Trustworthy AI Autonomy

M1-3: Adversarial Robustness

Ding Zhao

Assistant Professor

Carnegie Mellon University

Carnegie Mellon University

Safe AI Lab @CMU

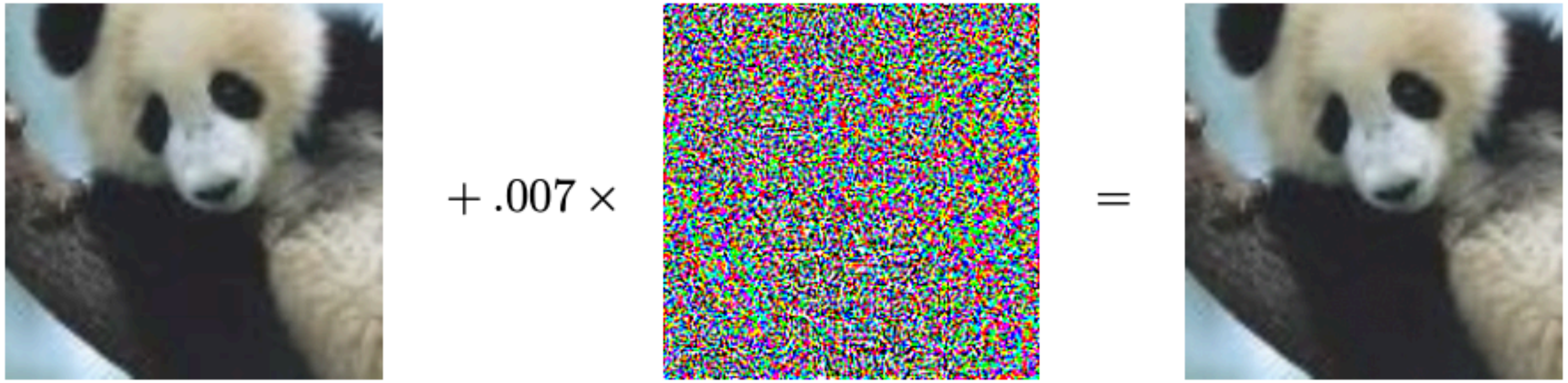
Plan for today

- Why should we care about adversarial learning?
- Adversarial attack approaches
    - Poisoning
    - Evasion: Fast Gradient Sign Method (FGSM)
    - *Case study*: adversarial examples in self-driving
- Adversarial defense approaches
    - Building trustworthy models

Adversarial examples

- **Adversarial examples**: inputs that are **specially made** by adding **small perturbation** to original inputs to **fool classifiers**

> At OpenAI, we think adversarial examples are a good aspect of security to work on because they represent a concrete problem in AI safety that can be addressed in the short term, and because fixing them is difficult enough that it requires a serious research effort. (Though we'll need to explore many aspects of machine learning security to achieve our goal of building safe, widely distributed AI.)

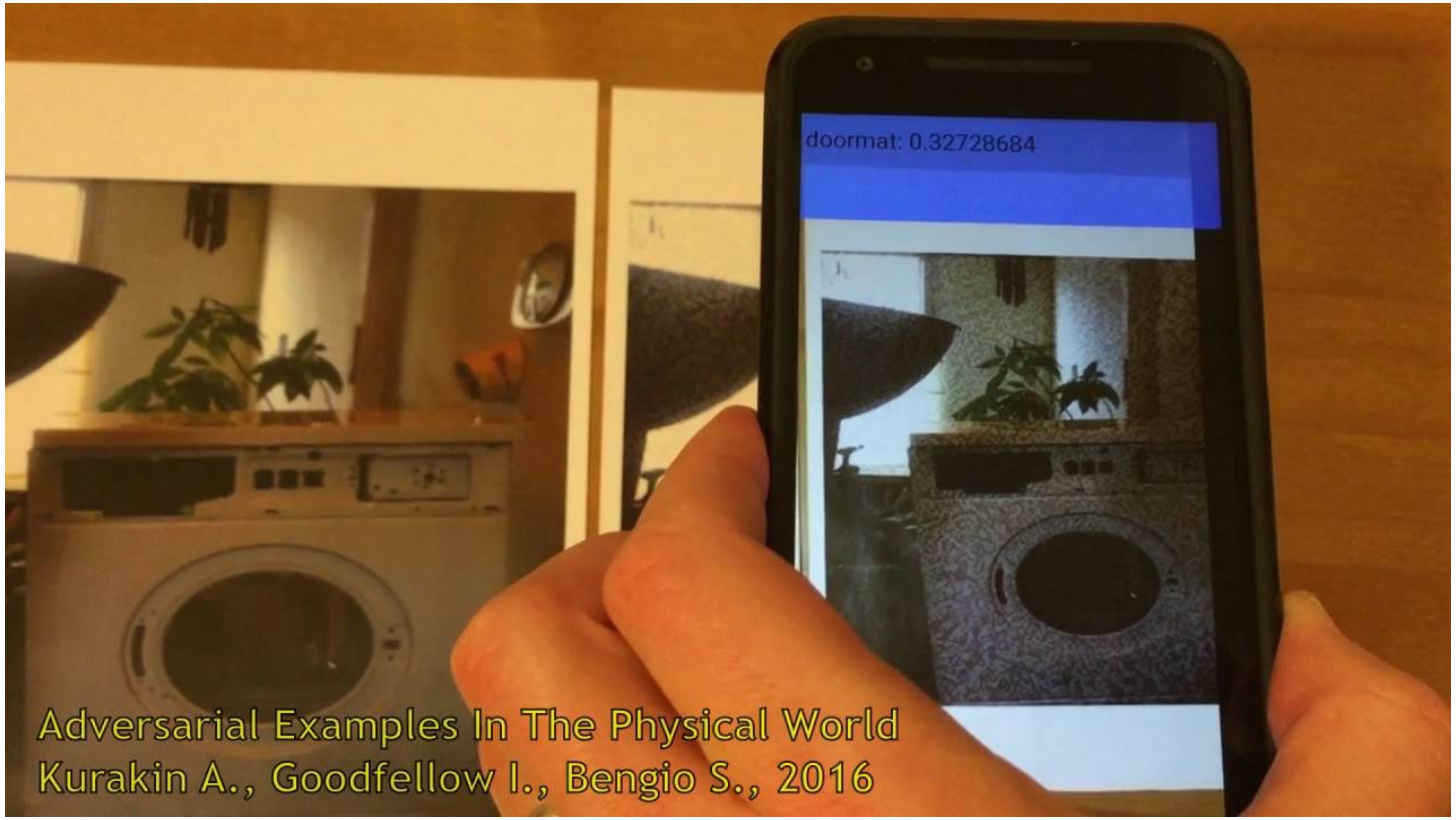


x	δ	$x + \epsilon \cdot \delta$
$\hat{y}$ = "panda" 57.7% confidence	small perturbation	$\hat{y}$ = "gibbon" 99.3% confidence

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

Adversarial examples

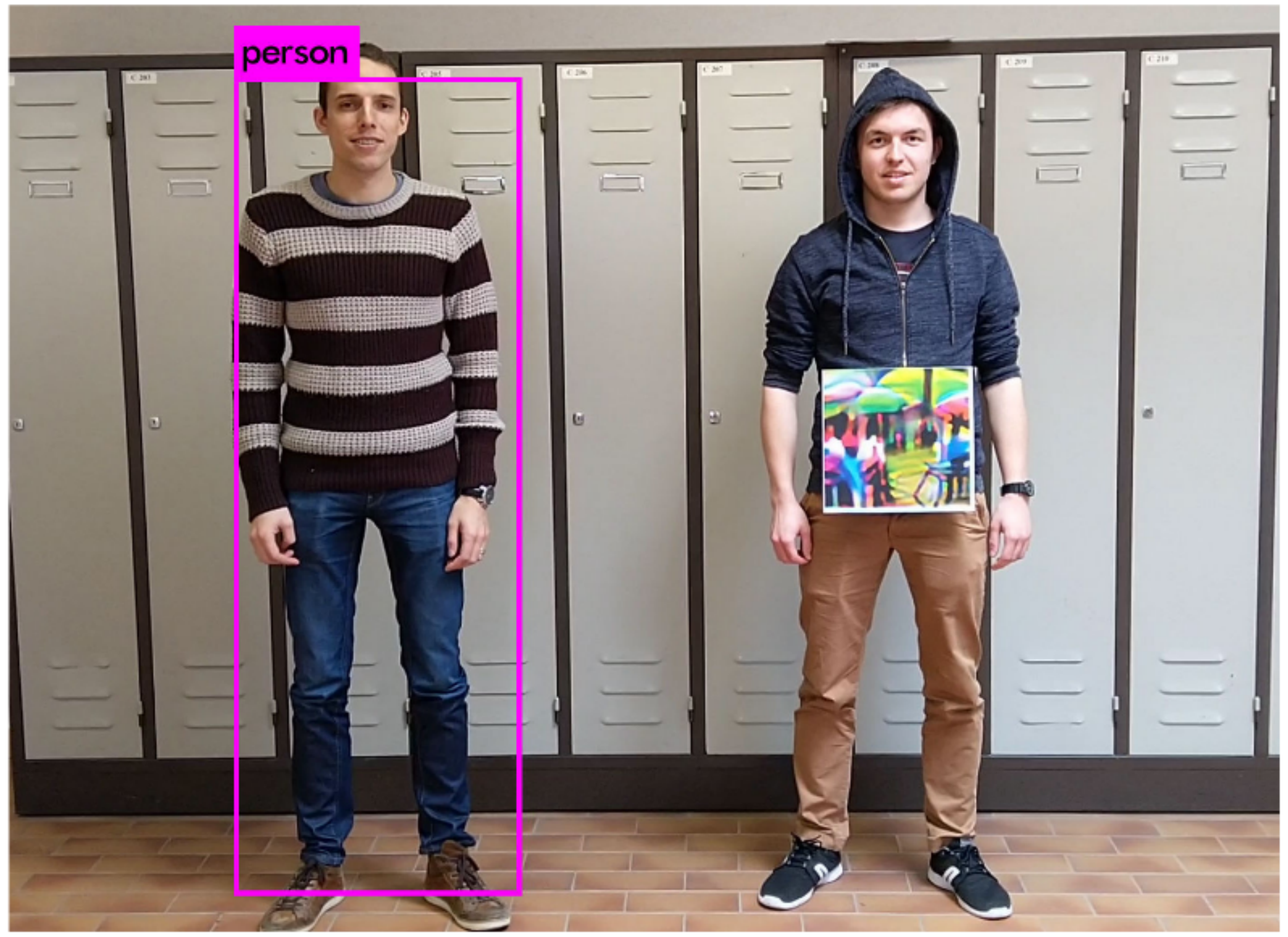
- Adversarial examples can be applied physically and are robust



Credits: https://www.youtube.com/watch?v=zQ_uMenoBCk

Adversarial examples

- Adversarial examples pose significant real-world threats

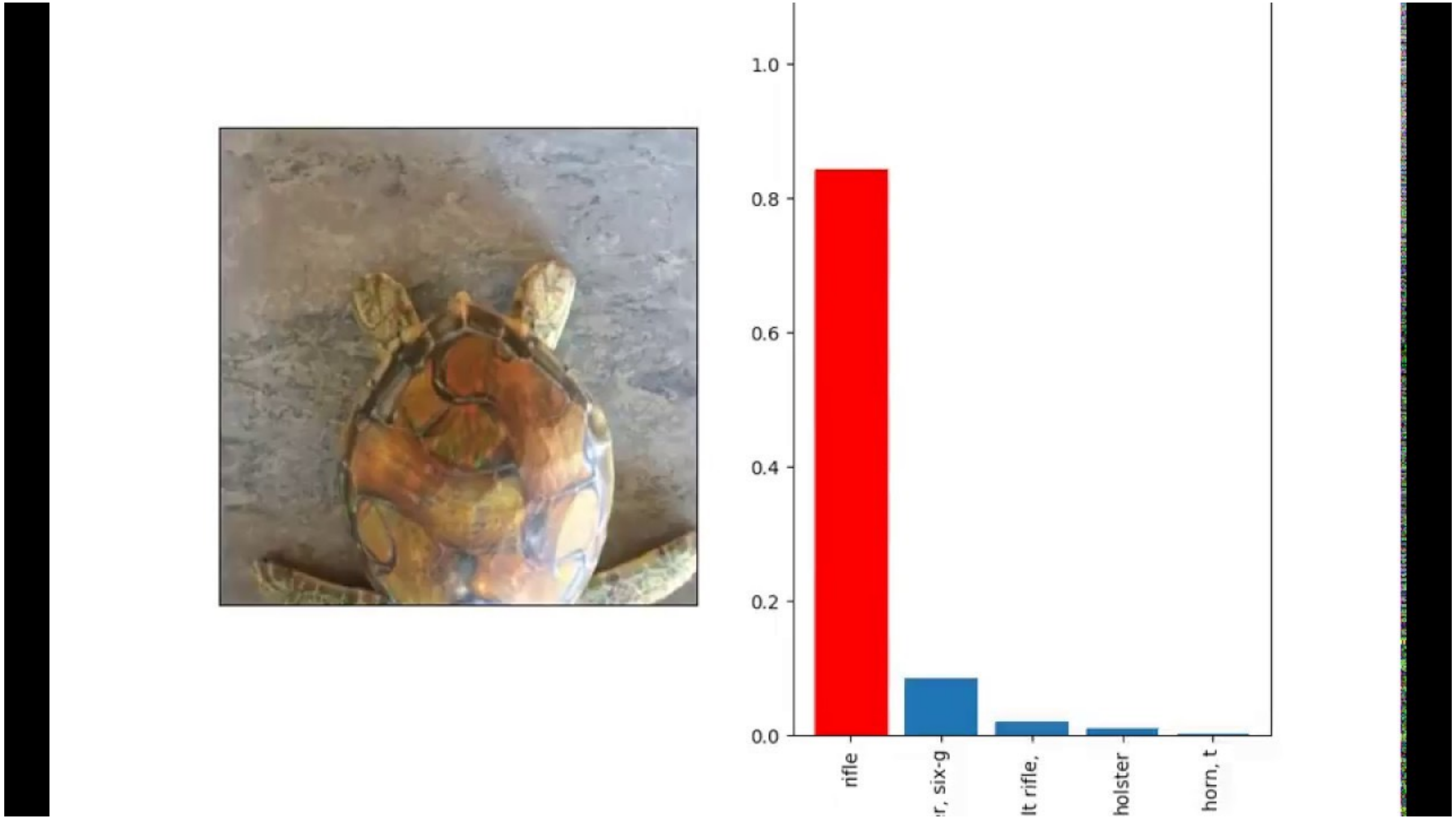


The adversarially added patches make the objects undetected by classifier (dodging)

Sharif, Mahmood, et al. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition." Proceedings of the 2016 acm sigsac conference on computer and communications security. 2016.
Ya-guan, Q. I. A. N., et al. "Spot Evasion Attacks: Adversarial Examples for License Plate Recognition Systems with Convolutional Neural Networks." Computers & Security (2020): 101826.
Thys, Simen, Wiebe Van Ranst, and Toon Goedemé. "Fooling automated surveillance cameras: adversarial patches to attack person detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019.

Adversarial examples

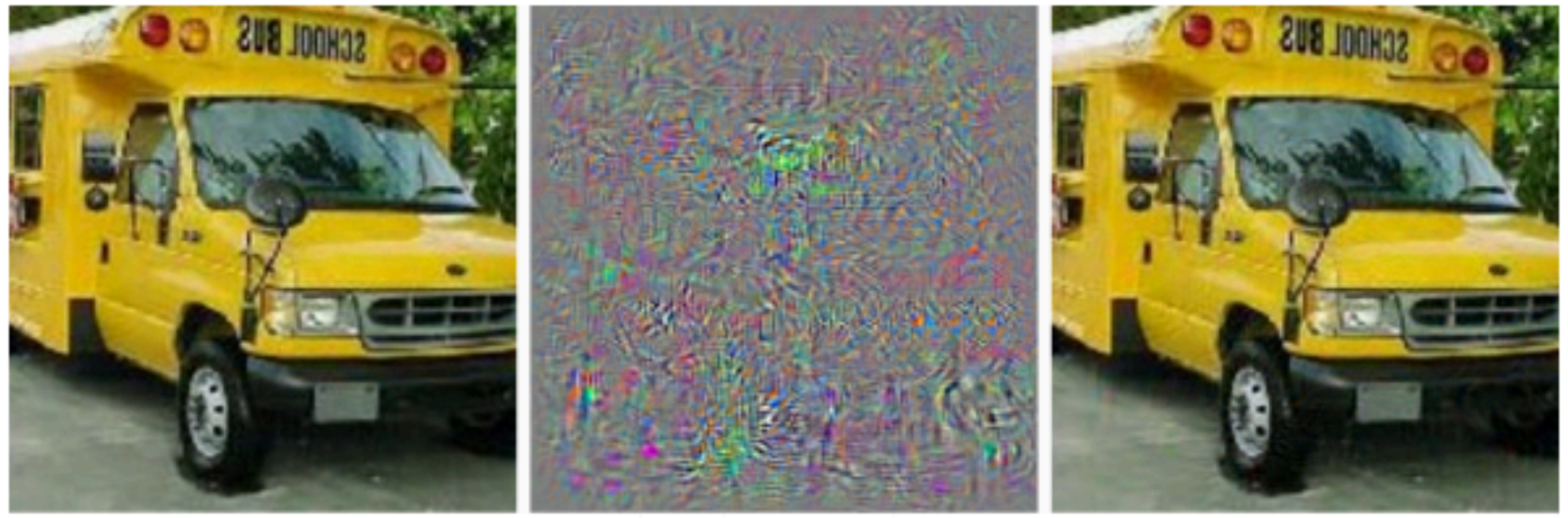
Source: https://youtube.com/watch?v=YXy6oX1iNoA



Adversarial examples can also change prediction (impersonation / falsification)

Adversarial examples

- Adversarial examples pose significant real-world threats



label: bus prediction: ostrich label: stop sign prediction: - (unseen)

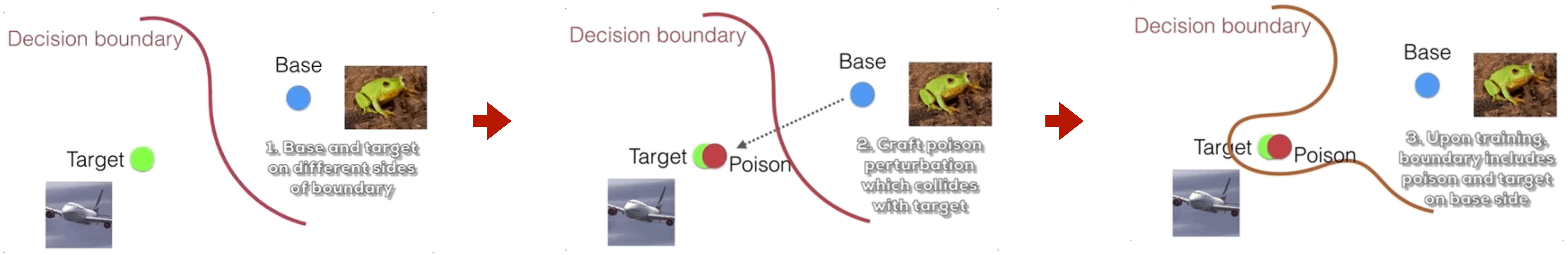
Szegedy, Christian, et al. "Intriguing properties of neural networks." arXiv preprint arXiv:1312.6199 (2013).
Eykholt, Kevin, et al. "Robust physical-world attacks on deep learning visual classification." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

Adversarial attacks

- Adversarial attack: performing actions to fool a given classifier
  - **Poisoning**: injecting “poisonous” training data to weaken the model
  - **Evasion**: finding “evaded” data points with training set and model fixed

Poisoning-type adversarial attacks

- Performed by finding data point $\tilde{x}$ that is close to the target instance t in the feature space and also close to the base instance b in input space



Credits: https://wrhuang.com/project/poison-frog/
Shafahi, Ali, et al. "Poison frogs! targeted clean-label poisoning attacks on neural networks." Advances in Neural Information Processing Systems 31 (2018): 6103-6113.

Poisoning-type adversarial attacks

- Performed by finding data point $\tilde{x}$ that is close to the target instance t in the feature (classification) space and also close to the base instance b in input space

  $$\tilde{x} = \arg\min_{x} \|\phi(x) - \phi(t)\|_2^2 + \beta\|x - b\|_2^2$$

  where β is a scalar constant and $\phi(\cdot)$ is some feature generator

- Note: for a vector $x \in \mathbb{R}^d$, its norm is defined as:

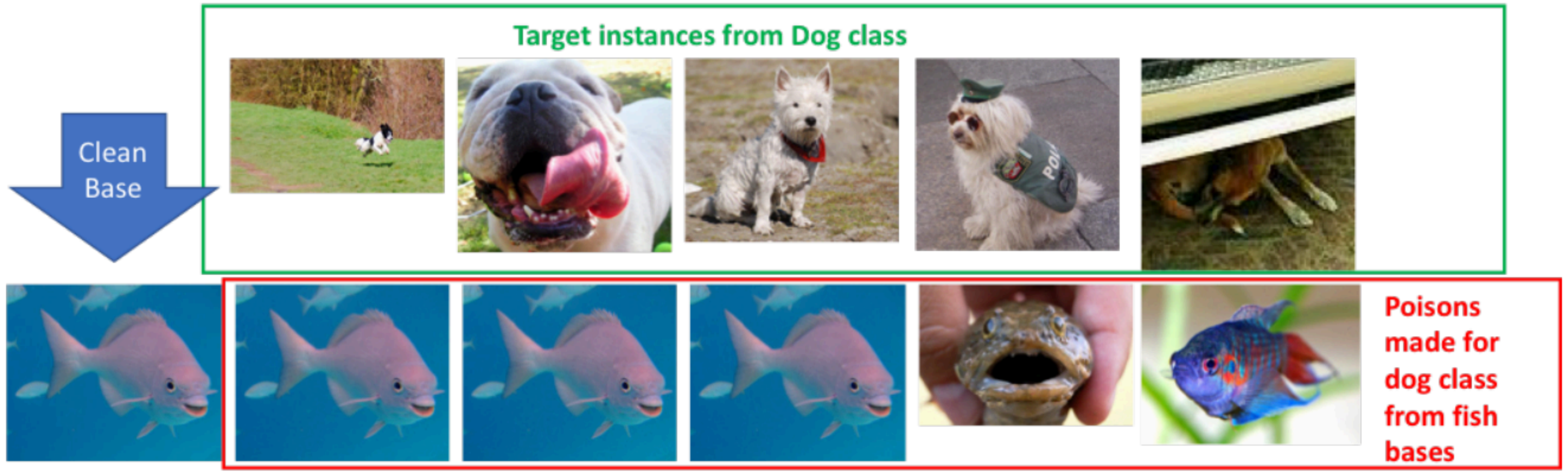
  $\|x\|_p = \sqrt[p]{(x_1)^p + (x_2)^p + \cdots + (x_d)^p}$, for some integer $p \geq 1$, and

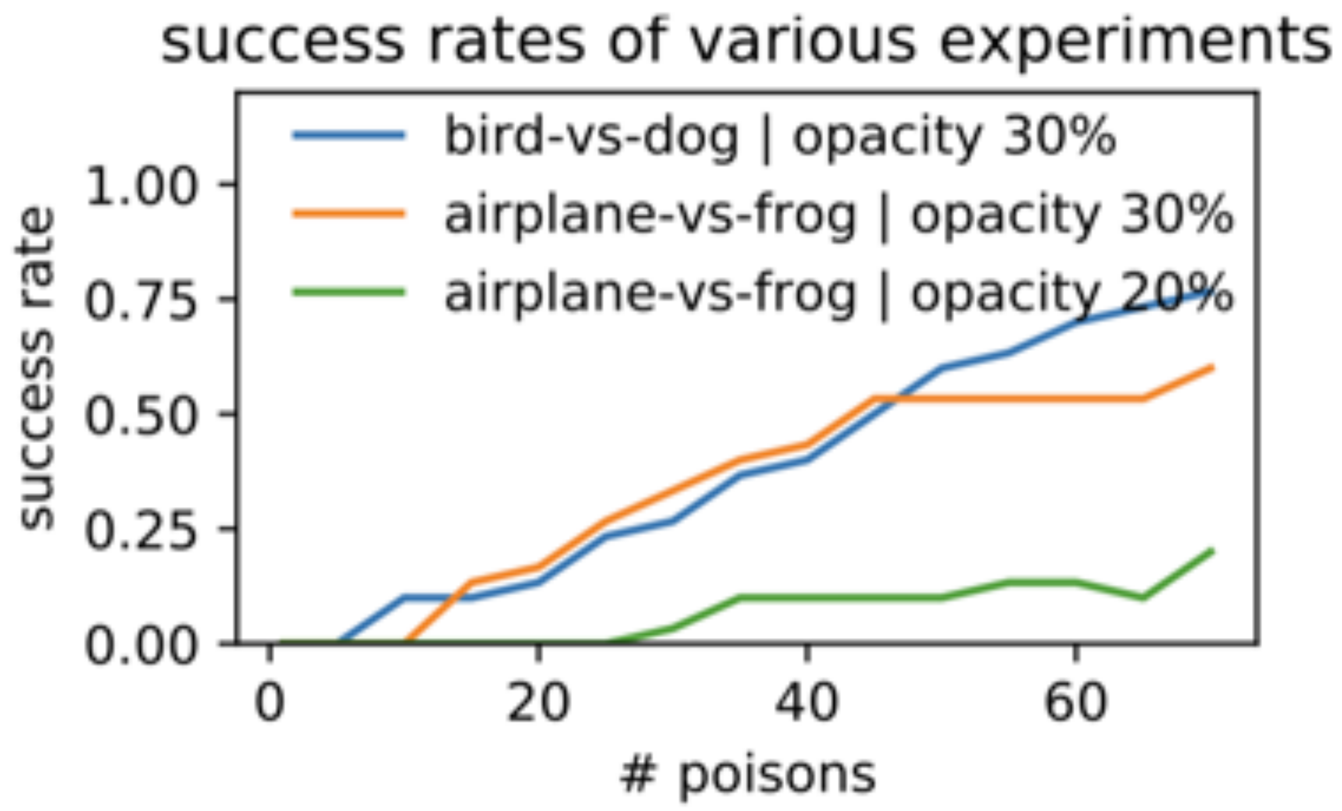
  $\|x\|_\infty = \max\{|x_1|, |x_2|, \cdots, |x_d|\}$

Shafahi, Ali, et al. "Poison frogs! targeted clean-label poisoning attacks on neural networks." Advances in Neural Information Processing Systems 31 (2018): 6103-6113.

Poisoning-type adversarial attacks

- More poisonous samples gives higher success rate





Credits: https://wrhuang.com/project/poison-frog/
Shafahi, Ali, et al. "Poison frogs! targeted clean-label poisoning attacks on neural networks." Advances in Neural Information Processing Systems 31 (2018): 6103-6113.

Evasion-type adversarial attacks

- Performed by adding small perturbation to input to maximize the classifier loss function

$$\delta^* = \arg\max_{\delta \in \Delta} L(f(x + \delta; \theta), y)$$

where

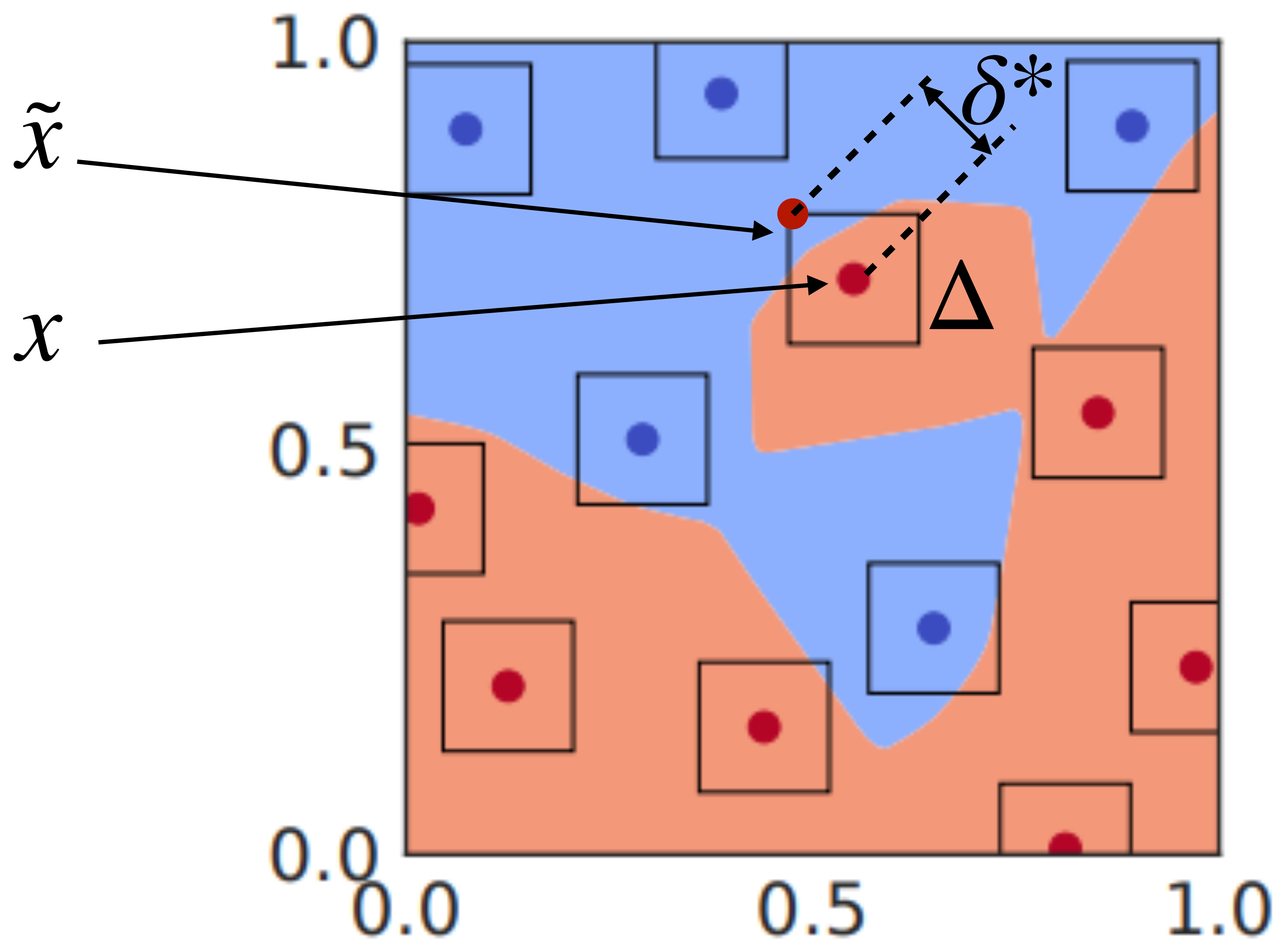
- x is the original sample,

- y is the label,

- $f(\cdot\,;\theta)$ is the targeted classifier

- $\Delta = \{\delta : \|\delta\|_\infty \leq \epsilon\}$ is an ϵ-small ball

- $\tilde{x} = x + \delta^*$ is the adversarial example



Wong, Eric, and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope." International Conference on Machine Learning. PMLR, 2018.

Evasion-type adversarial attacks

- How to solve the maximization problem?
  - For $\tilde{x}$ close to x, linearization gives us

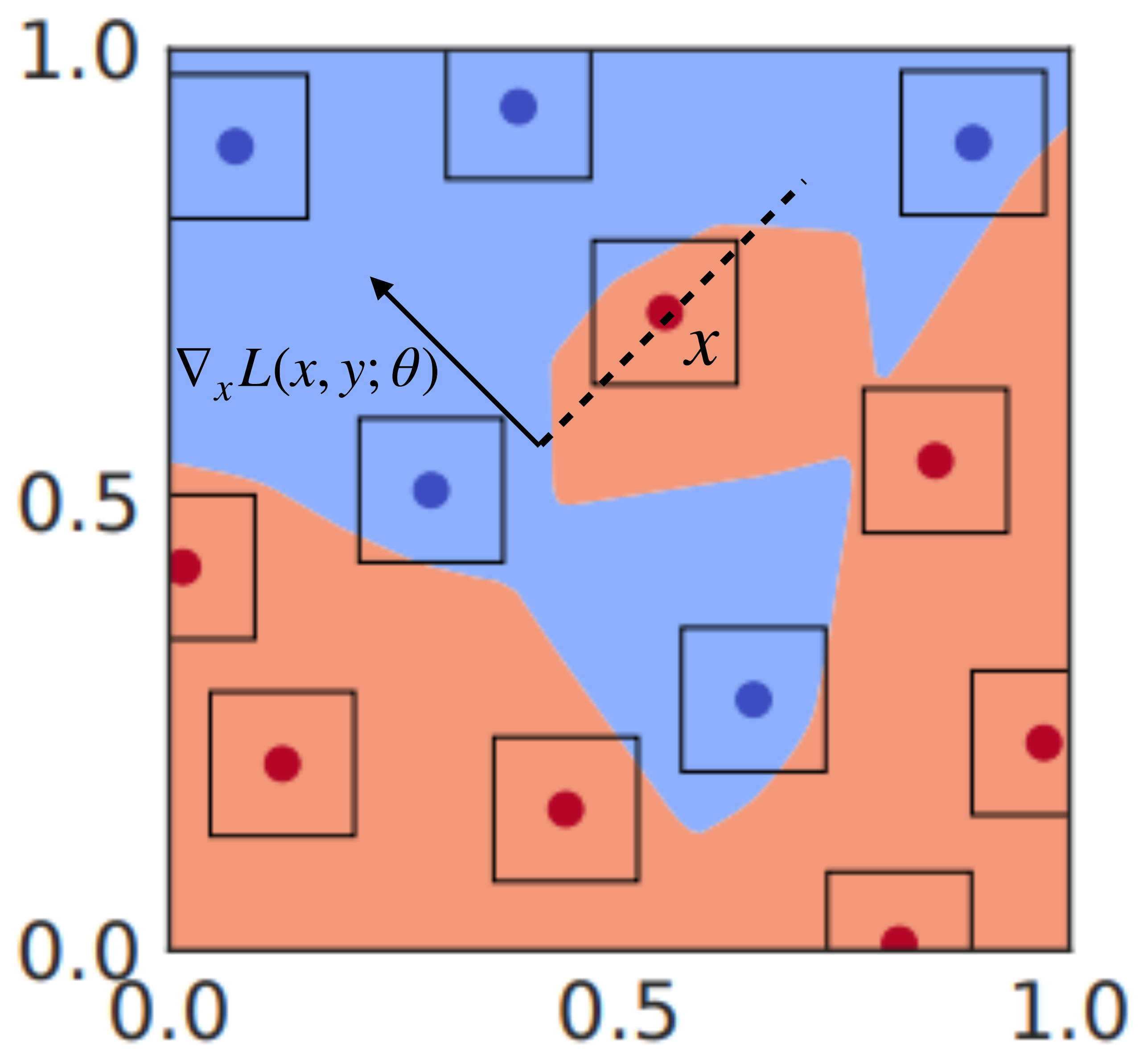
    $$L(\tilde{x}, y; \theta) \approx L(x, y; \theta) + (\tilde{x} - x)^\intercal \nabla_x L(x, y; \theta)$$

  - We rewrite the problem:

    $$\max_{\tilde{x}} L(\tilde{x}, y; \theta) = \max_{\tilde{x}} (\tilde{x} - x)^\intercal \nabla_x L(x, y; \theta)$$

    s.t. $\|\tilde{x} - x\|_\infty \leq \epsilon$

    where $\epsilon > 0$ is some small scalar



Wong, Eric, and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope." International Conference on Machine Learning. PMLR, 2018.

Evasion-type adversarial attacks

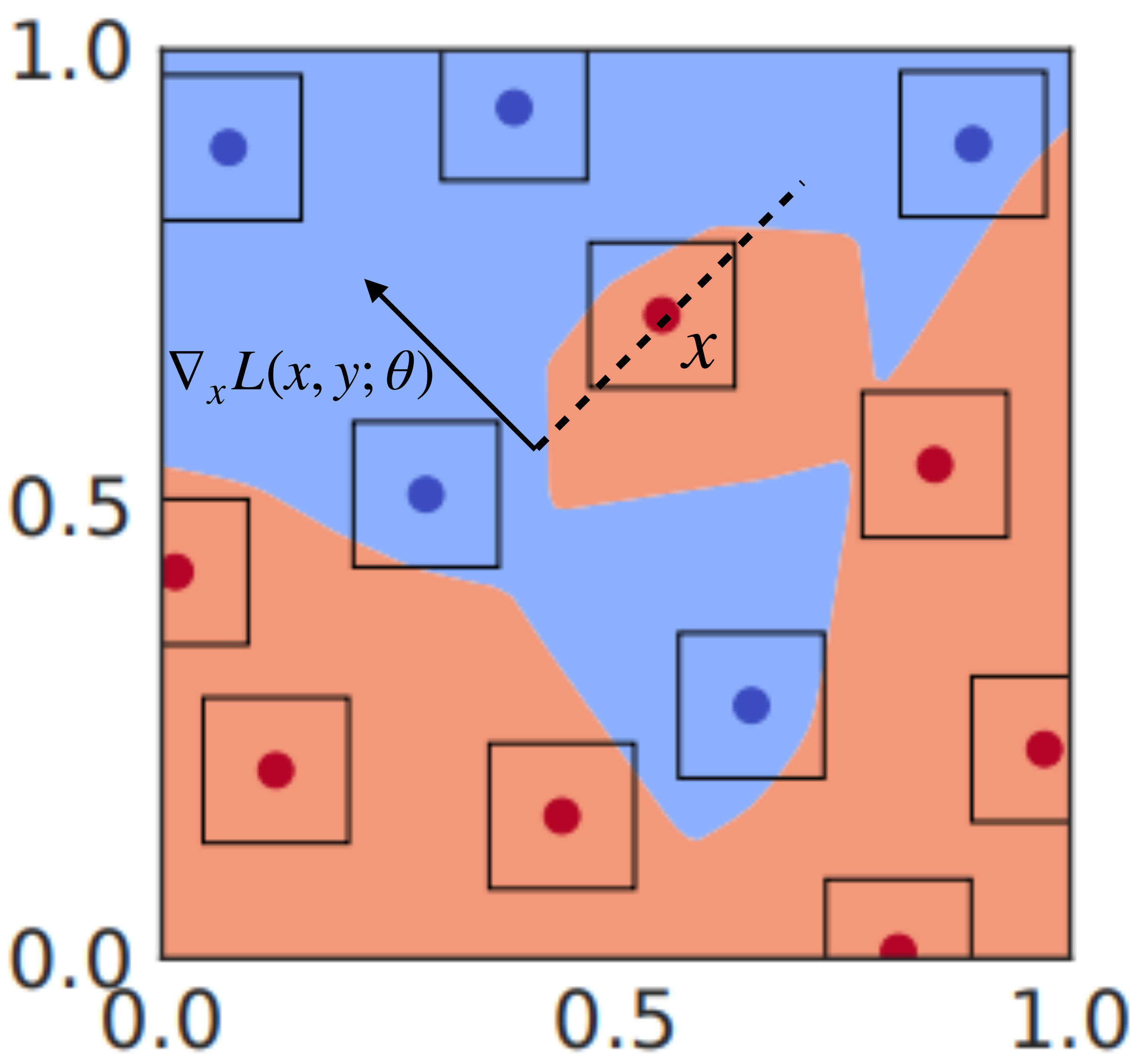
- Our optimization problem:

$$\max_{\tilde{x}} \ (\tilde{x} - x)^\intercal \nabla_x L(x, y; \theta)$$

$$\text{s.t. } \|\tilde{x} - x\|_\infty \leq \epsilon$$

- The solution is $\tilde{x} = x + \epsilon \cdot \text{sign}(\nabla_x L(x, y; \theta))$

$$\text{where sign}(z) = \begin{cases} -1, & z < 0 \\ 0, & z = 0 \\ 1, & z > 1 \end{cases}$$



Wong, Eric, and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope." International Conference on Machine Learning. PMLR, 2018.

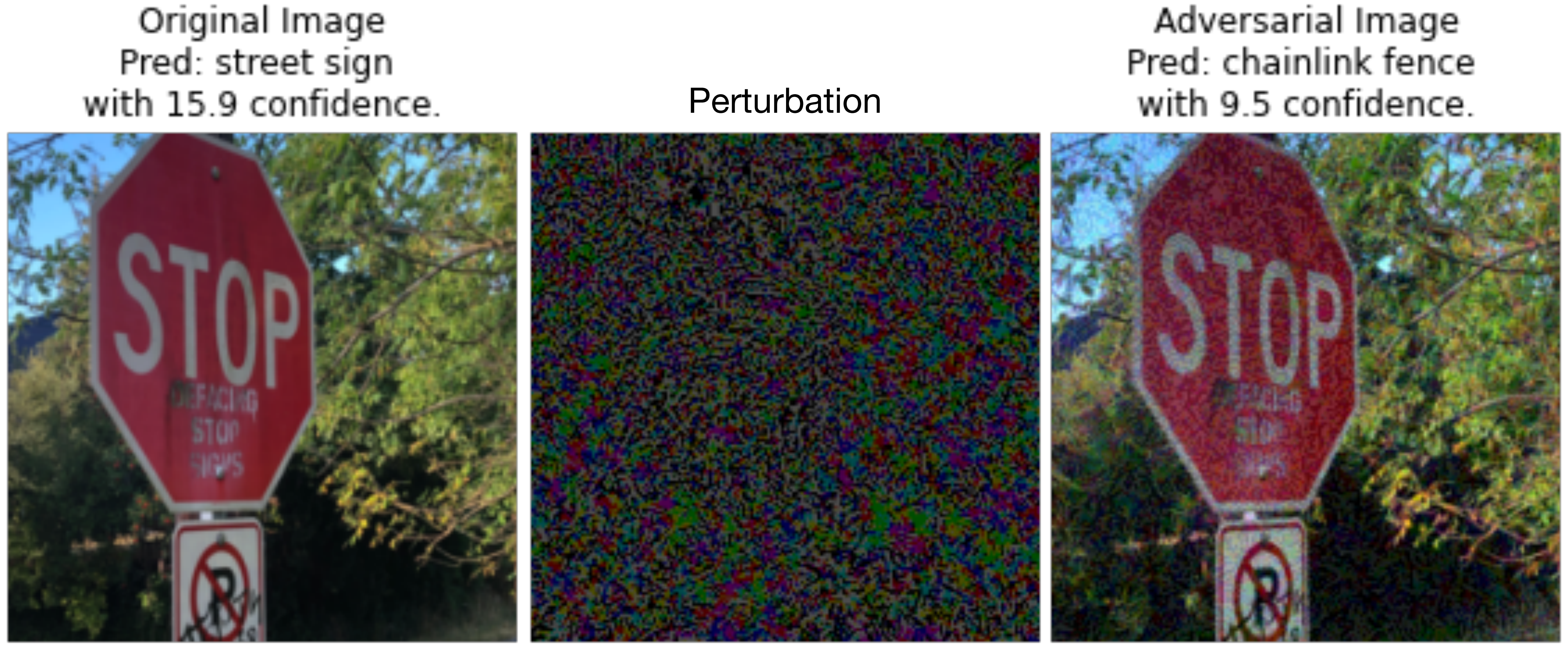
$\text{sign}(\nabla_x L(x, y; \theta))$ has the same dimension of x; it will push $\tilde{x}$ to the "corner"

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

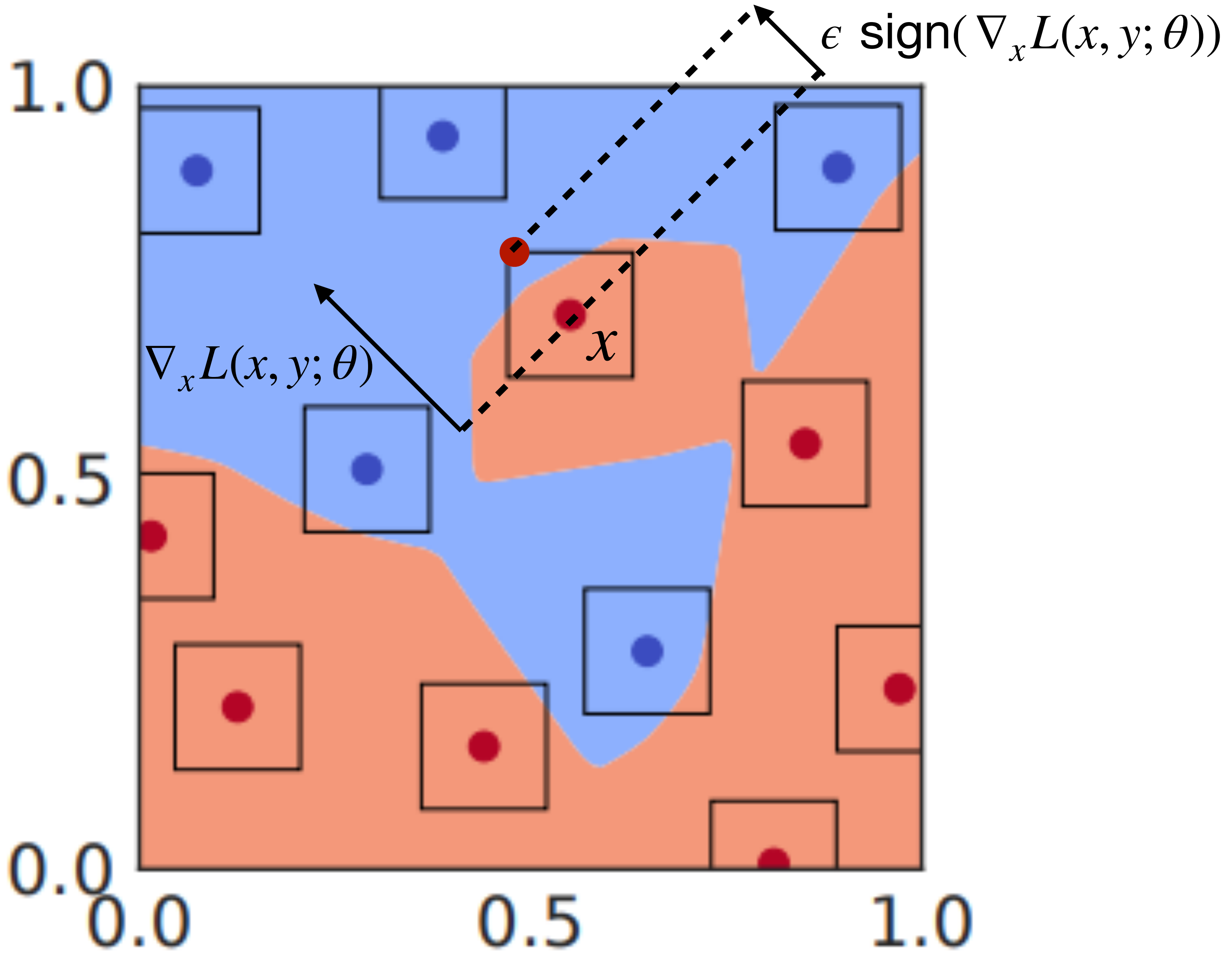
Evasion-type adversarial attacks

- Fast Gradient Sign Method (FGSM)

$$\tilde{x} = x + \epsilon \cdot \text{sign}(\nabla_x L(x, y; \theta))$$



x $\quad$ $\delta^* = \epsilon\ \text{sign}(\nabla_x L(x, y; \theta))$ $\quad$ $\tilde{x} = x + \delta^*$



Wong, Eric, and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope." International Conference on Machine Learning. PMLR, 2018.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

Evasion-type adversarial attacks

- Can be further categorized based on:
  - Information available to the attackers
    - **White-box**: Attackers have full knowledge about the model (architectures, parameters, gradients, etc.) and its output
    - **Black-Box:** Attackers have no information about the model other than its output
  - Loss functions being maximized:
    - **Targeted attack**: Attackers perturb the input image such that the model predicts a specific target class
    - **Untargeted attack:** Attackers perturb the input image such that the model predict any class other than the true class

Evasion-type adversarial attacks

Examples:

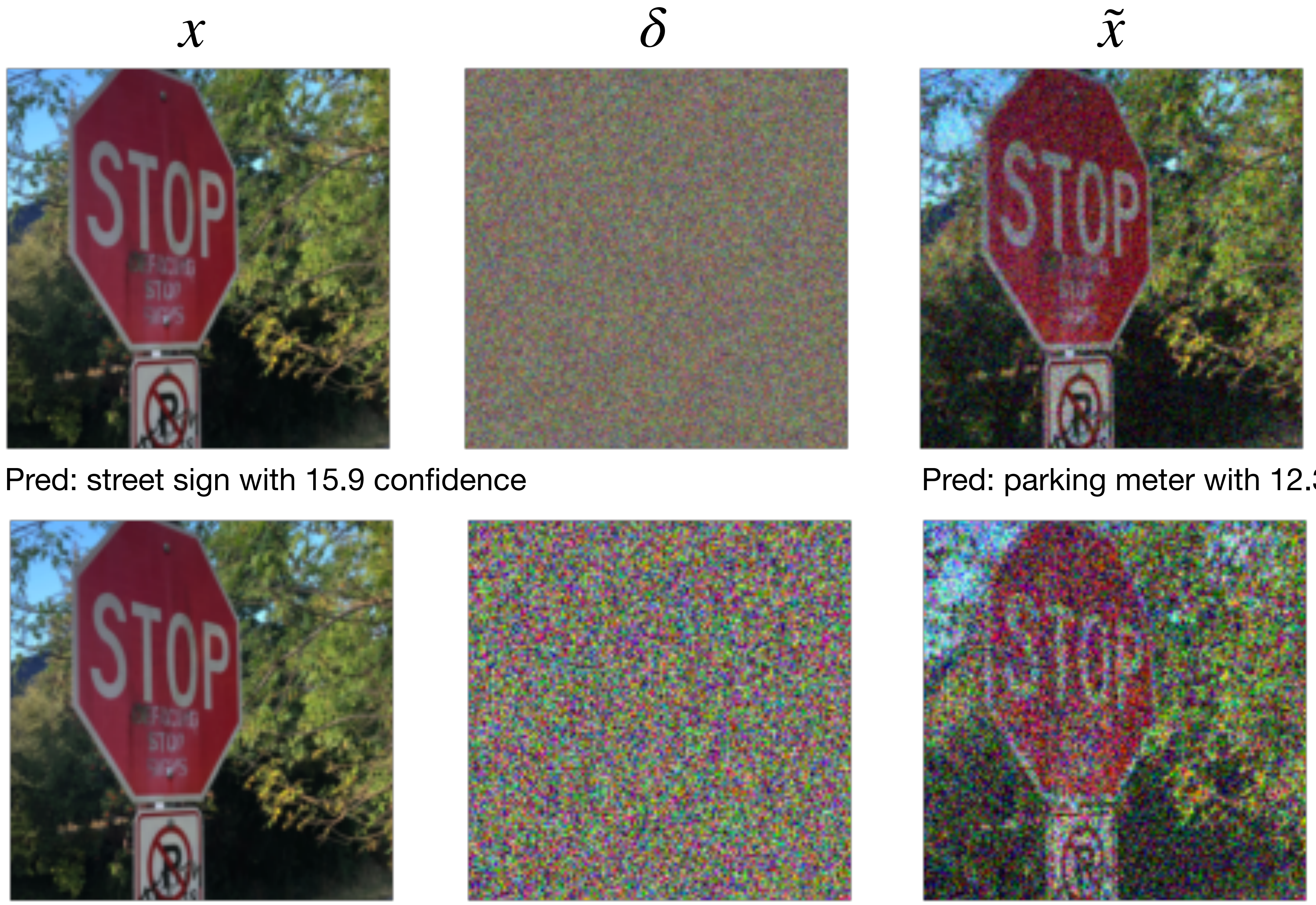
- **Noise Attack** *(black-box, untargeted attack)*
- **Boundary Attack** *(black-box, targeted/untargeted attack)*
- **FGSM** *(white-box, untargeted attack)*
- **Projected Gradient Descent** *(white-box, targeted/untargeted attack)*

Evasion-type adversarial attacks

- **Noise Attack**
  (black-box, untargeted attack):

  Just add random noise to an input image

  $$\tilde{x} = x + \delta, \quad \delta \sim N(0,\sigma^2 I)$$



Pred: street sign with 15.9 confidence

Pred: parking meter with 12.37 confidence

Pred: street sign with 15.9 confidence

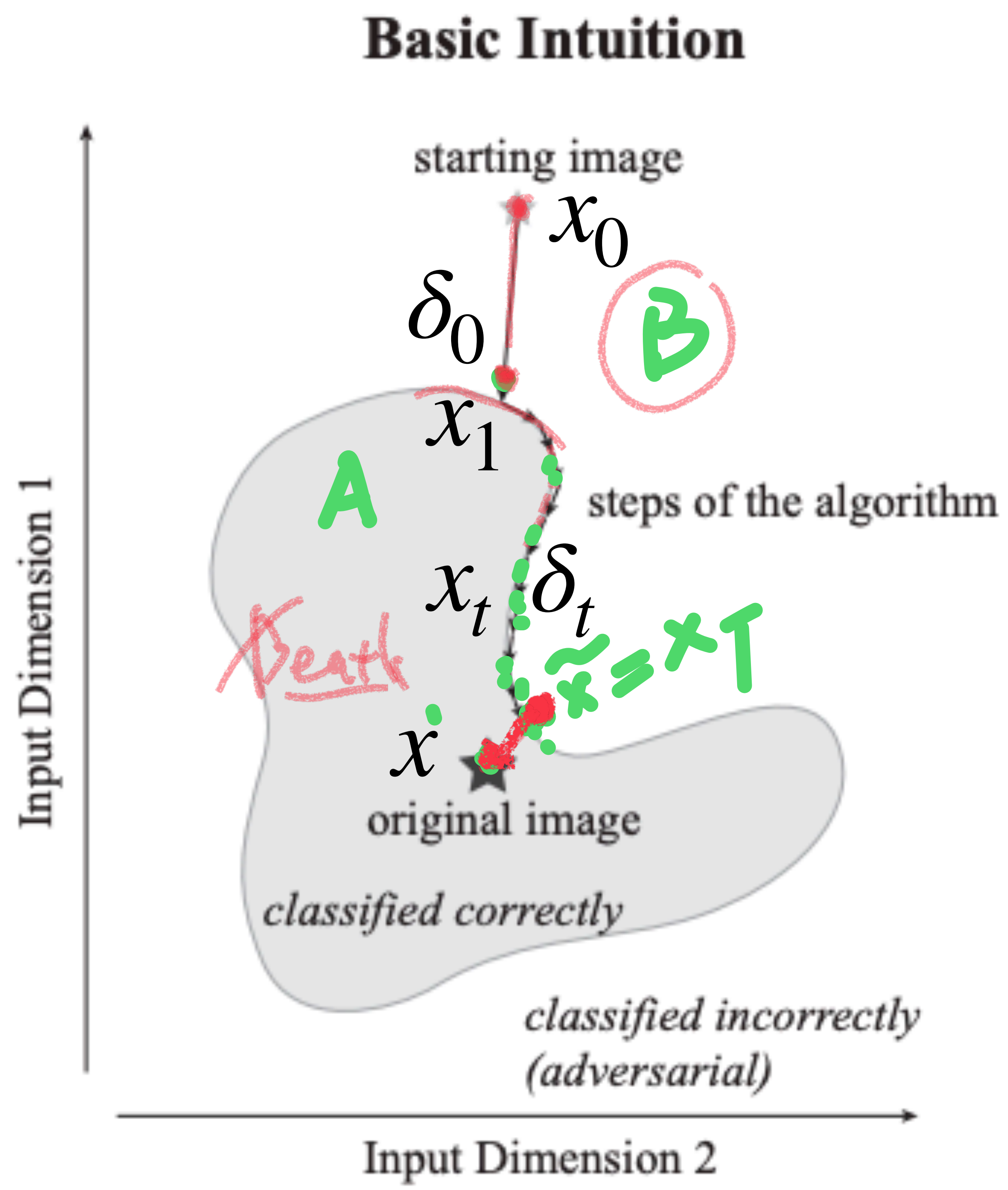
Pred: bubble with 10.96 confidence

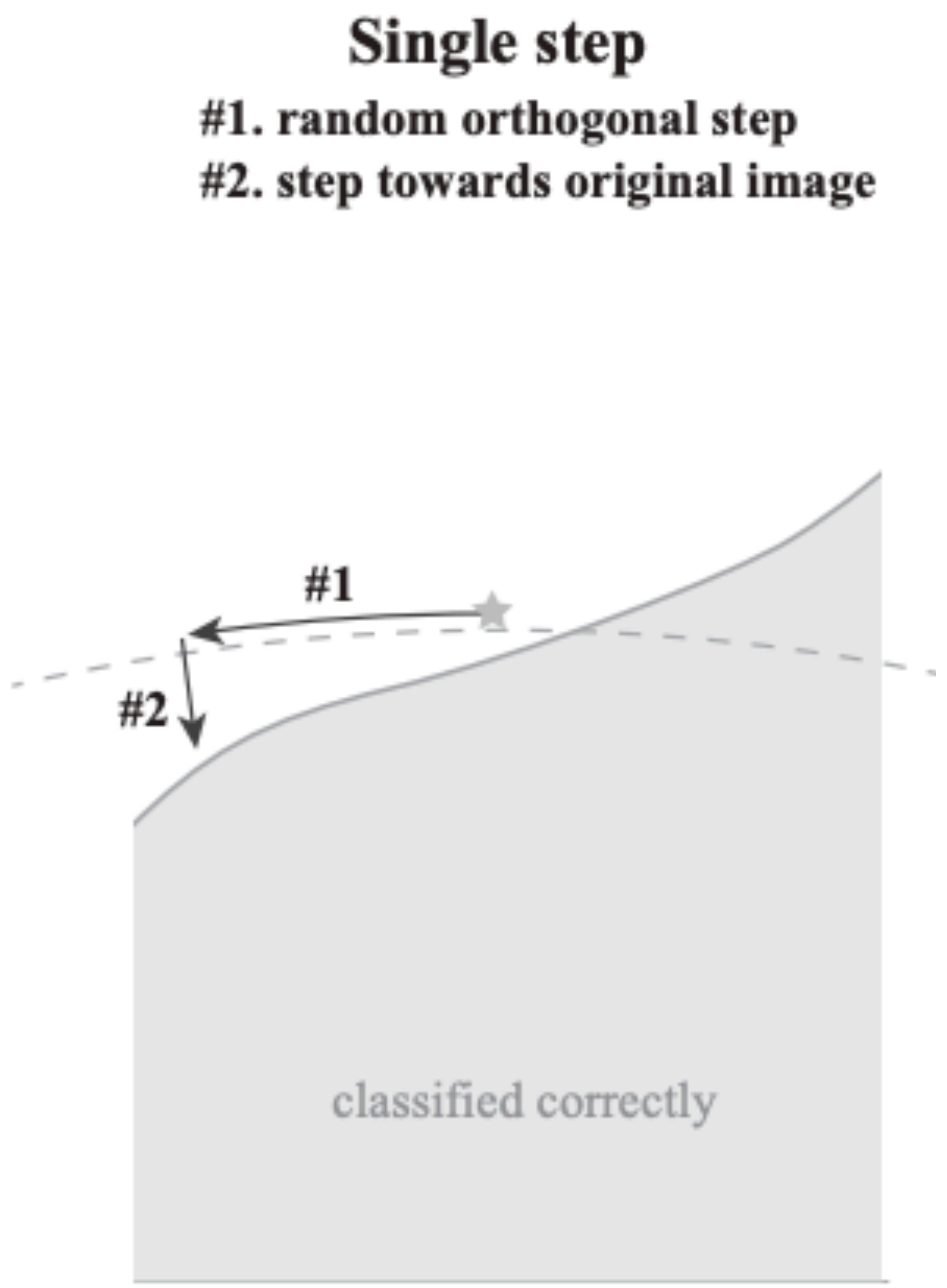
Evasion-type adversarial attacks

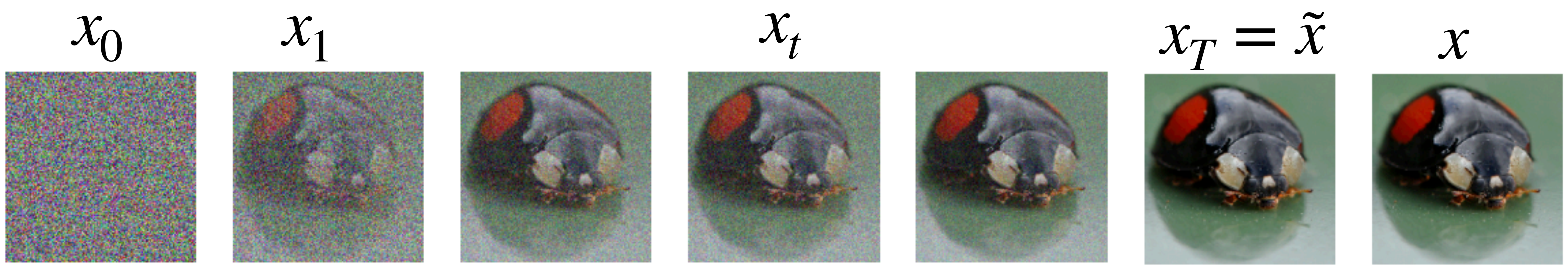
- **Boundary attack**
  (black-box, targeted/untargeted attack):

  rejection sampling along the boundary of adversarial & non-adversarial samples

  $$\tilde{x} = x_{T-1} + \delta_{T-1},\ x_0 \sim \mathcal{N}(0, \sigma^2 I)$$







Original noise

Intermediary image (t = 711)

Adversarial image (T = 200667)

Original image

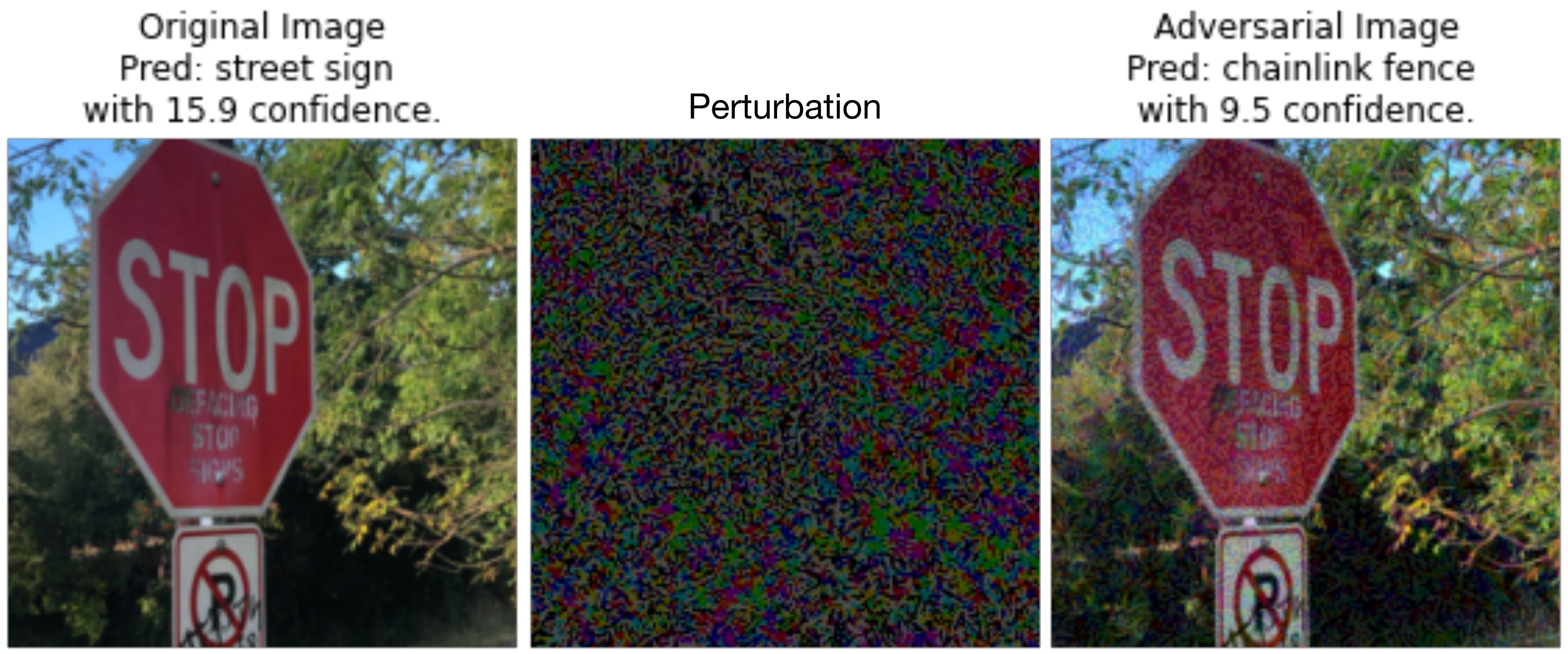
Brendel, Wieland, Jonas Rauber, and Matthias Bethge. "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models." *arXiv preprint arXiv:1712.04248* (2017).

Evasion-type adversarial attacks

- **Fast Gradient Sign Method**
  (white-box, untargeted attack)

  adding adversarial noise in the direction that maximizes the classifier loss

  $$\tilde{x} = x + \epsilon \cdot \text{sign}(\nabla_x L(x, y; \theta))$$



x $\quad$ $\delta* = \epsilon \text{ sign}(\nabla_x L(x, y; \theta))$ $\quad$ $\tilde{x} = x + \delta*$

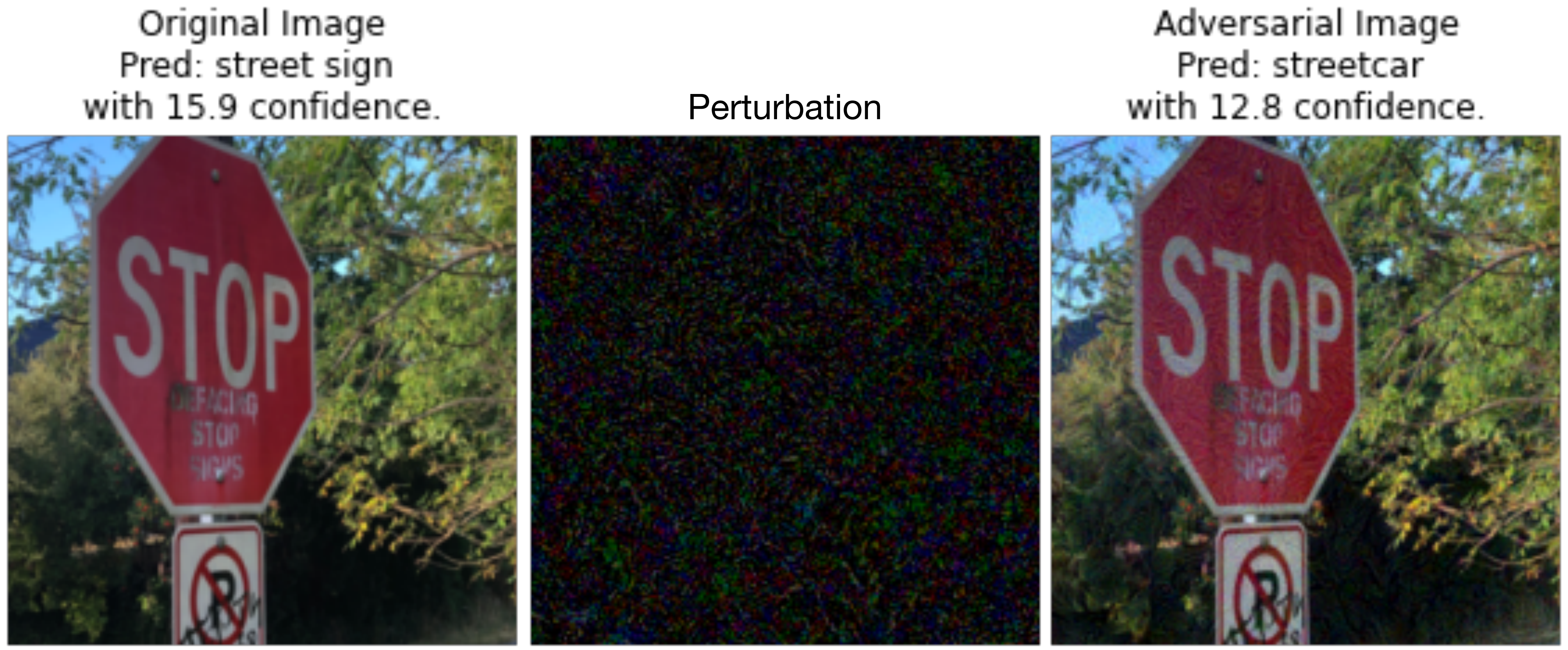
Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).

Evasion-type adversarial attacks

- **Projected Gradient Descent**
  (white-box, targeted/untargeted attack)

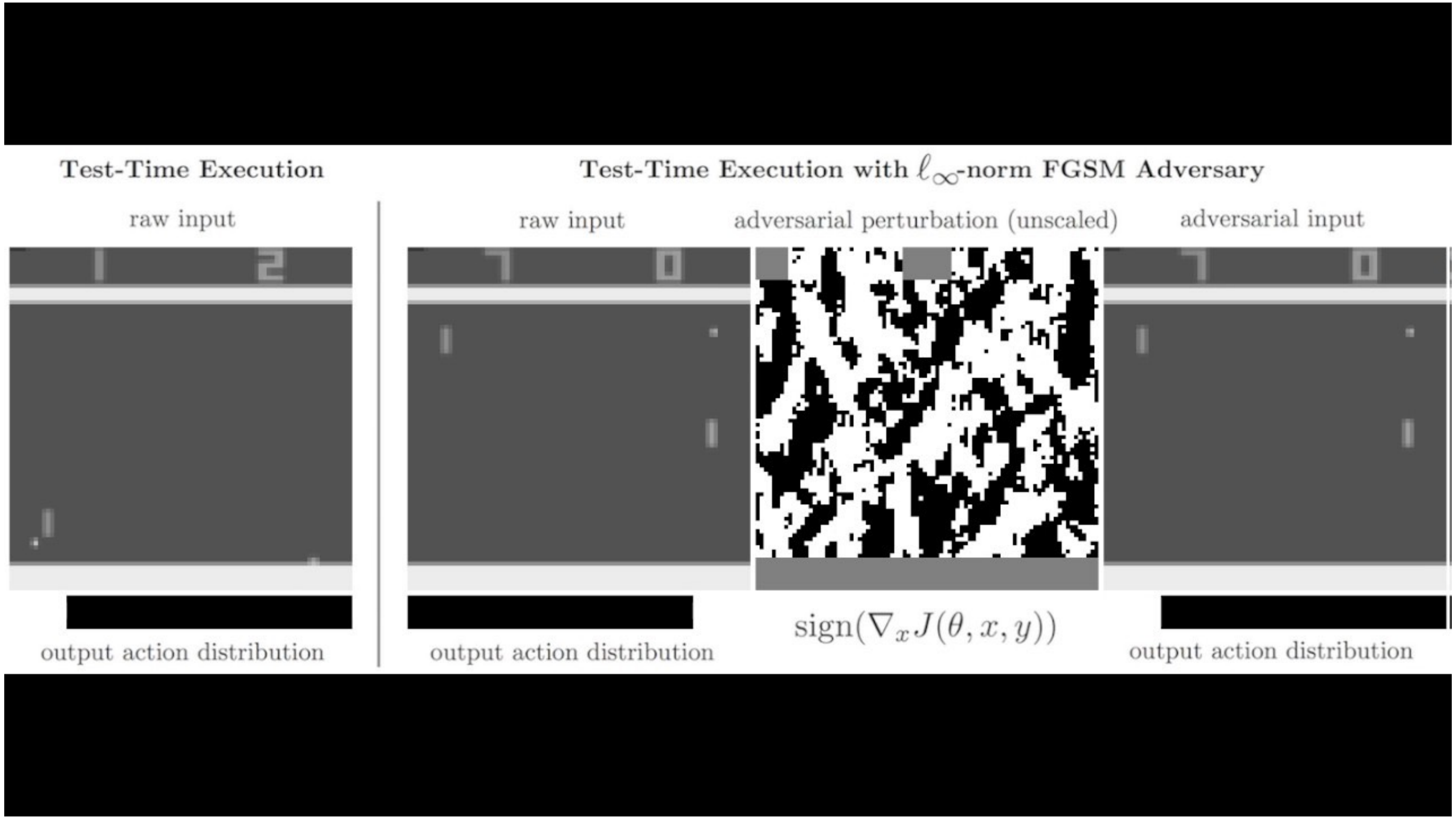
  iteratively performing FGSM to keep increasing the confidence of the target class $y*$

  $$\delta_t = \epsilon \cdot \text{sign}(\nabla_x L(x_t, y*; \theta))$$
  $$x_t = x_{t-1} + \delta_{t-1}$$



$x = x_0$ $\qquad$ $\delta* = \sum_t \delta_t$ $\qquad$ $\tilde{x} = x_0 + \delta*$

Adversarial on reinforcement learing



Adversarial examples can attack decision making algorithms (e.g. RL agent)

Source: https://www.youtube.com/watch?v=4r_KCjKHV_M

Huang, Sandy, et al. "Adversarial attacks on neural network policies." *arXiv preprint arXiv:1702.02284* (2017).

Other adversarial attacks

- Attacking methods are (still) actively developed

`L2ContrastReductionAttack`	Reduces the contrast of the input usin
`VirtualAdversarialAttack`	Second-order gradient-based attack or
`DDNAttack`	The Decoupled Direction and Norm L2
`L2ProjectedGradientDescentAttack`	L2 Projected Gradient Descent
`LinfProjectedGradientDescentAttack`	Linf Projected Gradient Descent
`L2BasicIterativeAttack`	L2 Basic Iterative Method
`LinfBasicIterativeAttack`	L-infinity Basic Iterative Method
`L2FastGradientAttack`	Fast Gradient Method (FGM)
`LinfFastGradientAttack`	Fast Gradient Sign Method (FGSM)
`L2AdditiveGaussianNoiseAttack`	Samples Gaussian noise with a fixed L2
`L2AdditiveUniformNoiseAttack`	Samples uniform noise with a fixed L2
`L2ClippingAwareAdditiveGaussianNoiseAttack`	Samples Gaussian noise with a fixed L2
`L2ClippingAwareAdditiveUniformNoiseAttack`	Samples uniform noise with a fixed L2
`LinfAdditiveUniformNoiseAttack`	Samples uniform noise with a fixed L-ir
`L2RepeatedAdditiveGaussianNoiseAttack`	Repeatedly samples Gaussian noise wi
`L2RepeatedAdditiveUniformNoiseAttack`	Repeatedly samples uniform noise with
`L2ClippingAwareRepeatedAdditiveGaussianNoiseAttack`	Repeatedly samples Gaussian noise wi
`L2ClippingAwareRepeatedAdditiveUniformNoiseAttack`	Repeatedly samples uniform noise with
`LinfRepeatedAdditiveUniformNoiseAttack`	Repeatedly samples uniform noise with
`InversionAttack`	Creates “negative images” by inverting

Source: https://foolbox.readthedocs.io/en/stable/modules/attacks.html

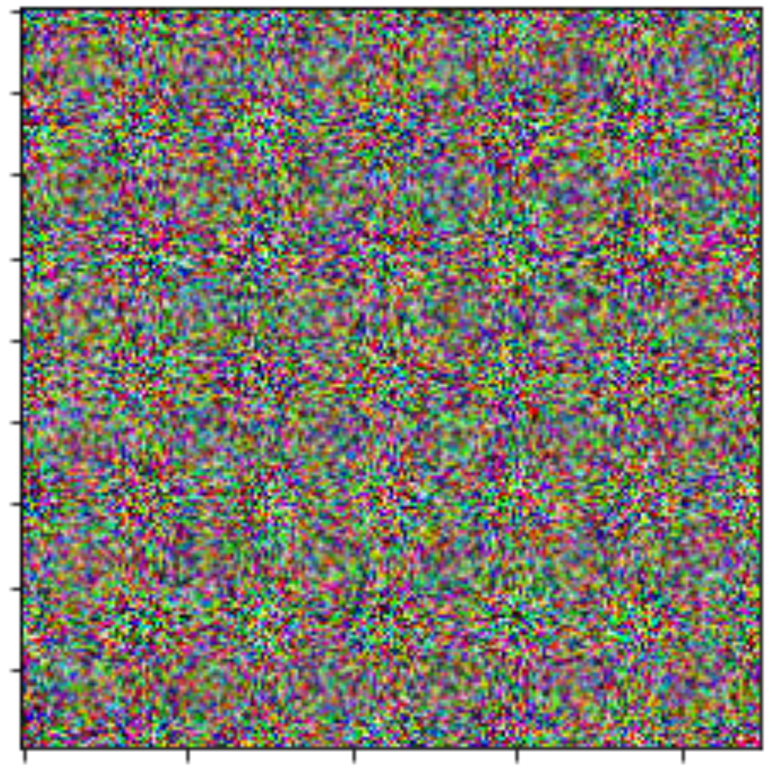
`BinarySearchContrastReductionAttack`	Reduces the contrast of the input usin
`LinearSearchContrastReductionAttack`	Reduces the contrast of the input usin
`L2CarliniWagnerAttack`	Implementation of the Carlini & Wagn
`NewtonFoolAttack`	Implementation of the NewtonFool At
`EADAttack`	Implementation of the EAD Attack wit
`GaussianBlurAttack`	Blurs the inputs using a Gaussian filter
`L2DeepFoolAttack`	A simple and fast gradient-based adve
`LinfDeepFoolAttack`	A simple and fast gradient-based adve
`SaltAndPepperNoiseAttack`	Increases the amount of salt and pepp
`LinearSearchBlendedUniformNoiseAttack`	Blends the input with a uniform noise i
`BinarizationRefinementAttack`	For models that preprocess their input
`DatasetAttack`	Draws randomly from the given datase
`BoundaryAttack`	A powerful adversarial attack that requ
`L0BrendelBethgeAttack`	L0 variant of the Brendel & Bethge adv
`L1BrendelBethgeAttack`	L1 variant of the Brendel & Bethge adv
`L2BrendelBethgeAttack`	L2 variant of the Brendel & Bethge adv
`LinfinityBrendelBethgeAttack`	L-infinity variant of the Brendel & Beth
`FGM`	alias of `foolbox.attacks.fast_gradient`
`FGSM`	alias of `foolbox.attacks.fast_gradient`
`L2PGD`	alias of `foolbox.attacks.projected_gra`
`LinfPGD`	alias of `foolbox.attacks.projected_gra`
`PGD`	alias of `foolbox.attacks.projected_gra`

Physical adversarial attacks

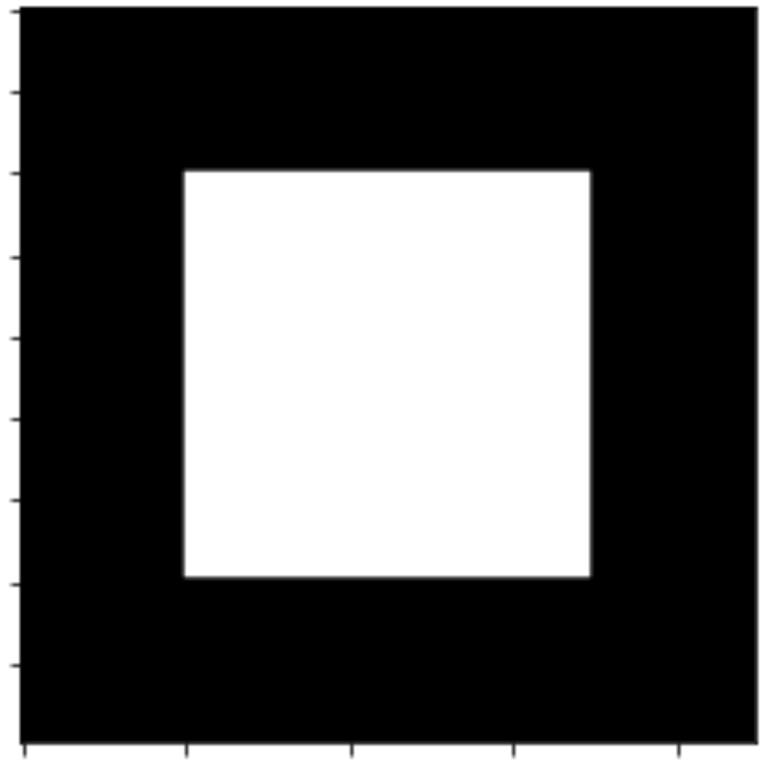
- Goal: given an image x, we want to flip the classifier prediction from y to another class $y*$.
  - Perturbing only part of the image: use filter M_x as a mask/filter

    $$\min_{\delta} L(x + M_x \odot \delta, y, y*; \theta) + \lambda ||M_x \odot \delta||_p$$

    where $\odot$ denotes element-wise multiplication



δ

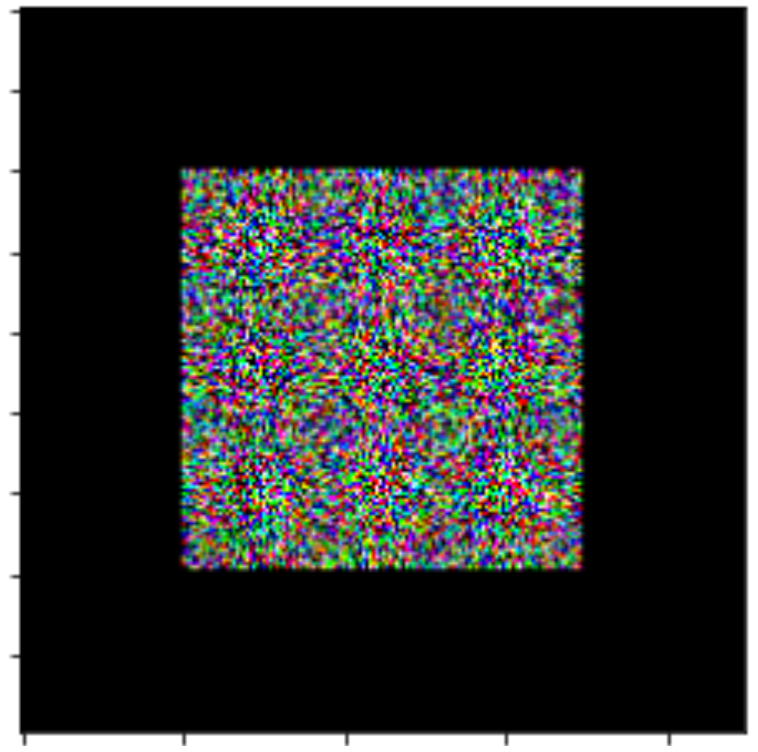


M_x

Original image

x

Masked perturbation



$M_x \odot \delta$

Perturbed image

$x + M_x \odot \delta$

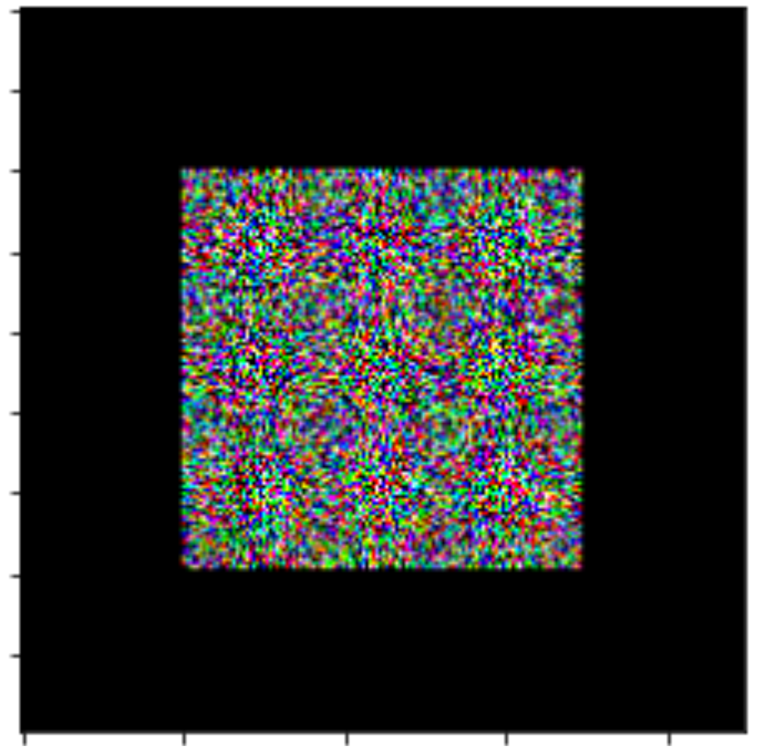
Physical adversarial attacks

- Add physical constraints:

    - Transformation: apply $T(\cdot)$ to the masked perturbation

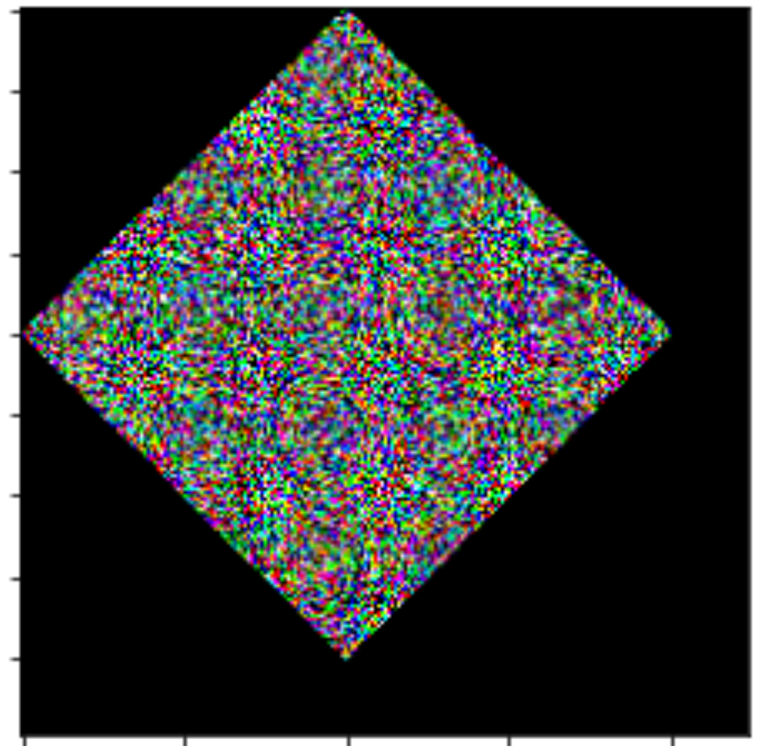
$$\min_{\delta} L(x + T(M_x \odot \delta), y^*; \theta) + \lambda \|M_x \odot \delta\|_p$$

Masked perturbation



$M_x \odot \delta$

Transformed masked perturbation

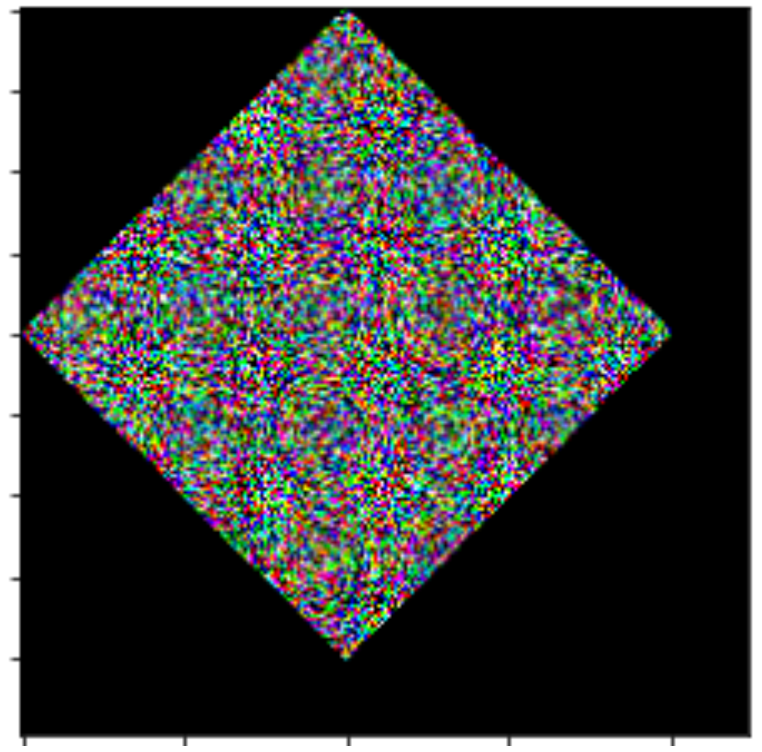


$T(M_x \odot \delta)$

Original image

x

Transformed masked perturbation



$T(M_x \odot \delta)$

Perturbed image

$x + T(M_x \odot \delta)$

Physical adversarial attacks

- Add physical constraints:

  - Penalizing for non-printable perturbation (NPS penalty):

$$\min_{\delta} \lambda ||M_x \cdot \delta||_p + L(x + T(M_x \cdot \delta), y^*; \theta) + NPS$$

x high NPS low NPS

Given a set of printable colors (RGB triples) P and a set $R(\delta)$ of (unique) RGB triples used in the perturbation that need to be printed out in physical world, the non-printability score is given by:

$$NPS(\delta) = \sum_{\hat{p} \in R(\delta)} \prod_{p' \in P} |\hat{p} - p'| \tag{1}$$

Eykholt, Kevin, et al. "Robust physical-world attacks on deep learning visual classification." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

Physical adversarial attacks

- Examples:

Distance/Angle	Subtle Poster	Subtle Poster Right Turn	Camouflage Graffiti
5′ 0°			
5′ 15°			
10′ 0°			
10′ 30°			
40′ 0°			
Targeted-Attack Success	100%	73.33%	66.67%

Eykholt, Kevin, et al. "Robust physical-world attacks on deep learning visual classification." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

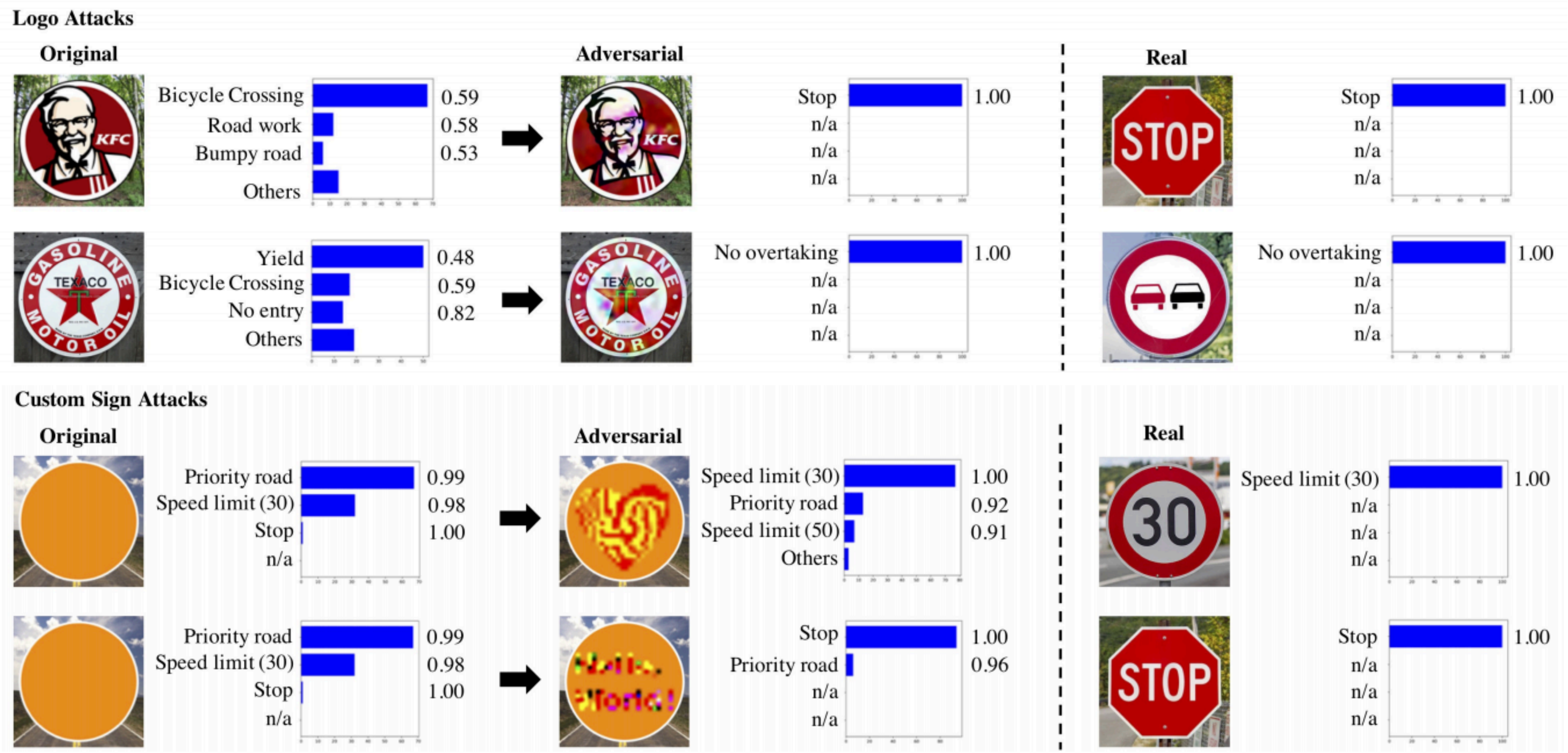
Adversarial examples in autonomous driving

- Adversarial examples can fool autonomous vehicles perception systems

Morgulis, Nir, et al. "Fooling a real car with adversarial traffic signs." arXiv preprint arXiv:1907.00374 (2019).

Adversarial examples in autonomous driving

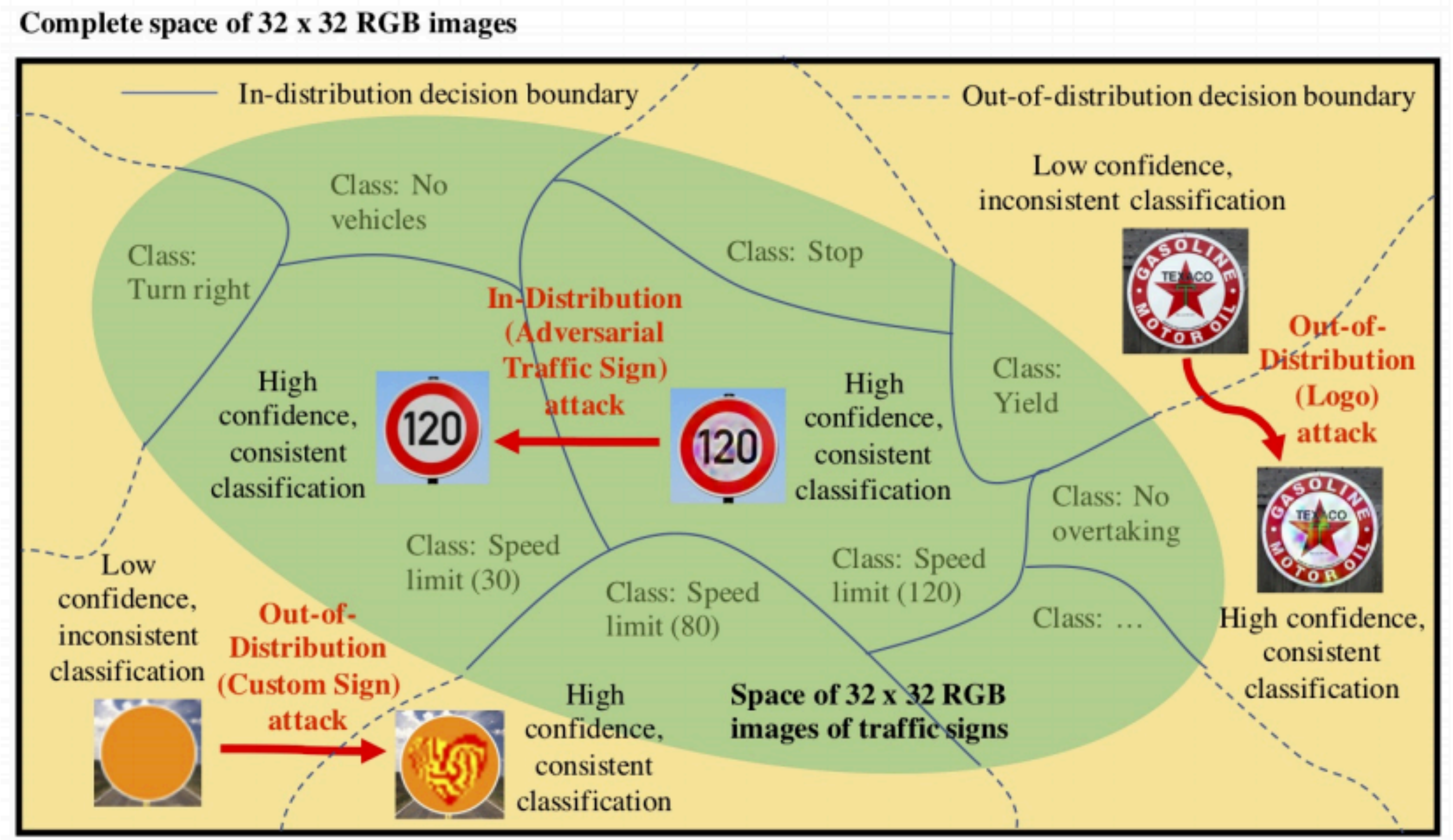
- Success rate of logo-based attacks



Sitawarin, Chawin, et al. "Darts: Deceiving autonomous cars with toxic signs." arXiv preprint arXiv:1802.06430 (2018).

Adversarial examples in autonomous driving

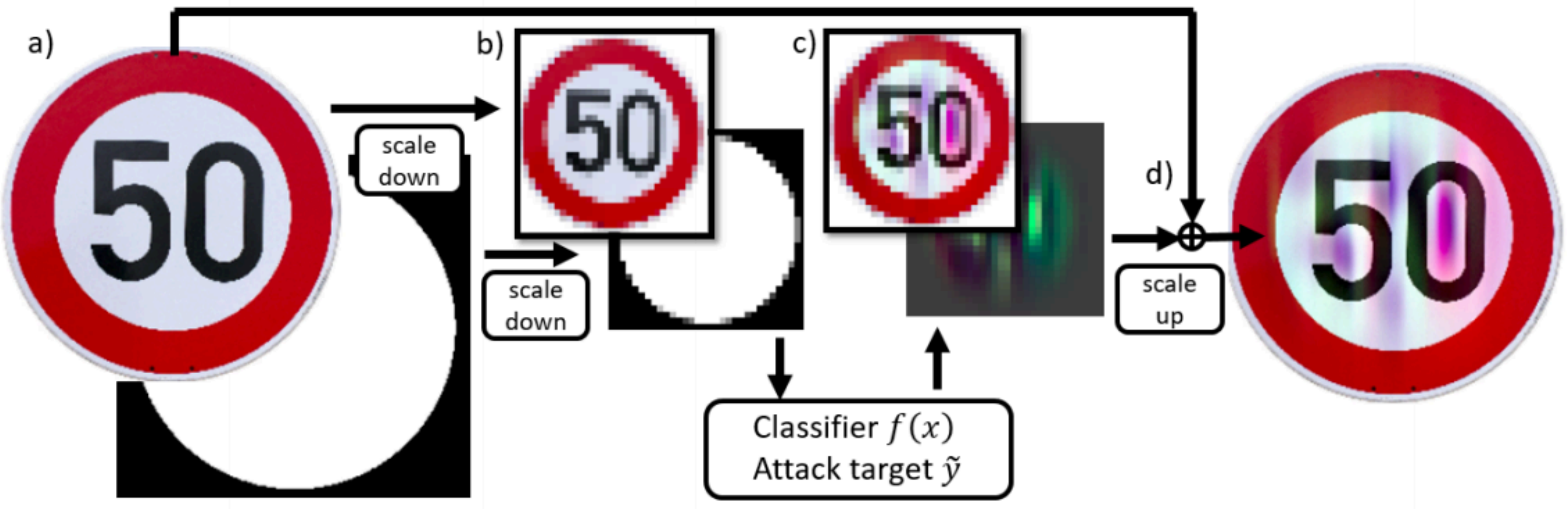
- Illustration of attacks in image spaces



Sitawarin, Chawin, et al. "Darts: Deceiving autonomous cars with toxic signs." arXiv preprint arXiv:1802.06430 (2018).

Adversarial examples in autonomous driving

- **Pipeline**:
  a) Take a high-resolution image
  b) Scale down for easier attack
  c) Find adversarial perturbation within L_p neighborhood
  d) Scale up the perturbation then add to original image



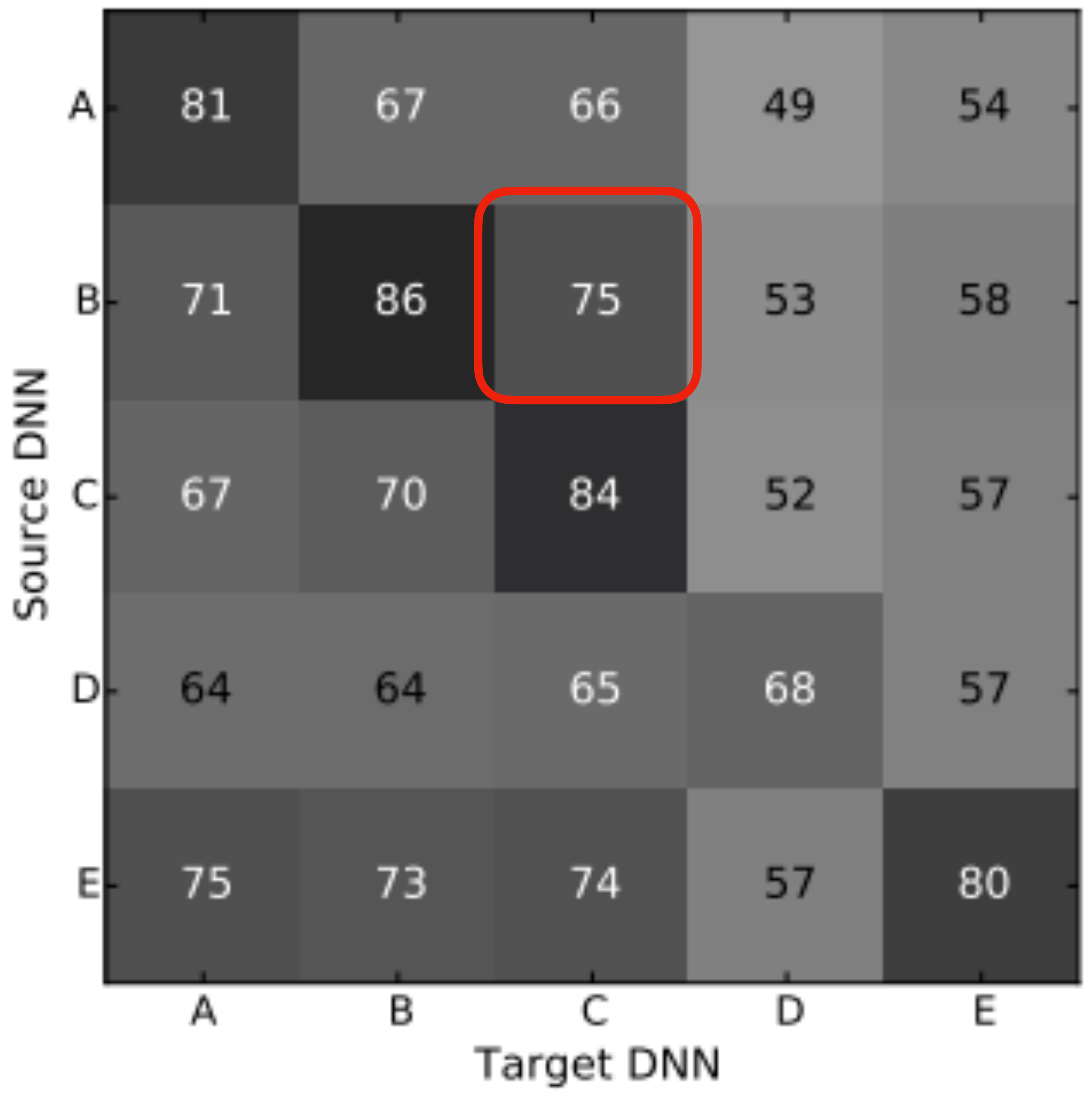
Morgulis, Nir, et al. "Fooling a real car with adversarial traffic signs." arXiv preprint arXiv:1907.00374 (2019).

The generality of adversaries

- Adversarial examples are generalizable across models and architectures

ACROSS MODELS (SAME ARCHITECTURE)

Success percentage

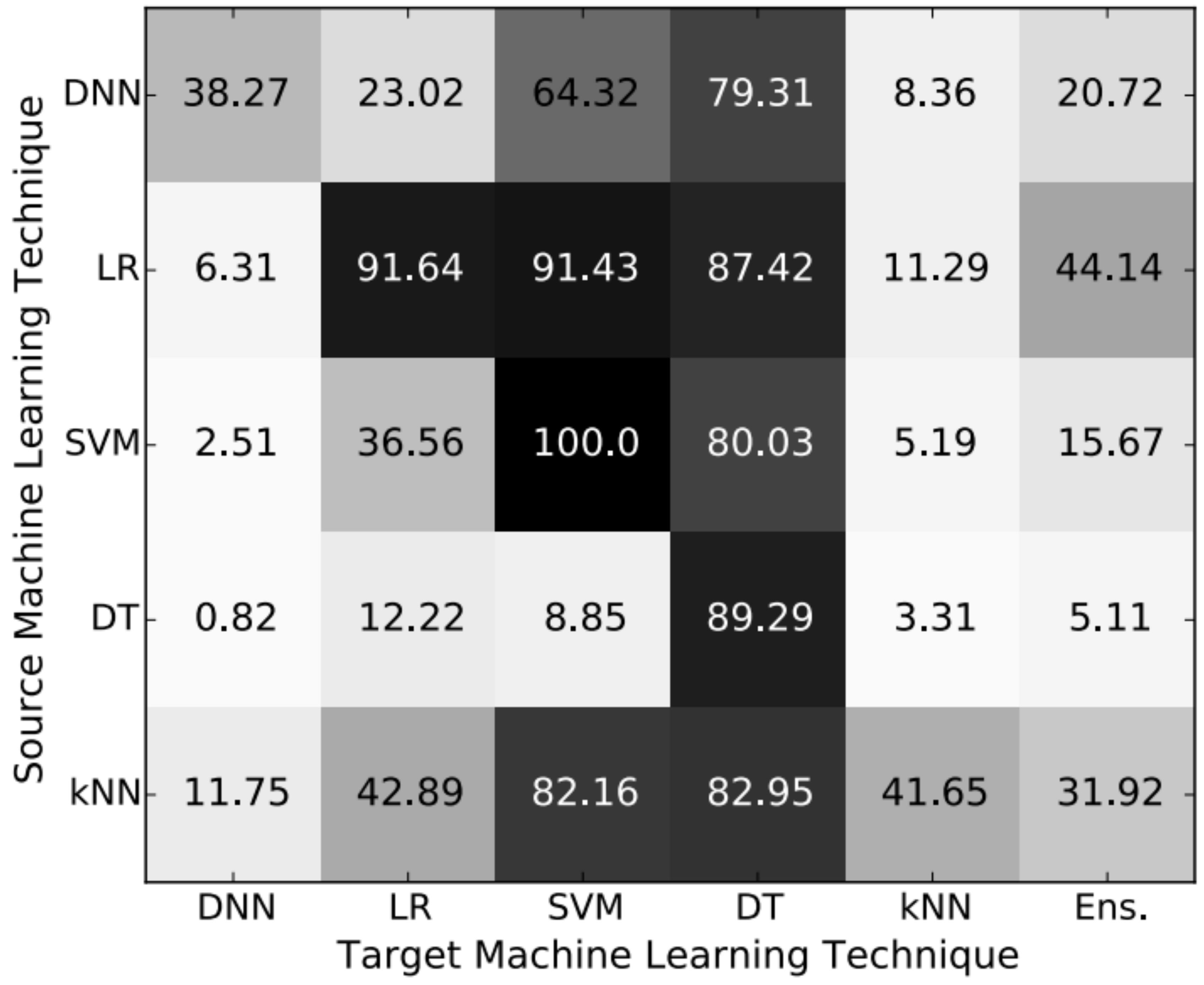


E.g. Adversarial samples from DNN B has 75% success rate attacking DNN C

A, B, C, D, E are models trained with the same method with different datasets

ACROSS ARCHITECTURES

Success percentage



DNN: deep neural networks
LR: logistic regression
SVM: support vector machines
DT: decision trees
kNN: k-nearest neighbors
Ens: ensemble models

Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." arXiv preprint arXiv:1605.07277 (2016).

Adversarial examples are not bugs, they are features

- World 1: **Adversarial examples exploit directions irrelevant for classification ("bugs")**. In this world, adversarial examples occur because classifiers behave poorly off-distribution, when they are evaluated on inputs that are not natural images. Here, adversarial examples would occur in arbitrary directions, having nothing to do with the true data distribution.

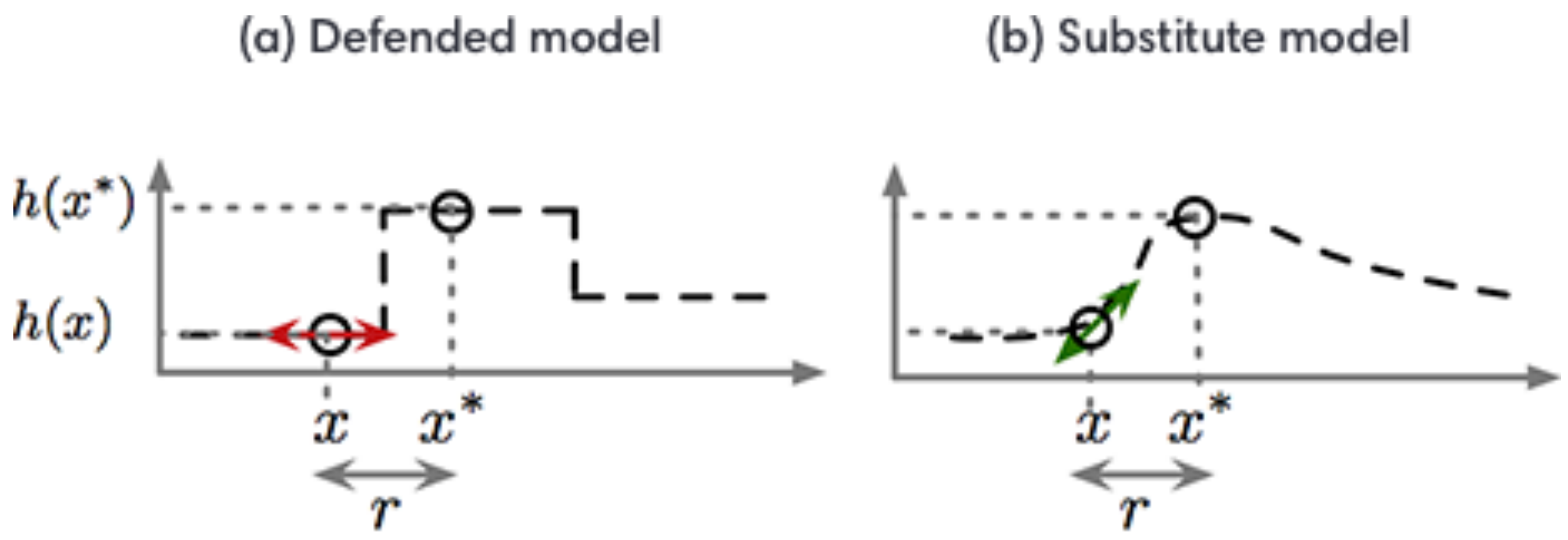
- World 2: **Adversarial examples exploit useful directions for classification ("features")**. In this world, adversarial examples occur in directions that are still "on-distribution", and which contain features of the target class. For example, consider the perturbation that makes an image of a dog to be classified as a cat. In World 2, this perturbation is not purely random, but has something to do with cats. Moreover, we expect that this perturbation transfers to other classifiers trained to distinguish cats vs. dogs.

There are still argument about this ..

https://distill.pub/2019/advex-bugs-discussion/

Defending against adversaries

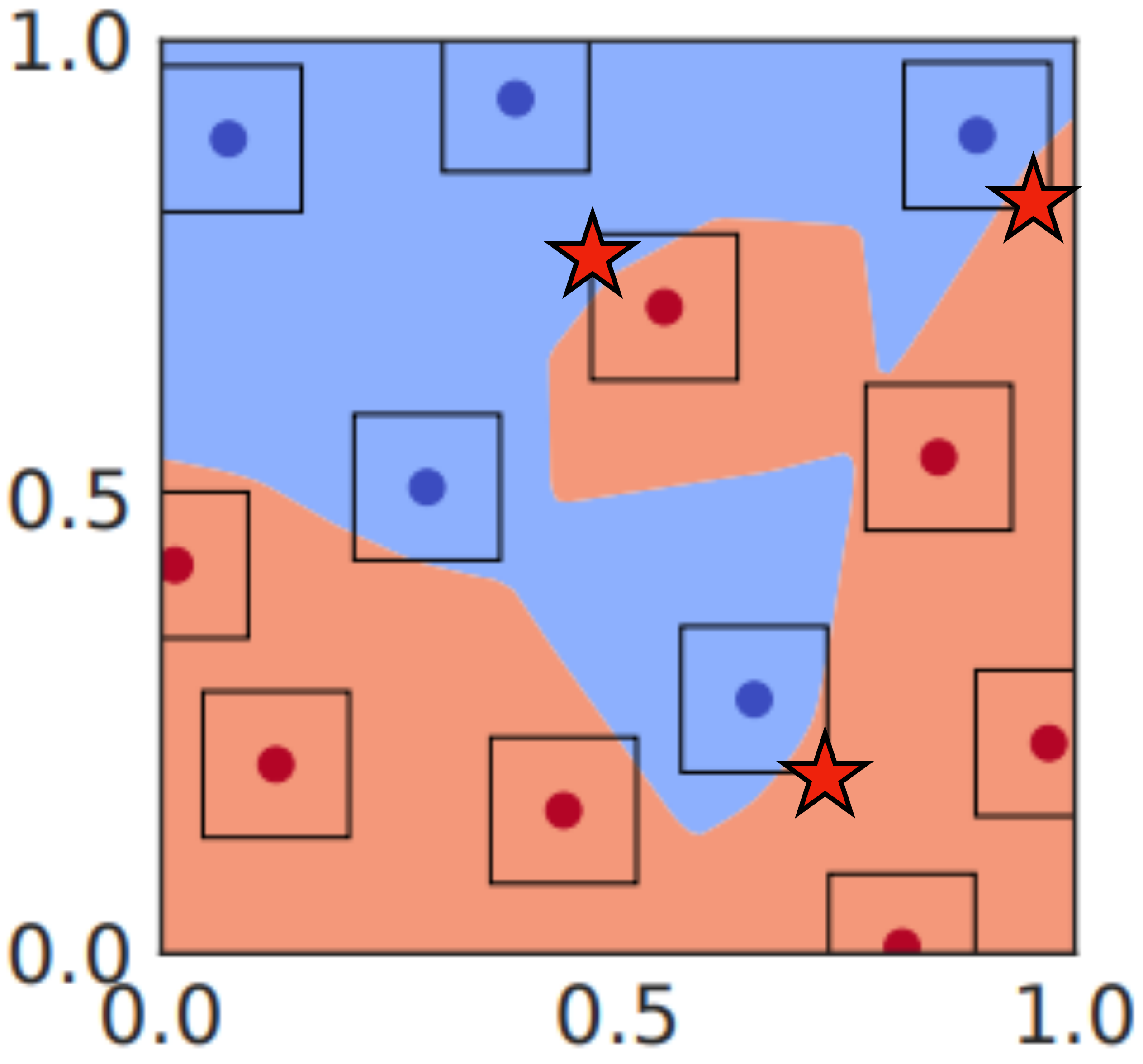
- The adversary generality means we cannot simply hide the model
  - attackers may use a substitute model and still have with a high success rate
  - this is what makes adversarial defense challenging



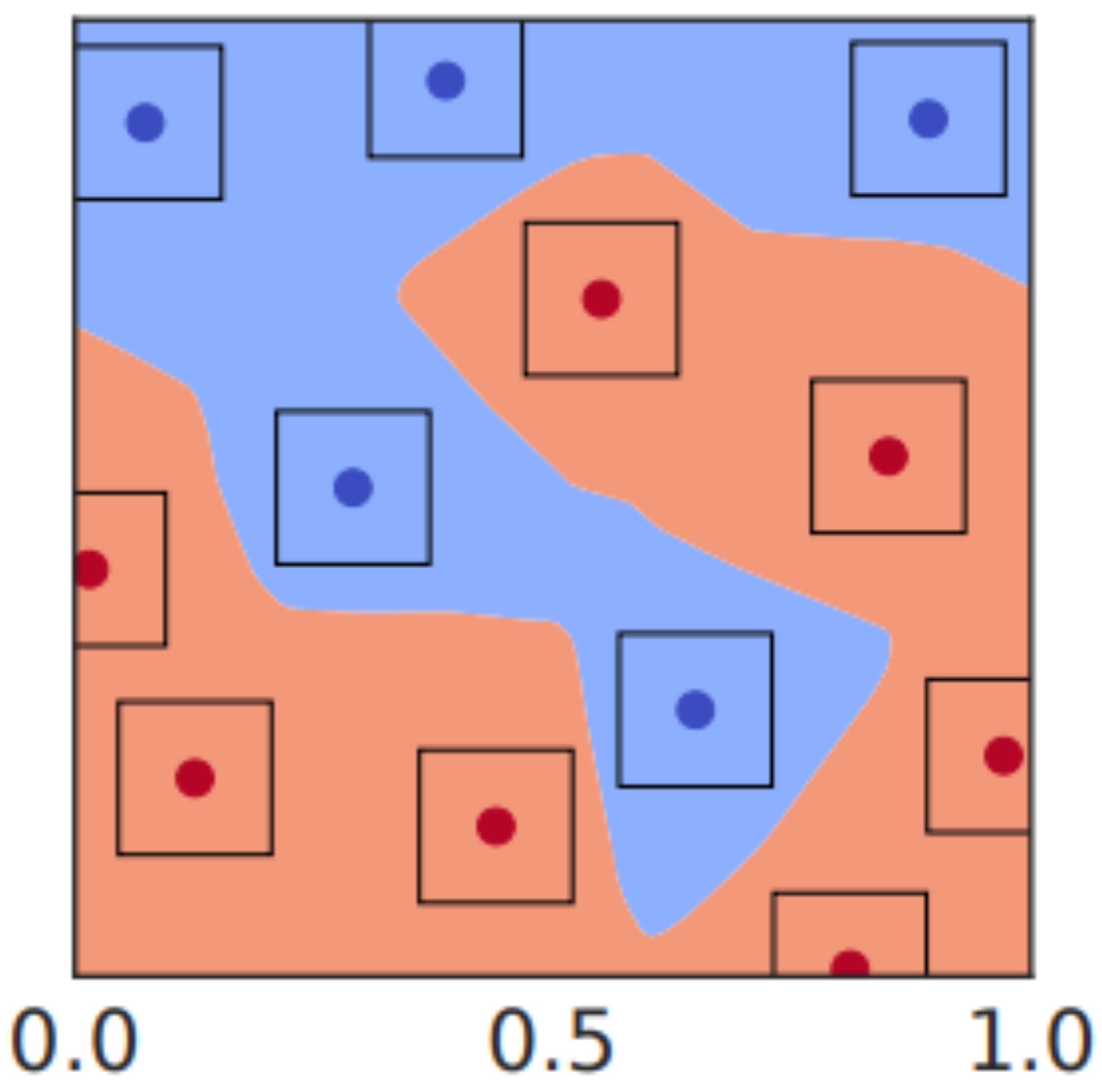
Source: https://openai.com/blog/adversarial-example-research/

Adversarial defenses

- To make the model more robust, we want to push the decision boundary of f further away from the data points



Less robust model
(can easily find adversarial examples within the samples neighborhood)



More robust model
(no adversarial examples within the samples neighborhood)

Wong, Eric, and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope." International Conference on Machine Learning. PMLR, 2018.

Adversarial defenses

- **Method 1**: minimize the worst-case loss by adversarial training

  $$\min_{\theta} \mathbb{E}_{(X,Y)\sim\mathcal{D}} \left[\max_{\delta\in\Delta} L(f(X+\delta;\theta), Y)) \right]$$

  Procedure:

  - find adversarial examples
  - append adversarial examples into the training set
  - train the model with the new training set

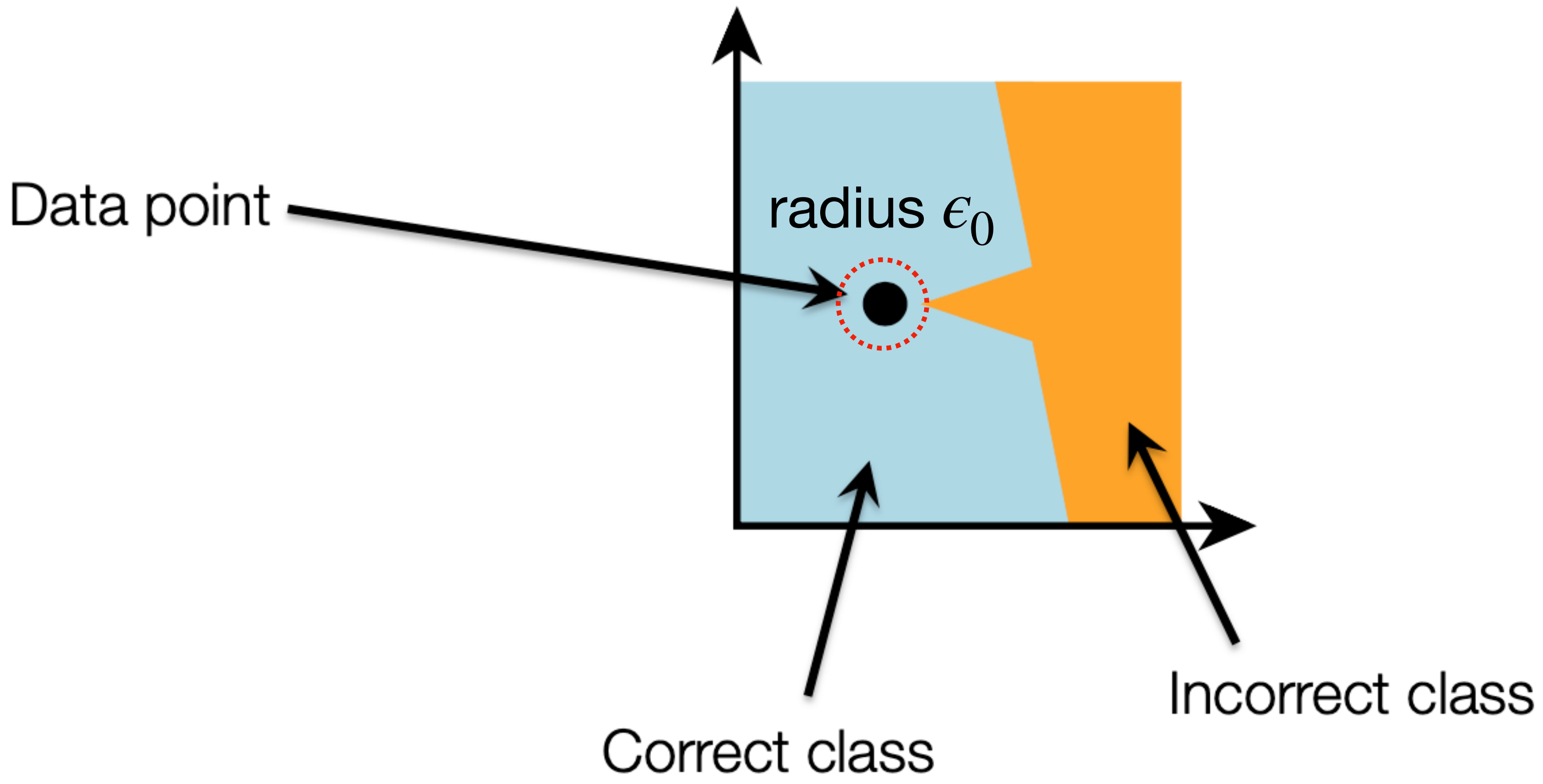
- Challenges: attackers may use many ways to define the loss function, e.g. L-1, L-2, L-∞. Need to cover them all.

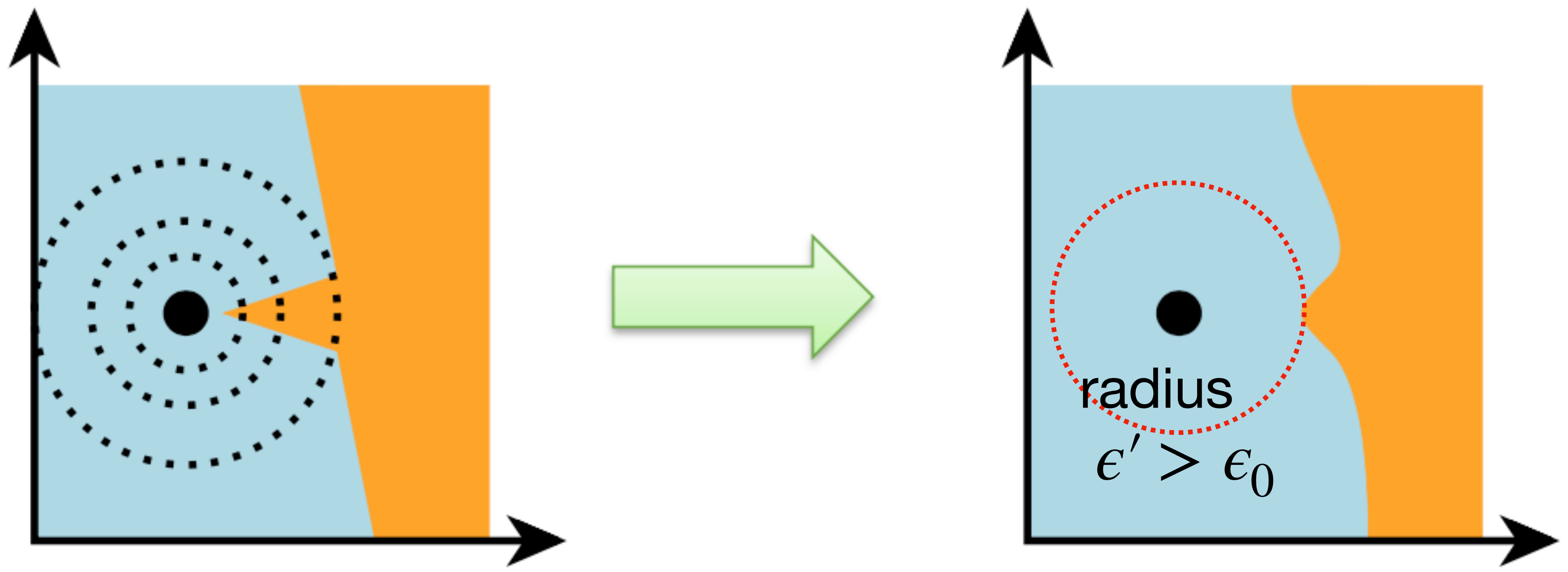
Adversarial defenses

- **Method 2**: regularize using FGSM

$$\tilde{L}(x, y; \theta) = \alpha L(x, y; \theta) + (1 - \alpha) L(\underbrace{x + \epsilon \cdot \text{sign}(\nabla L_x(x, y; \theta))}_{\text{FGSM adv. example}}, y; \theta)$$

- **Method 3**: use randomized smoothing



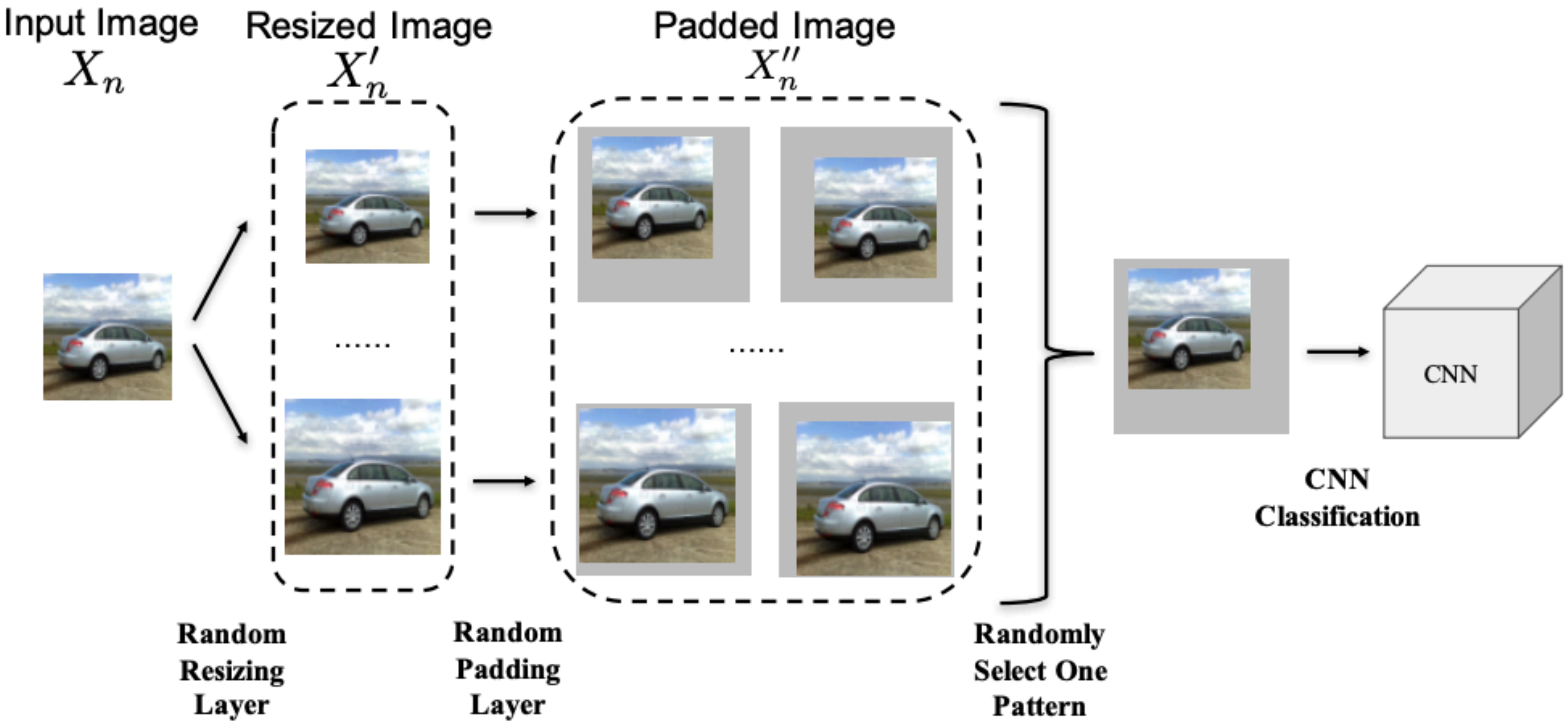


$f(x)$ $\quad$ $g(x) = \underset{y}{\text{argmax}}\, \mathbf{P}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)}[f(x + \epsilon) = y]$

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).
Cohen, Jeremy M., Elan Rosenfeld, and J. Zico Kolter. "Certified adversarial robustness via randomized smoothing." *arXiv preprint arXiv:1902.02918* (2019).

Adversarial defenses

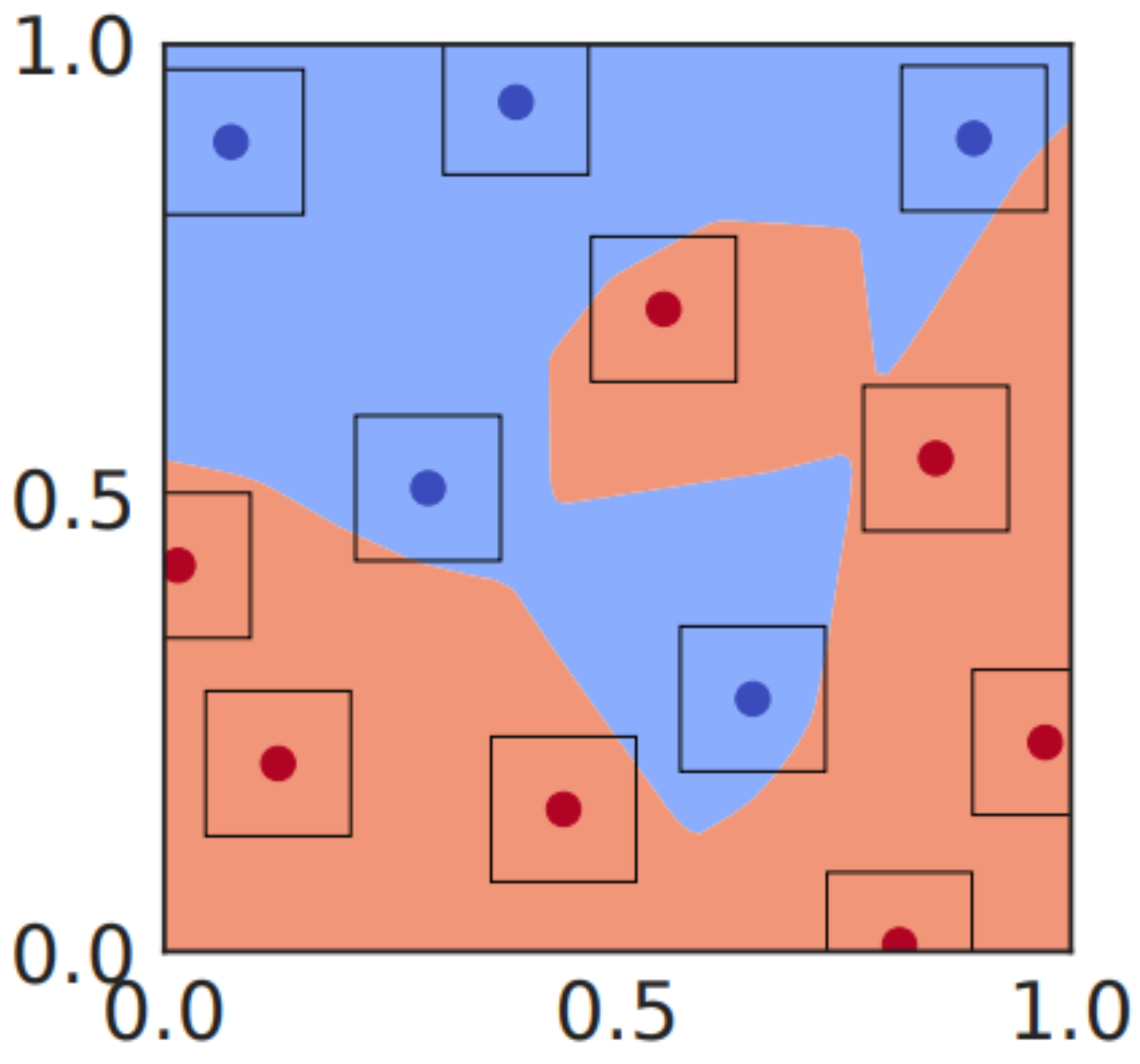
- Smoothing via randomized padding



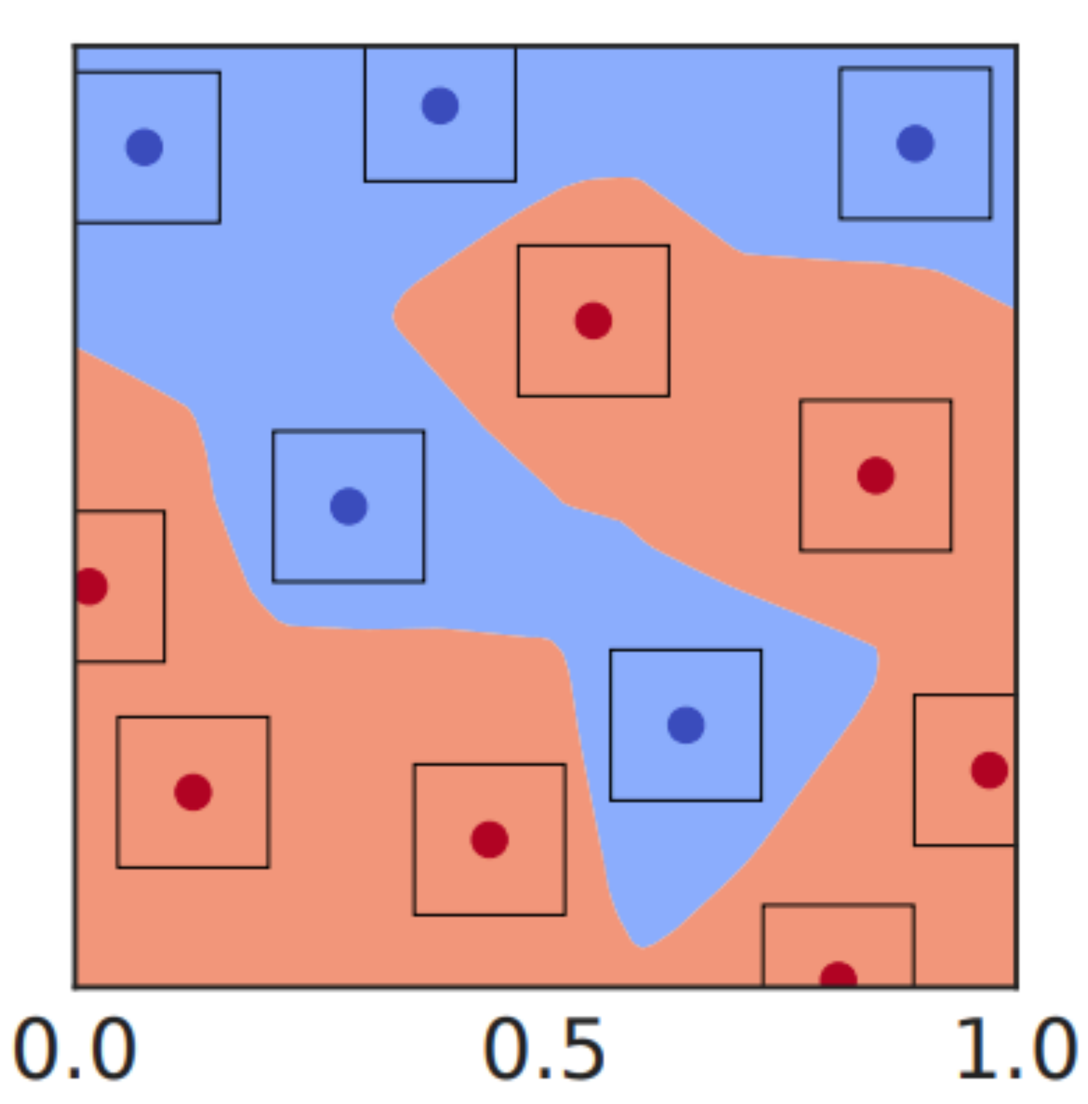
Xie, Cihang, et al. "Mitigating adversarial effects through randomization." *arXiv preprint arXiv:1711.01991* (2017).

Building trustworthy models

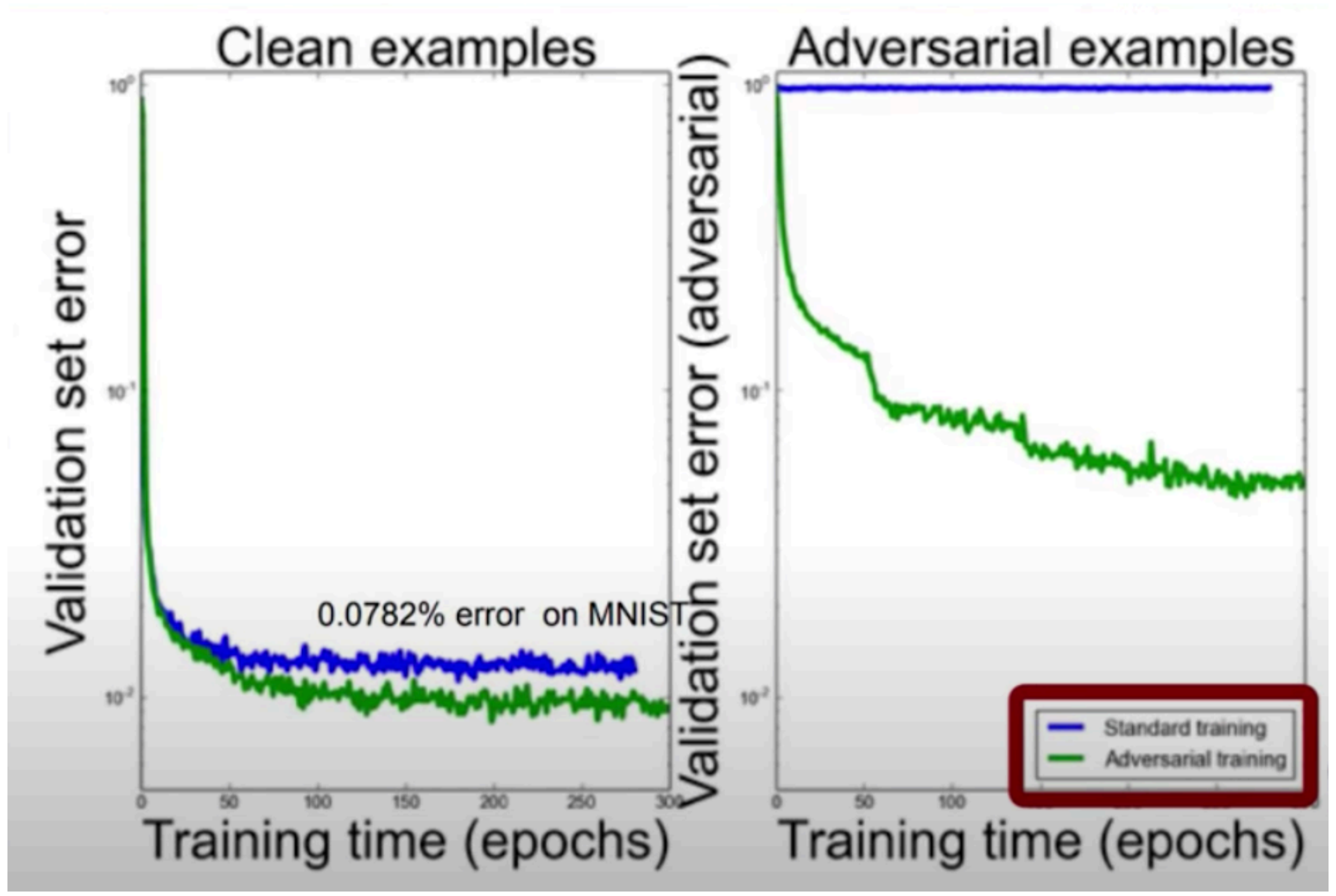
- Adversarial training regularizes and improves and generalization



Regular training: prone to small perturbation



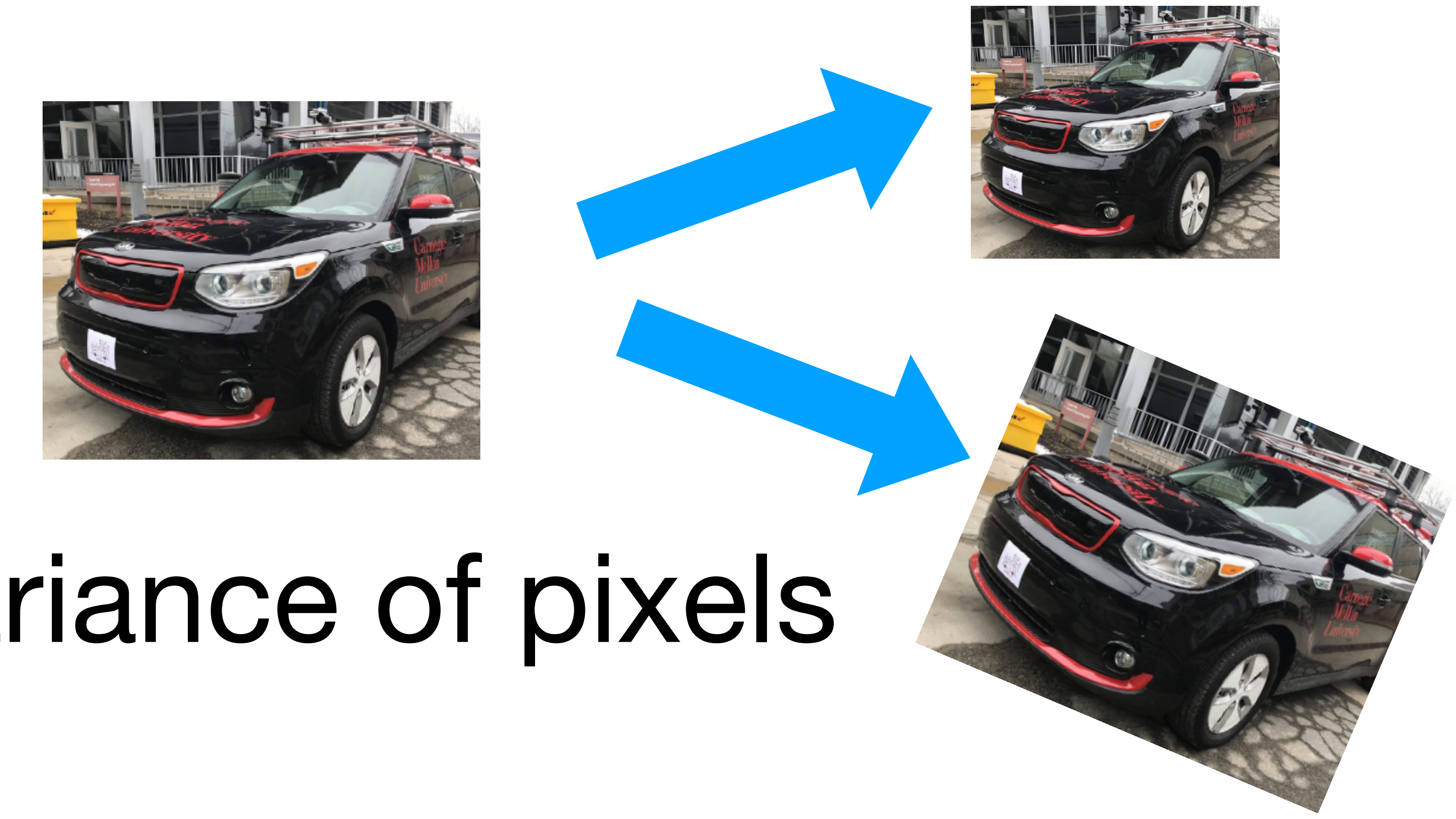
Adversarial training: more regularized and more robust



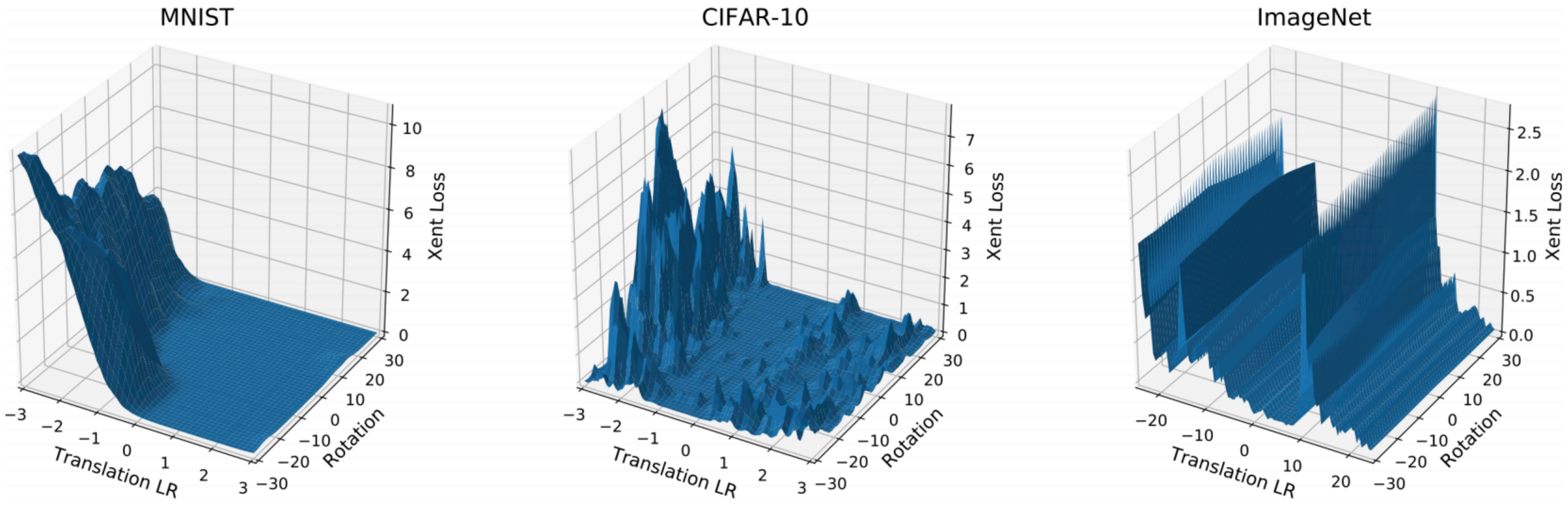
Better generalization observed after adversarial training

Wong, Eric, and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope." International Conference on Machine Learning. PMLR, 2018.

Building trustworthy models



- Adversarial defense can also happen with bigger variance of pixels
  - The loss of could be highly non-smooth w.r.t. translation & rotation parameter space
  - Adversarial training can help smoothen the loss in even in this space



Engstrom, Logan, et al. "Exploring the landscape of spatial robustness." International Conference on Machine Learning. PMLR, 2019.

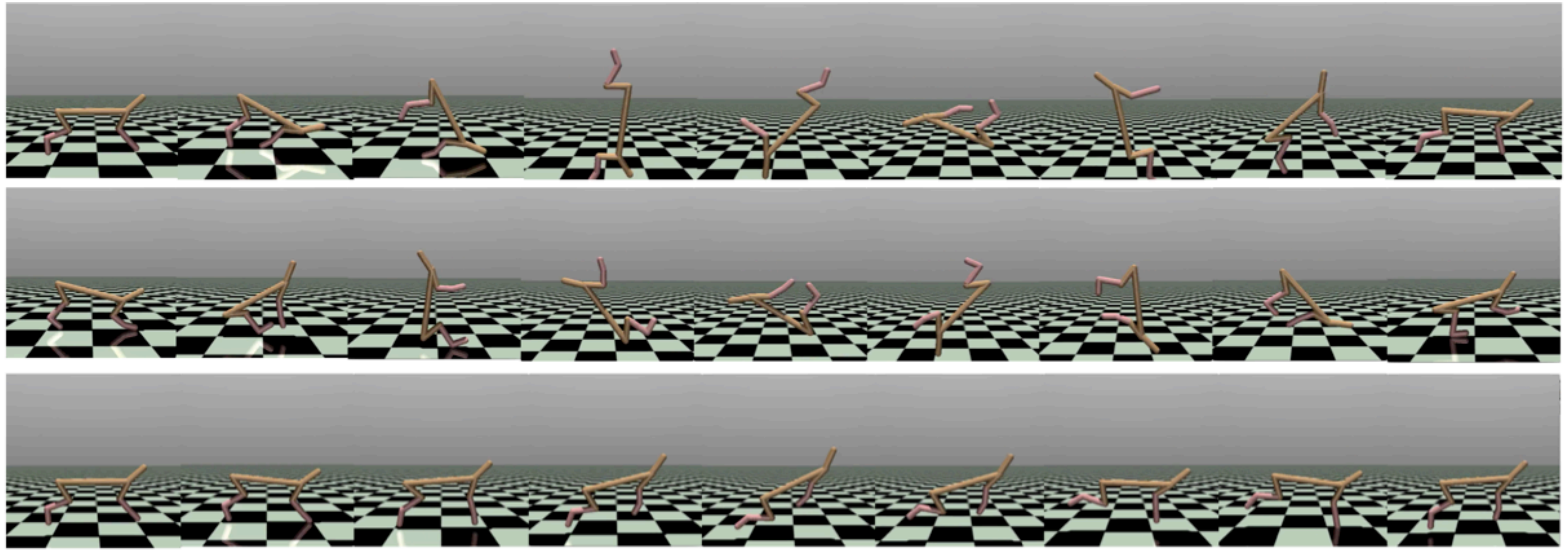
Use of adversarial examples for AI safety

Adversarial examples give us some traction on AI safety

When we think about the study of AI safety, we usually think about some of the most difficult problems in that field — how can we ensure that sophisticated reinforcement learning agents that are significantly more intelligent than human beings behave in ways that their designers intended?

Adversarial examples show us that even simple modern algorithms, for both supervised and reinforcement learning, can already behave in surprising ways that we do not intend.

- Example:
  We will revisit this paper again in the reinforcement learning session



OpenAI, “Attacking Machine Learning with Adversarial Examples”, 2017
Zichuan Lin, Garrett Thomas, Guangwen Yang, and Tengyu Ma. Model-based adversarial meta-reinforcement learning. NeurIPS 2020.

Summary

- Why should we care about adversarial learning?
- Adversarial attack approaches
  - Poisoning, Evasion
  - White-box and black-box attacks
  - *Case study*: adversarial examples in self-driving
- Adversarial defense approaches
  - Building trustworthy models
- Next: Probabilistic robustness

Worth Reading

- Robust physical attack:
  Eykholt, Kevin, et al. "Robust physical-world attacks on deep learning visual classification." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

- Certifiable robustness:
  Cohen, Jeremy M., Elan Rosenfeld, and J. Zico Kolter. "Certified adversarial robustness via randomized smoothing." *arXiv preprint arXiv:1902.02918* (2019).