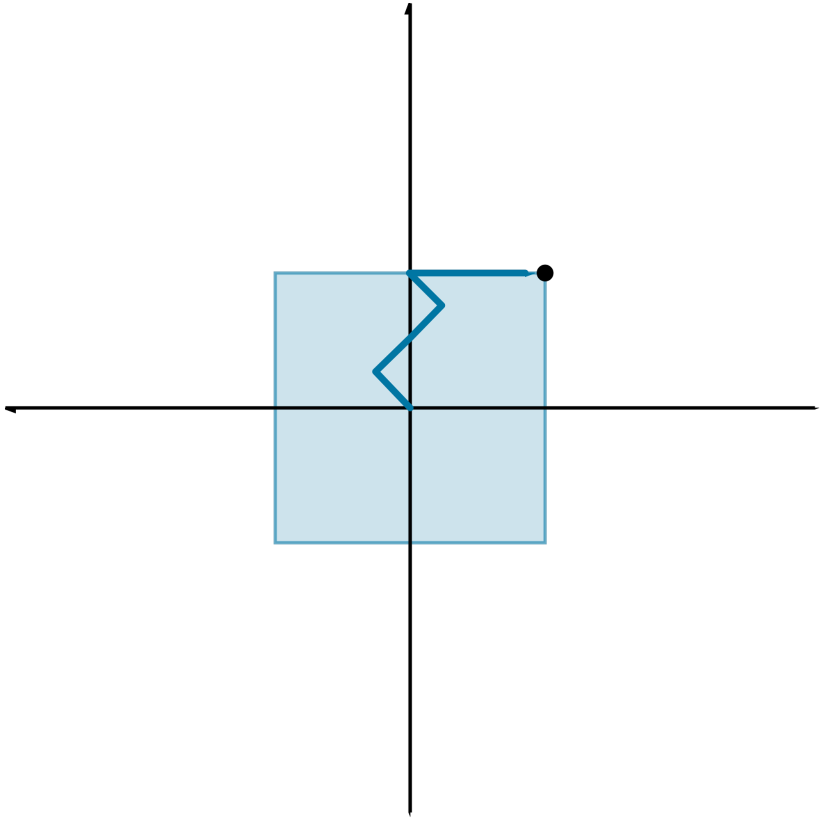


ℓ_∞ ball
+
ℓ_2 ball
+
ℓ_1 ball

PGD adversary for ℓ_∞ attacks

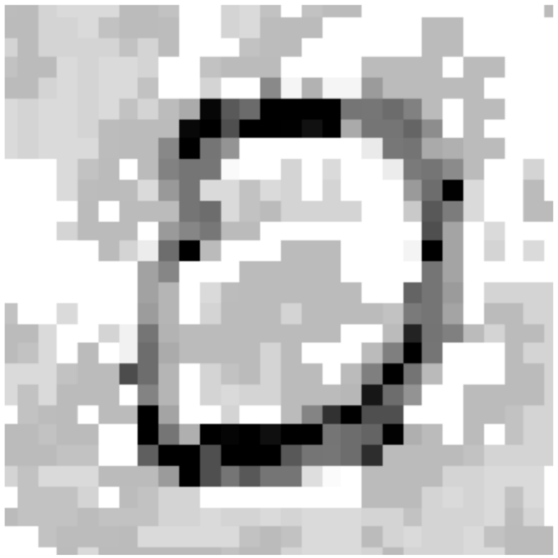
PGD ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{\theta}$) :

$\delta = 0$ *// or randomly initialized*
for $j = 1 \ldots N$:
    $\delta := \delta + \alpha \cdot \text{sign}(\nabla_\delta \ell(f_\theta(x_i + \delta), y_i))$ *// step*
    $\delta := \max(\min(\delta, \epsilon), -\epsilon)$ *// project*
end for



Adversaries confined within different ℓ_p balls have different optimal perturbations

Different perturbation types have different characteristics

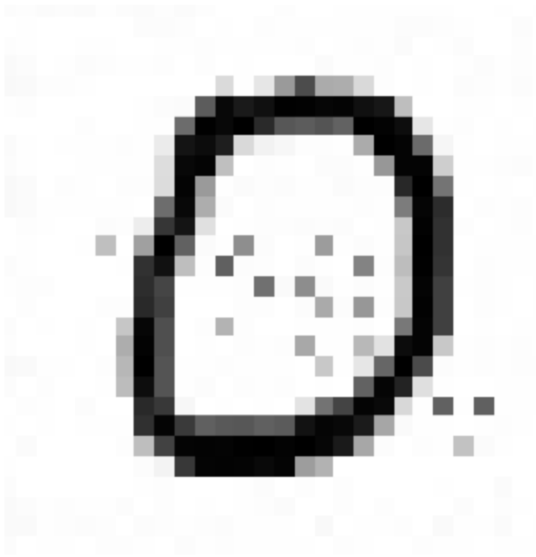


ℓ_∞ attack

$$\max |\delta_i| \leq \epsilon_\infty$$

ℓ_2 attack

$$\sqrt{\sum |\delta_i|^2} \leq \epsilon_2$$



ℓ_1 attack

$$\sum |\delta_i| \leq \epsilon_1$$

Adversarial Training

[Goodfellow et. al. 2014]

repeat :

Select minibatch $\mathcal{B}$

for $(x, y) \in \mathcal{B}$,

$\delta^*(x \mid y, \theta) = \textbf{PGD}(x, y, \theta)$

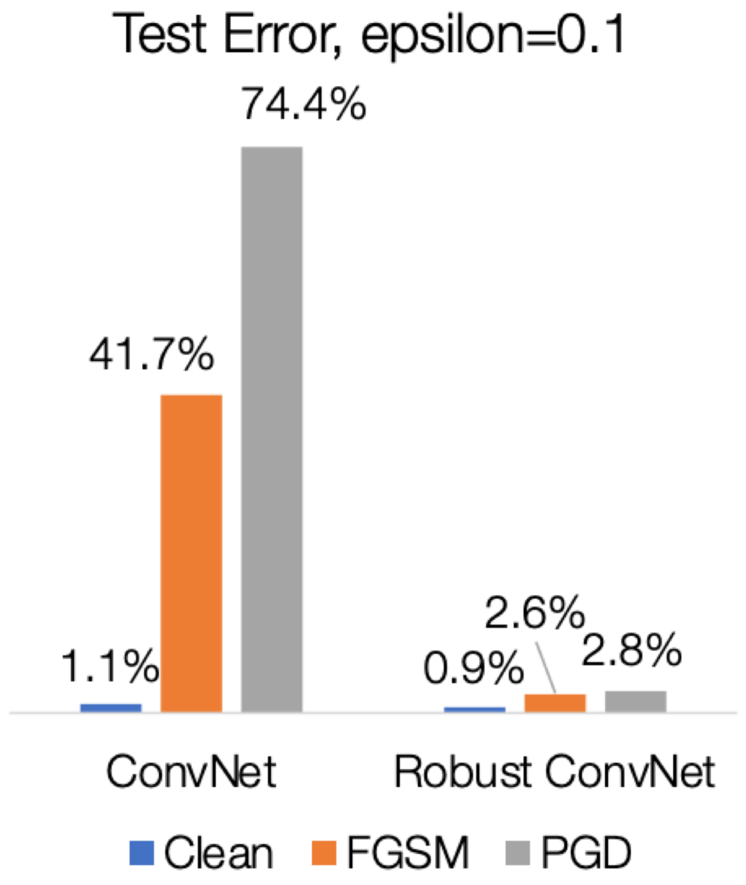
$x_{adv} = x + \delta^*(x, y, \theta)$

end for

// Update parameters

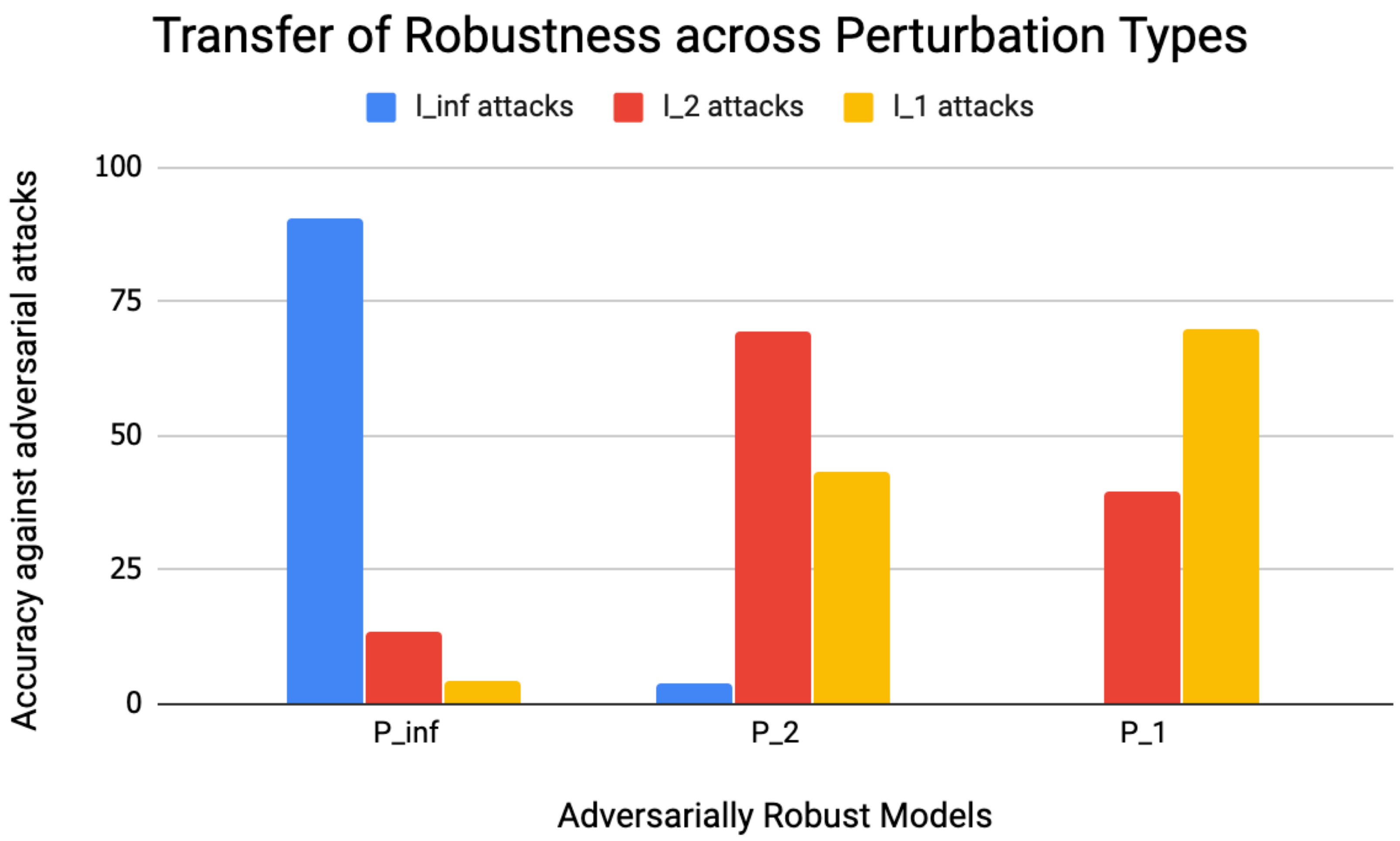
$\theta := \theta - \frac{1}{|\mathcal{B}|} \sum_{x,y \in \mathcal{B}} \nabla_\theta \ell(f_\theta(x_{adv}), y)$

until convergence



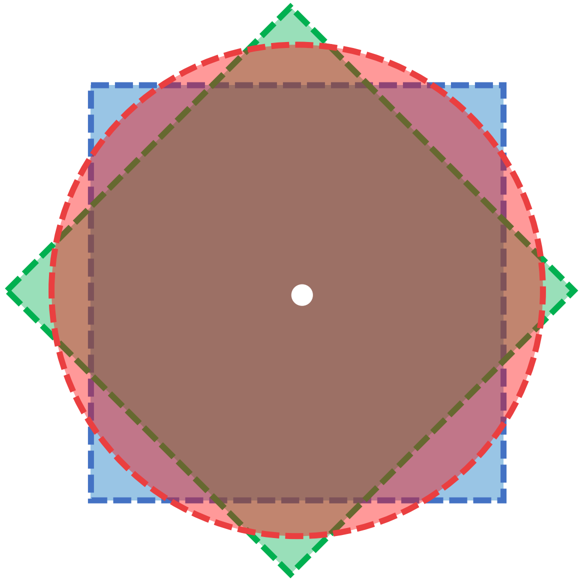
[Kolter & Madry, 2018]

Robustness does not transfer across perturbation types



Robustness against multiple perturbation types is important

- Adversaries can attack a system irrespective of the perturbation ball it was 'trained' to be robust against.
- Robustness against 'all' types of 'imperceptible' noises is essential for real world deployment.



Goal: Develop an algorithm to train a single model robust against multiple perturbation types

Naïve approaches

Let S represent a set of threat models, such that $p \in S$ corresponds to the ℓ_p threat model $\Delta_{\mathrm{p},\epsilon}$

- MAX (Worst-case Perturbation) (Tramer et. al. 2019)

$$\delta_p = \arg \max_{\delta \in \Delta_{p,\epsilon}} \ell(f_\theta(x+\delta), y) \qquad \delta^* \approx \arg\max_{\delta_p} \ell(f_\theta(x+\delta_p), y)$$

- AVG (Train over all perturbations) (Tramer et. al. 2019)

$$\min_\theta \sum_i \sum_{p \in S} \max_{\delta \in \Delta_{p,\epsilon}} \ell(f_\theta(x_i+\delta), y)$$

While the naïve approaches work to some extent, they converge to suboptimal local minima and are difficult to tune.

Multi Steepest Descent

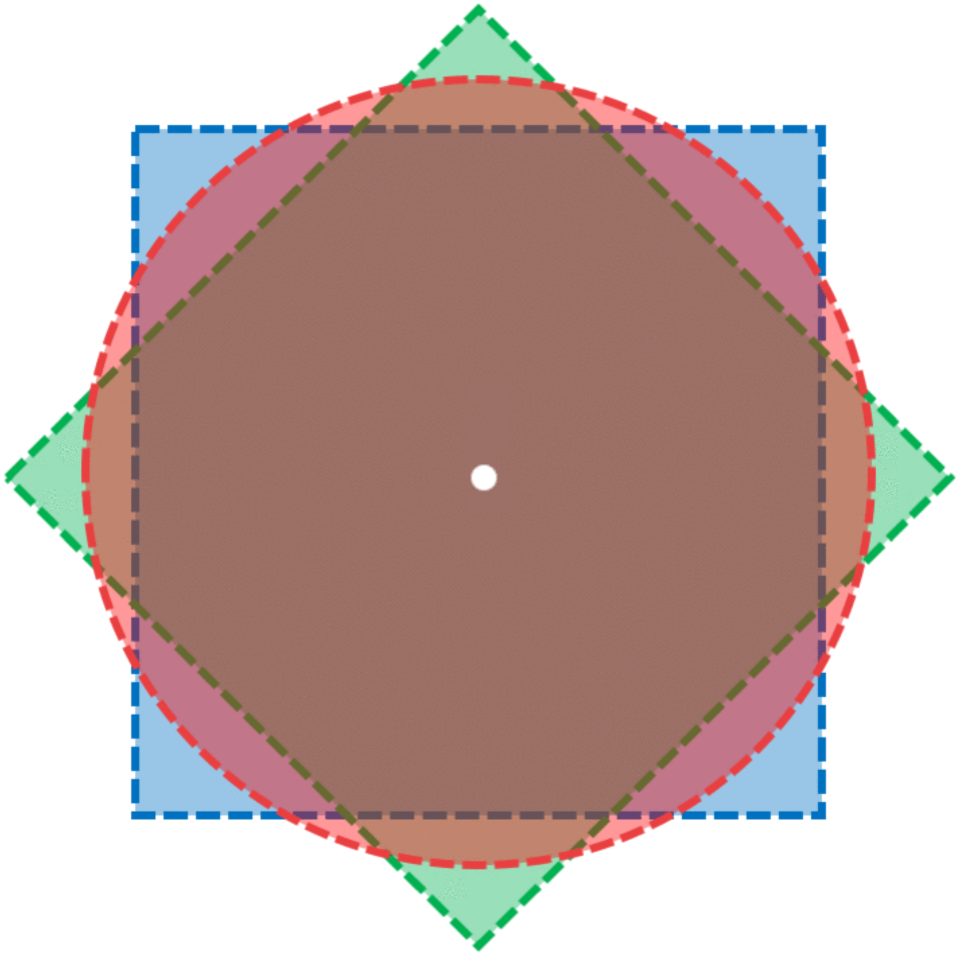
MSD ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{\theta}$) :

$\delta = 0$ *// or randomly initialized*
for $j = 1 \ldots N$:
    for $p \in \{1, 2, \infty\}$:
        $\delta_p =$ **step−and−project** $(\delta, x, y, p; \theta)$
    end for
    $\delta = \text{argmax}_{\delta_p} \ell(f_\theta(x + \delta_p), y)$
end for

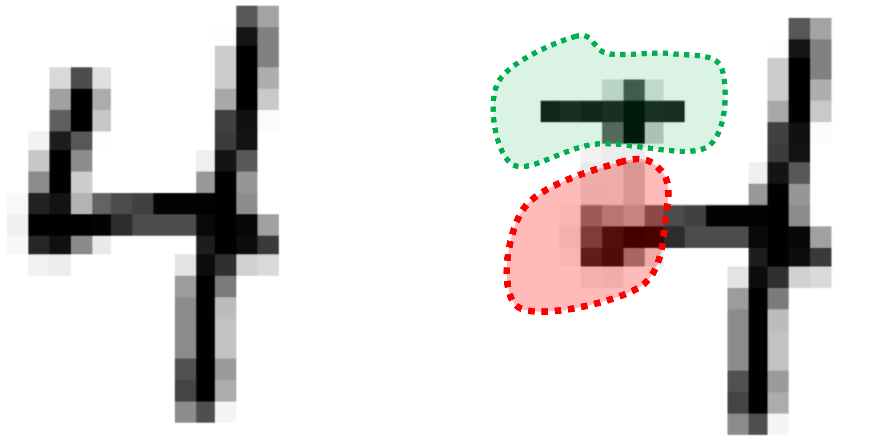
Multi Steepest Descent

MSD ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{\theta}$) **:**

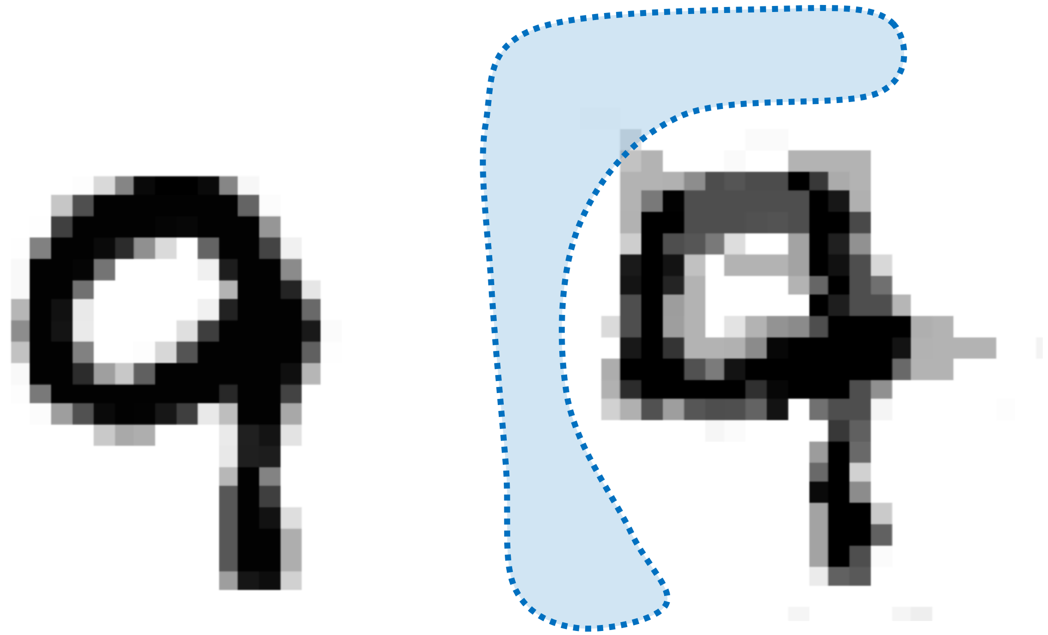
$\delta = 0$ *// or randomly initialized*
for $j = 1 \dots N$:
    for $p \in \{1, 2, \infty\}$:
        $\delta_p =$ **step−and−project** $(\delta, x, y, p; \theta)$
    end for
    $\delta = \text{argmax}_{\delta_p} \ell(f_\theta(x + \delta_p), y)$
end for



How do MSD attacks look



Original Adversarial



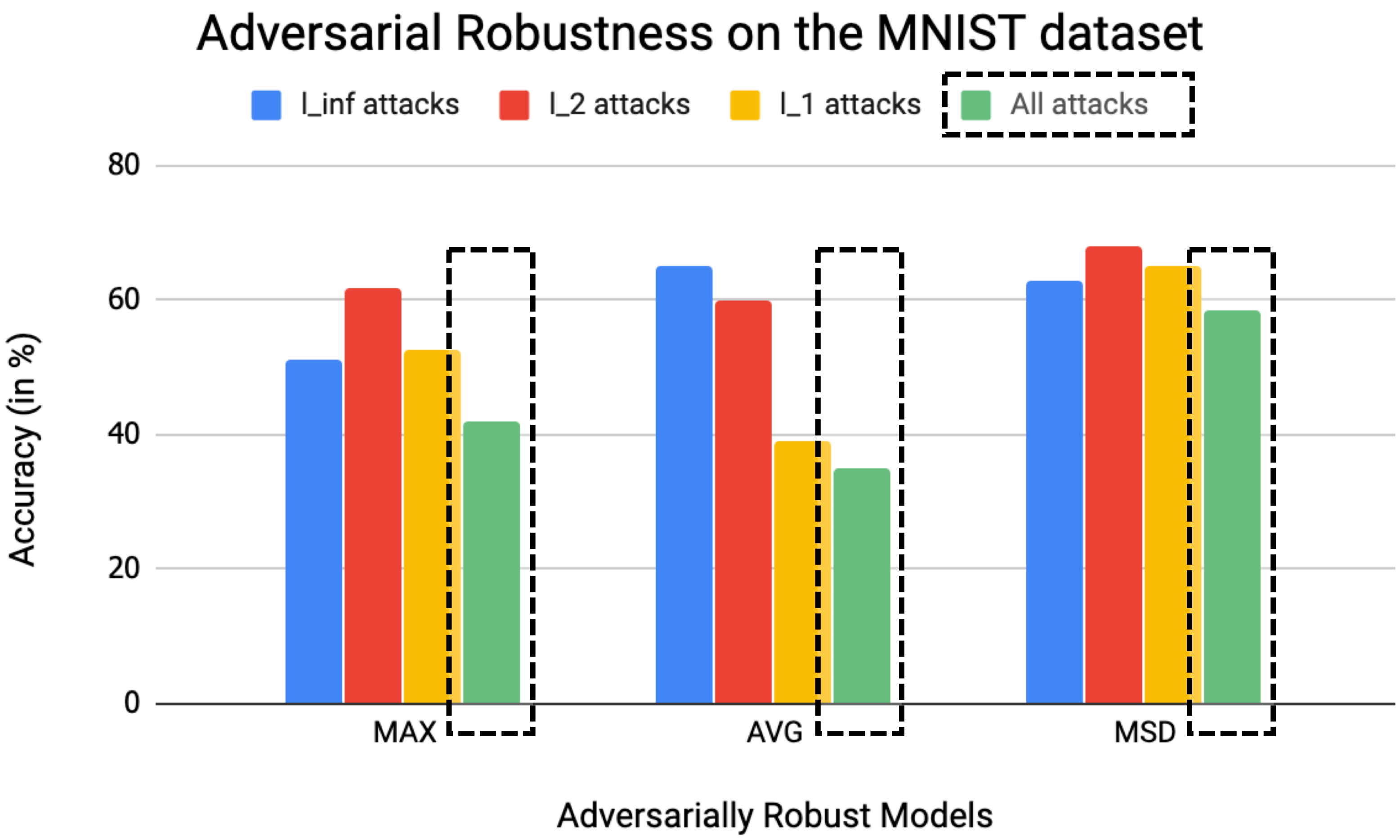
Original Adversarial

MSD is significantly more robust on MNIST

- Evaluation is performed over a wide-suite of 15 gradient-based and gradient-free attacks
- MSD significantly improves over naïve approaches on the MNIST dataset.

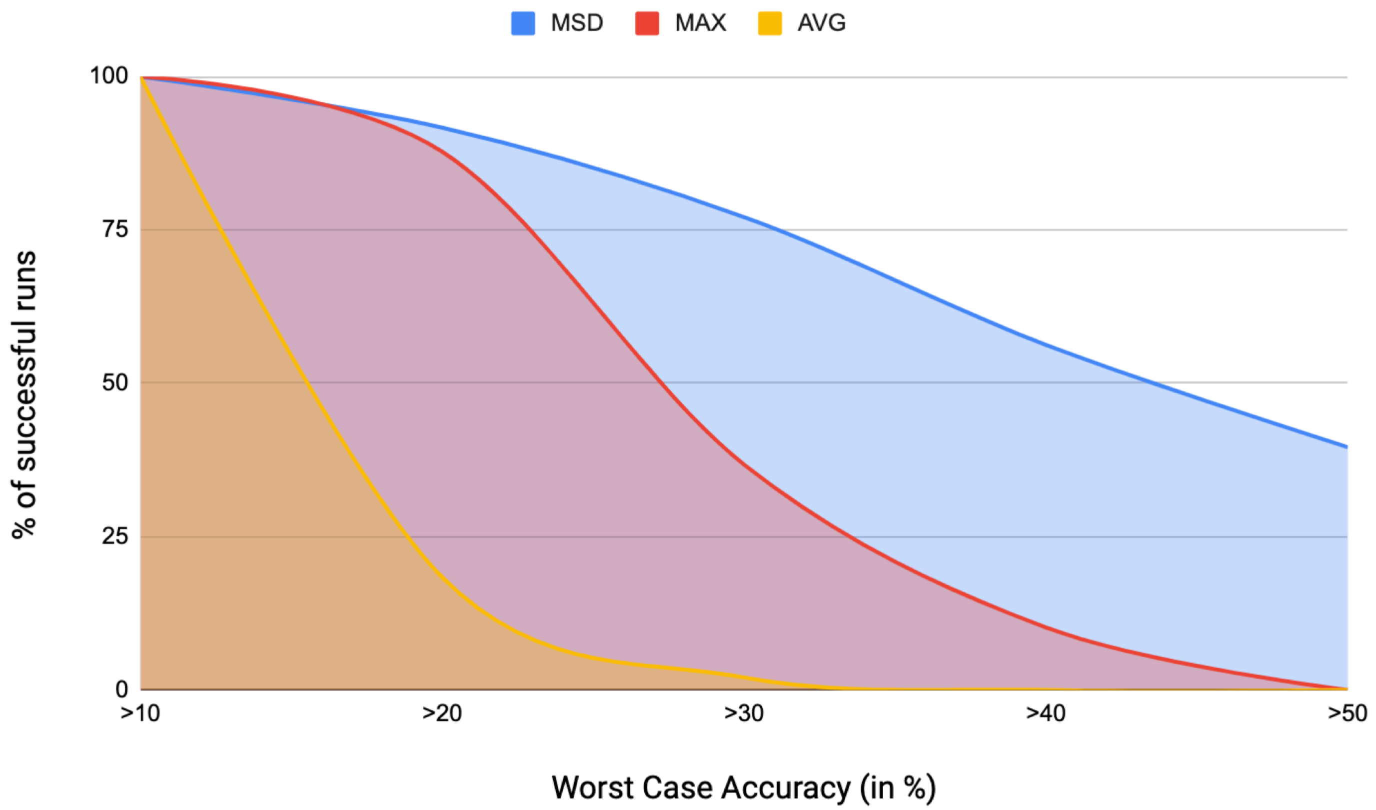
Gradient-based Attacks	Gradient-free Attacks
Fast Gradient Sign Method	Salt & Pepper Attack
Projected Gradient Descent	Pointwise Attack
Momentum Iterative Method	Gaussian Noise Attack
DeepFool Attack	Boundary Attack
DDN Attack	
C&W Attack	

MSD is significantly more robust on MNIST



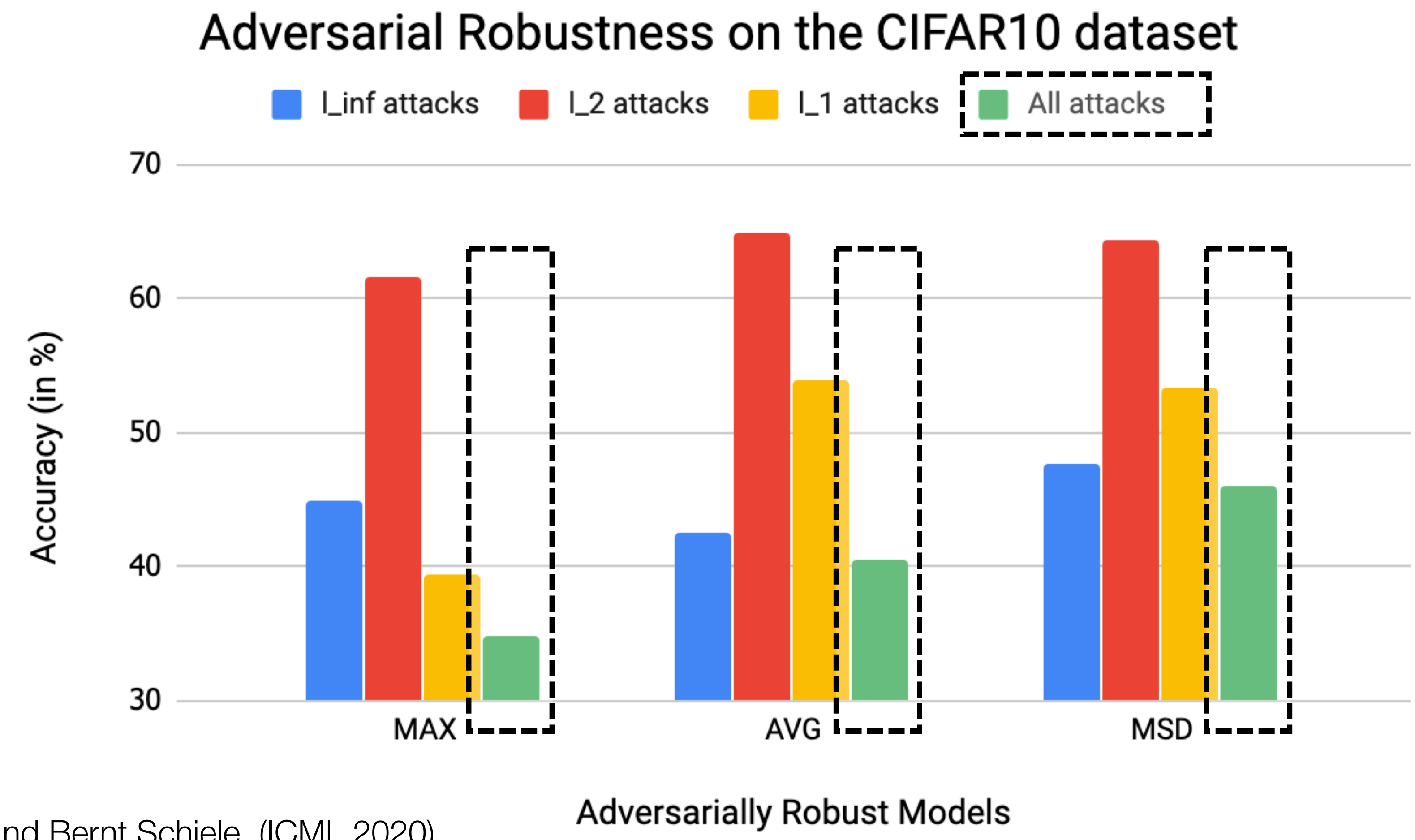
MSD is less sensitive to hyperparameter changes

The algorithm is much more stable to train and does not require any heuristic adjustments for different datasets unlike previous work.



MSD improves over previous baselines on CIFAR10

- The results on both MNIST and CIFAR10 have been reproduced.[1]



[1]David Stutz, Matthias Hein and Bernt Schiele. (ICML 2020) Confidence-Calibrated Adversarial Training: Generalizing to Unseen Attacks

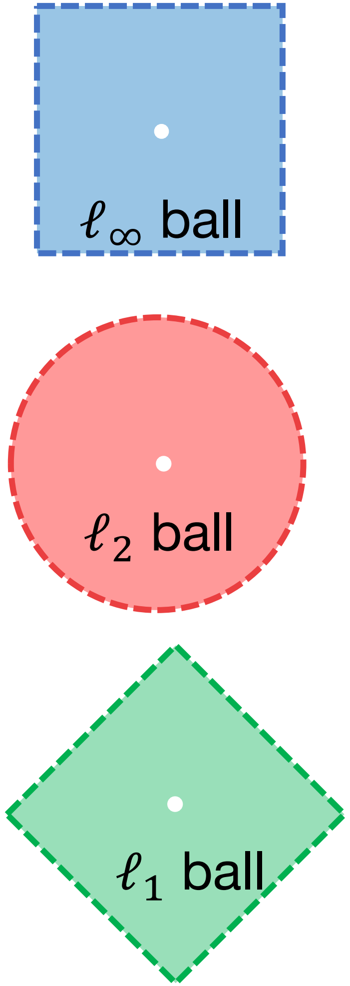
Conclusions from multiple perturbation adversarial training

- PGD training can be extended to make models robust to multiple perturbation types
- Naïve approaches
  - Can be highly variable (across parameters and datasets)
  - Are difficult to tune
  - Converge to suboptimal local minima

- MSD consistently outperforms them across both MNIST and CIFAR10

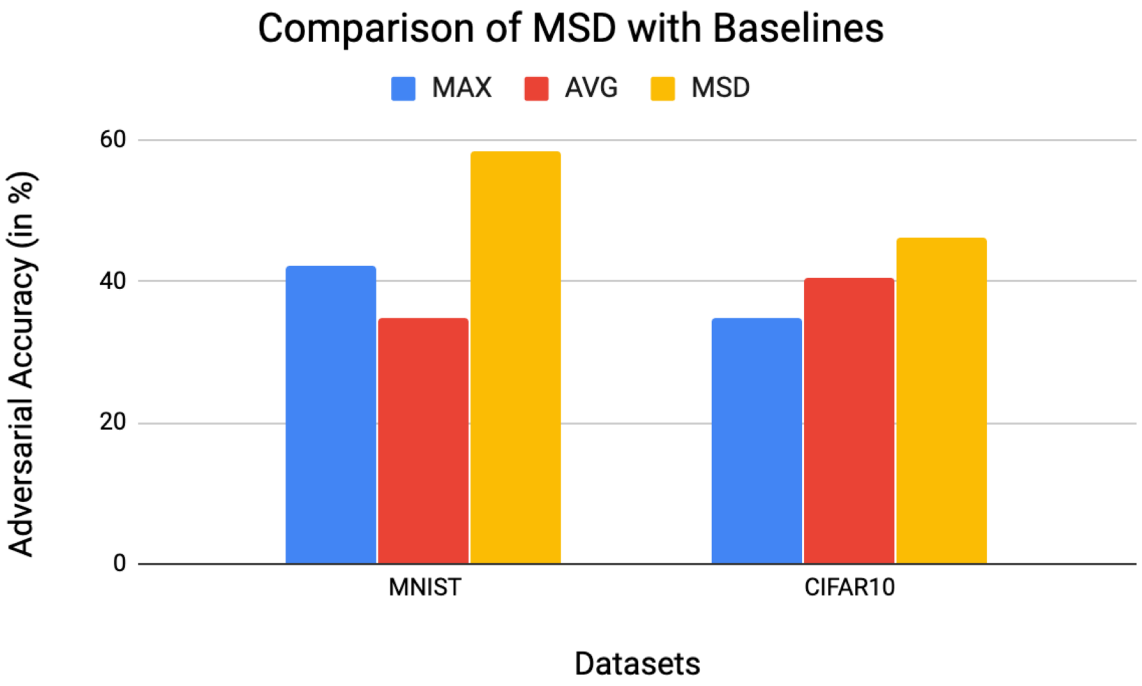
https://github.com/locuslab/robust_union

Different perturbation types have non-overlapping regions



Overview

- Robustness to multiple perturbation types is non-trivial, yet important
- Prior baselines can be difficult to tune and have suboptimal trade-offs
- MSD offers consistent benefits on both MNIST and CIFAR10



MSD $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$**:**

$\delta = 0$ *// or randomly initialized*
for $j = 1 \ldots N$:
  for $p \in \{1, 2, \infty\}$:
  $\delta_p =$ **step−project** $(\delta, x, y, p; \theta)$
  end for
  $\delta = \text{argmax}_{\delta_p}\, \ell(f_\theta(x + \delta_p), y)$
end for