

Introduction to Adversarial Machine Learning

Bibek Poudel

Sections

- Origin story
- Optimization problem
- Attacks
- Defenses
- Theories

Origin story



Christian Szegedy

[Google Research](#)

Verified email at google.com

Machine learning Computer Vision Artificial Intelligence Automated Reasoning



Christian Szegedy

[Google Research](#)

Verified email at google.com

Machine learning Computer Vision Artificial Intelligence Automated Reasoning

3	4	2	1	9	5	6	2	1	8
8	9	1	2	5	0	0	6	6	4
6	7	0	1	6	3	6	3	7	0
3	7	7	9	4	6	6	1	8	2
2	9	3	4	3	9	8	7	2	5
1	5	9	8	3	6	5	7	2	3
9	3	1	9	1	5	8	0	8	4
5	6	2	6	8	5	8	8	9	9
3	7	7	0	9	4	8	5	4	3
7	9	6	4	7	0	6	9	2	3



Christian Szegedy

[Google Research](#)

Verified email at google.com

Machine learning Computer Vision Artificial Intelligence Automated Reasoning



7



?



?



?



3



Christian Szegedy

[Google Research](#)

Verified email at google.com

Machine learning Computer Vision Artificial Intelligence Automated Reasoning

- Can I craft an optimization problem?





Christian Szegedy

[Google Research](#)

Verified email at google.com

Machine learning Computer Vision Artificial Intelligence Automated Reasoning

- Can I craft an optimization problem?

- ▶ Looks like a 7 to human eye
- ▶ But a model thinks its a 3





Christian Szegedy

[Google Research](#)

Verified email at google.com

Machine learning Computer Vision Artificial Intelligence Automated Reasoning

- “Intriguing properties of neural networks”
 - ▶ ICLR 2014, ~ 9000 citations
 - ▶ Birth of Adversarial Machine Learning (AML)



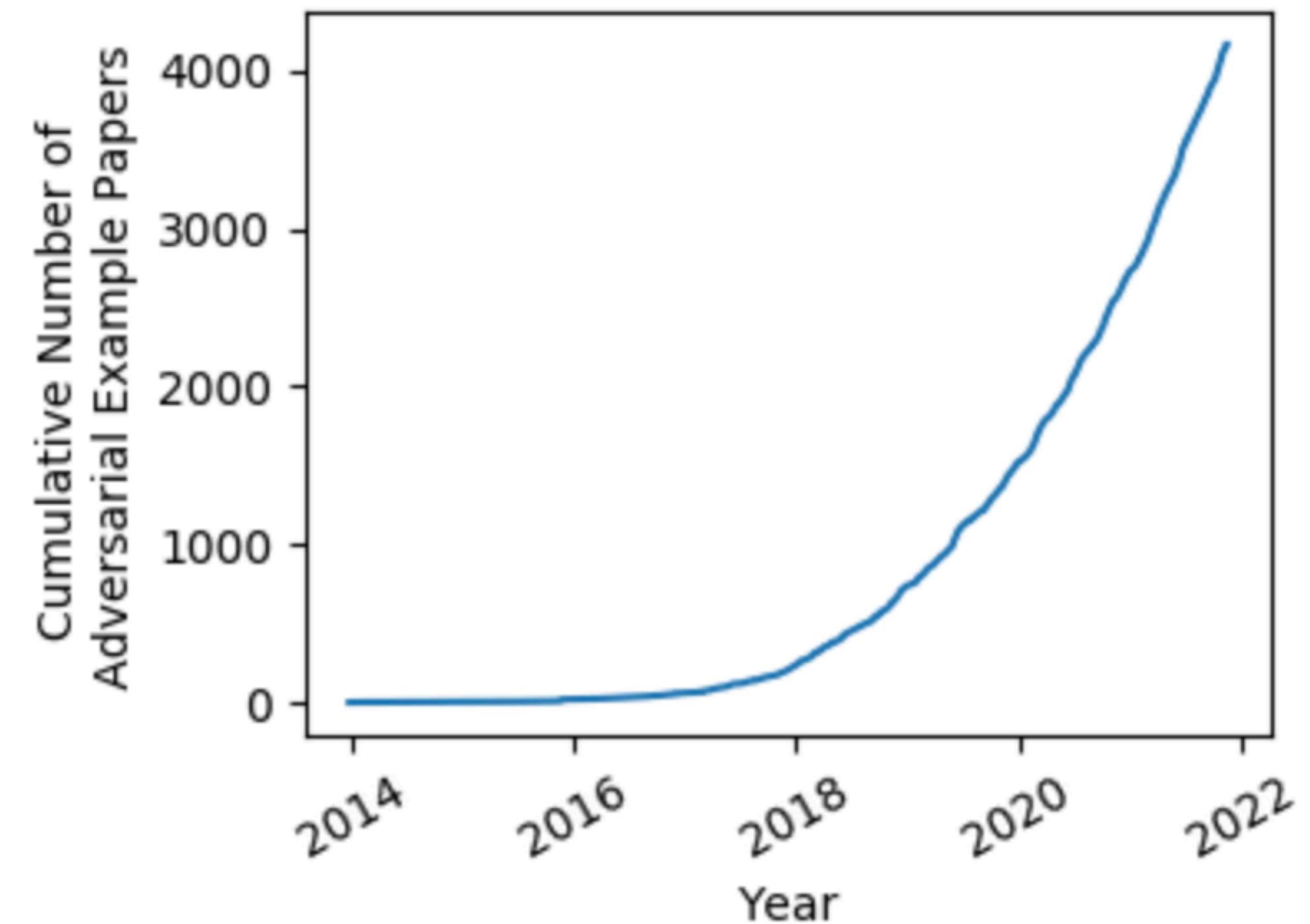
Christian Szegedy

[Google Research](#)

Verified email at google.com

Machine learning Computer Vision Artificial Intelligence Automated Reasoning

- Recent interest in AML



Adversarial examples in action

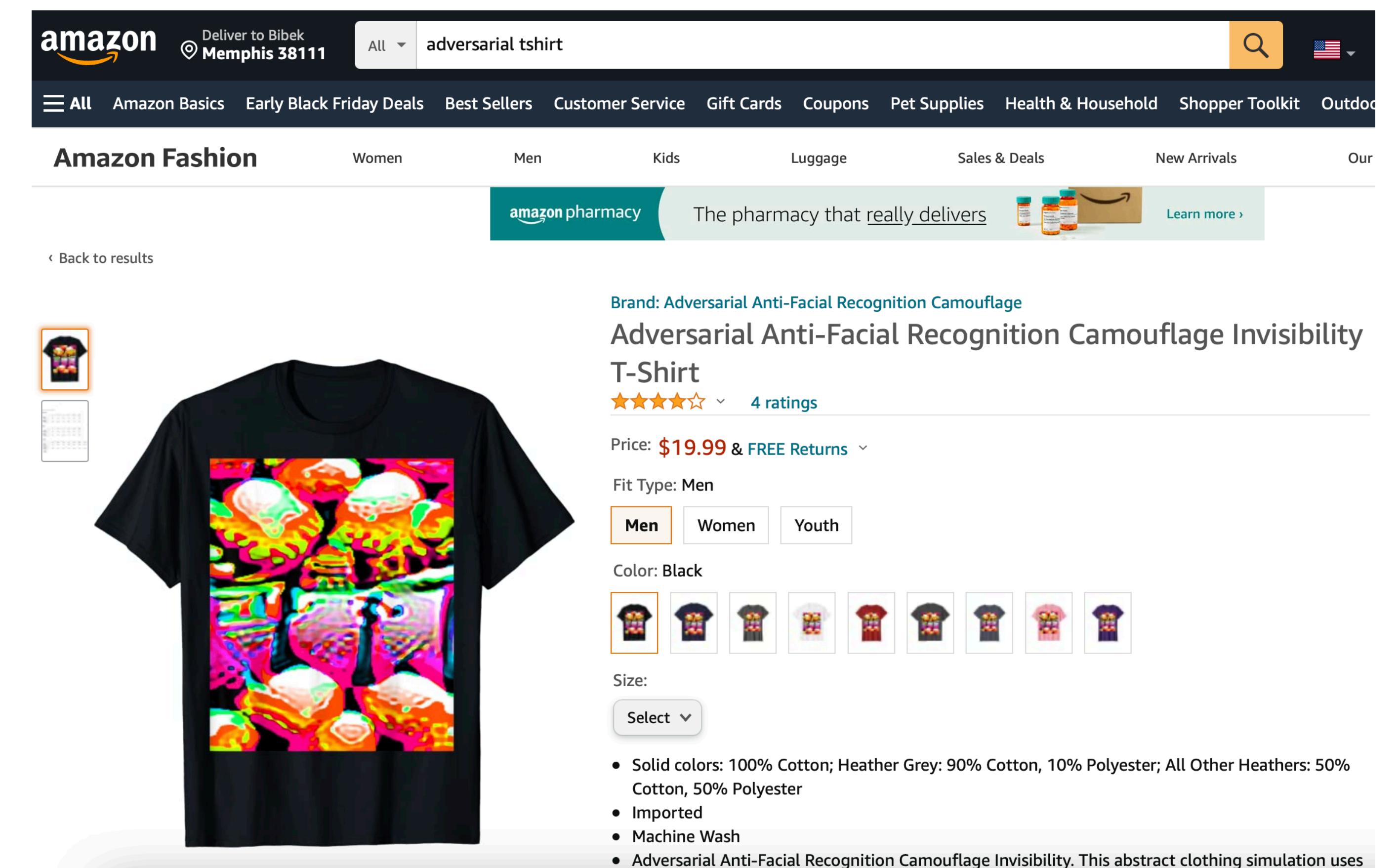
Adversarial examples in action

- Autonomous driving and traffic signs
- Video



Adversarial examples in action

- Surveillance, facial recognition
- Video



Adversarial examples in action

- Reinforcement learning
- Video

Optimization problem

The optimization problem

- “L_p norm” distance metric



Image 1



Image 2

The optimization problem

- “L_p norm” distance metric

37	128	64
18	220	59
100	50	33

Image 1

38	128	64
18	99	59
100	50	33

Image 2

The optimization problem

- “L_p norm” distance metric
 - ▶ L₀ distance = 2

37	128	64
18	220	59
100	50	33

Image 1

38	128	64
18	99	59
100	50	33

Image 2

The optimization problem

- “L_p norm” distance metric
 - ▶ L₁ distance = |37 - 38| + |220 - 99|

37	128	64
18	220	59
100	50	33

Image 1

38	128	64
18	99	59
100	50	33

Image 2

The optimization problem

- “L_p norm” distance metric
 - ▶ L₂ distance = $(37 - 38)^2 + (220 - 99)^2$

37	128	64
18	220	59
100	50	33

Image 1

38	128	64
18	99	59
100	50	33

Image 2

The optimization problem

- “L_p norm” distance metric
 - ▶ L[∞] distance = (220 - 99) , max difference

37	128	64
18	220	59
100	50	33

Image 1

38	128	64
18	99	59
100	50	33

Image 2

The optimization problem

- Objective + constraints

The optimization problem

- Objective + constraints

$$\text{minimize } D(x, x + \delta_x)$$

The optimization problem

- Objective + constraints

$$\text{minimize } D(x, x + \delta_x)$$

subject to:

$$\begin{aligned} f(x) &\neq f(x + \delta_x) \\ x + \delta_x &\in [0, 1]^n \end{aligned}$$

Attacks

Attacks

- Fast Gradient Sign Method (FGSM)
- “Explaining and harnessing adversarial examples”, Goodfellow et. al. 2015

Attacks

- Fast Gradient Sign Method (FGSM)
- “Explaining and harnessing adversarial examples”, Goodfellow et. al. 2015

$$\begin{array}{ccc}
 \text{---} & + .007 \times & \text{---} \\
 \text{---} & \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)) & = \\
 \text{---} & \text{“nematode”} & \text{---} \\
 \text{---} & 8.2\% \text{ confidence} & \text{---} \\
 \boldsymbol{x} & & \boldsymbol{x} + \\
 \text{“panda”} & & \epsilon \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)) \\
 57.7\% \text{ confidence} & & \text{“gibbon”} \\
 & & 99.3 \% \text{ confidence}
 \end{array}$$

The diagram illustrates the FGSM attack process. It shows a original image of a panda (\boldsymbol{x}) with 57.7% confidence in being a "panda". This image is combined with a scaled gradient sign ($+ .007 \times \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$) of a "nematode" image, which has 8.2% confidence. The result is a adversarial image ($\boldsymbol{x} + \epsilon \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$) with 99.3% confidence in being a "gibbon".

Attacks

- Fast Gradient Sign Method (FGSM)

$$x_{adv} = x + \delta$$

$$\delta = \epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))$$

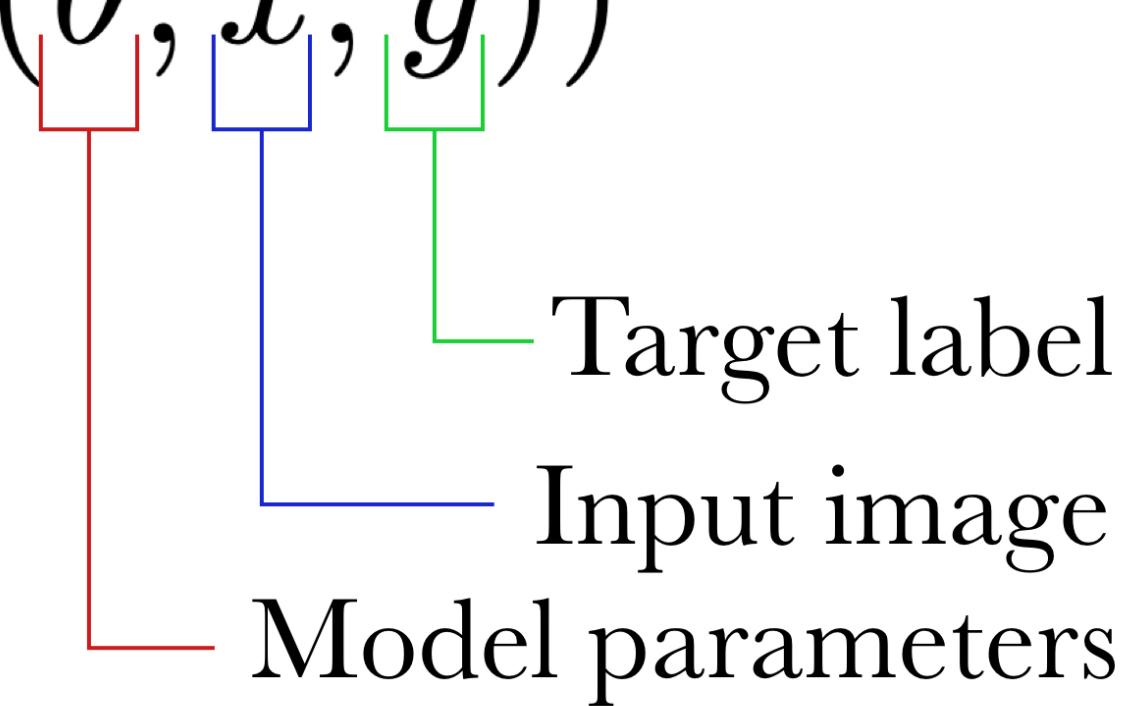
Attacks

- Fast Gradient Sign Method (FGSM)

$$\delta = \epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))$$

Attacks

- Fast Gradient Sign Method (FGSM)

$$\delta = \epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))$$


The diagram illustrates the components of the FGSM formula. It shows three colored brackets pointing to different parts of the equation: a red bracket under the term ∇_x , a blue bracket under the term $J(\theta, x, y)$, and a green bracket under the term sign . Below each bracket is a corresponding label: 'Target label' for the red bracket, 'Input image' for the blue bracket, and 'Model parameters' for the green bracket.

Attacks

- Fast Gradient Sign Method (FGSM)

$$\delta = \epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))$$

The diagram illustrates the components of the FGSM equation. A central expression is $\delta = \epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))$. Four colored lines point to specific parts of the expression: a red line points to θ (Model parameters), a blue line points to x (Input image), a green line points to y (Target label), and an orange line points to $J(\theta, x, y)$ (Loss value).

Attacks

- Fast Gradient Sign Method (FGSM)

$$\delta = \epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))$$

Gradient w.r.t.
input

Loss value

Target label

Input image

Model parameters

The diagram shows the components of the FGSM formula. The gradient term ∇_x is highlighted with a green bracket and labeled "Gradient w.r.t. input". The loss function $J(\theta, x, y)$ is highlighted with a yellow bracket and labeled "Loss value". The target label y is highlighted with a red bracket and labeled "Target label". The input image x is highlighted with a blue bracket and labeled "Input image". The model parameters θ is highlighted with a black bracket and labeled "Model parameters".

Attacks

- Fast Gradient Sign Method (FGSM)

$$\delta = \epsilon \text{ sign}(\nabla_x J(\theta, x, y))$$

Gradient w.r.t.
input

Loss value

Just take
the sign

Target label

Input image

Model parameters

The diagram shows the FGSM formula $\delta = \epsilon \text{ sign}(\nabla_x J(\theta, x, y))$ with annotations pointing to its components:

- Gradient w.r.t. input: Points to the term ∇_x .
- Loss value: Points to the term $J(\theta, x, y)$.
- Just take the sign: A note pointing to the $\text{sign}()$ function.
- Target label: Points to the term y .
- Input image: Points to the term x .
- Model parameters: Points to the term θ .

Attacks

- Fast Gradient Sign Method (FGSM)

$$\delta = \epsilon \text{sign}(\nabla_x J(\theta, x, y))$$

Gradient w.r.t.
input

Loss value

Scale

Just take
the sign

Target label

Input image

Model parameters

The diagram shows the FGSM formula $\delta = \epsilon \text{sign}(\nabla_x J(\theta, x, y))$ with annotations explaining its components. The 'Scale' is ϵ , the 'Just take the sign' part is $\text{sign}(\nabla_x J(\theta, x, y))$, and the gradient term is $\nabla_x J(\theta, x, y)$. The legend identifies the colors for each component: green for Gradient w.r.t. input and Loss value, orange for Loss value, white for Scale, pink for Just take the sign, blue for Input image, and red for Model parameters.

Attacks

- Projected Gradient Descent (PGD)

Attacks

- Projected Gradient Descent (PGD)
 - ▶ Add random noise + take multiple smaller FGSM steps
 - ▶ Iterative

Attacks

- Projected Gradient Descent (PGD)
 - ▶ Add random noise + take multiple smaller FGSM steps
 - ▶ Iterative



Input



FGSM



PGD

Attacks

- One pixel attack



SHIP
CAR(99.7%)



HORSE
FROG(99.9%)



DEER
AIRPLANE(85.3%)



HORSE
DOG(70.7%)



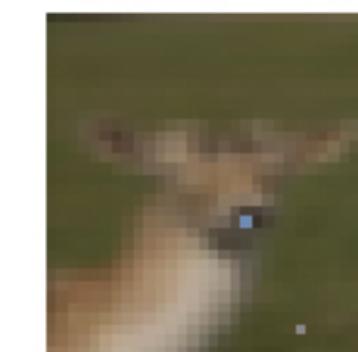
DOG
CAT(75.5%)



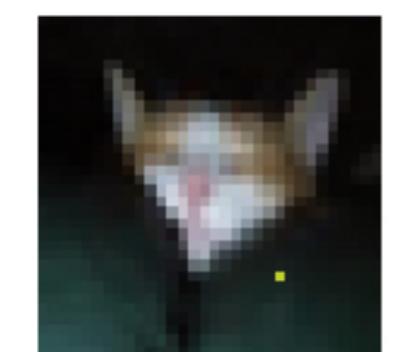
BIRD
FROG(86.5%)



CAR
AIRPLANE(82.4%)



DEER
DOG(86.4%)



CAT
BIRD(66.2%)



DEER
AIRPLANE(49.8%)



BIRD
FROG(88.8%)

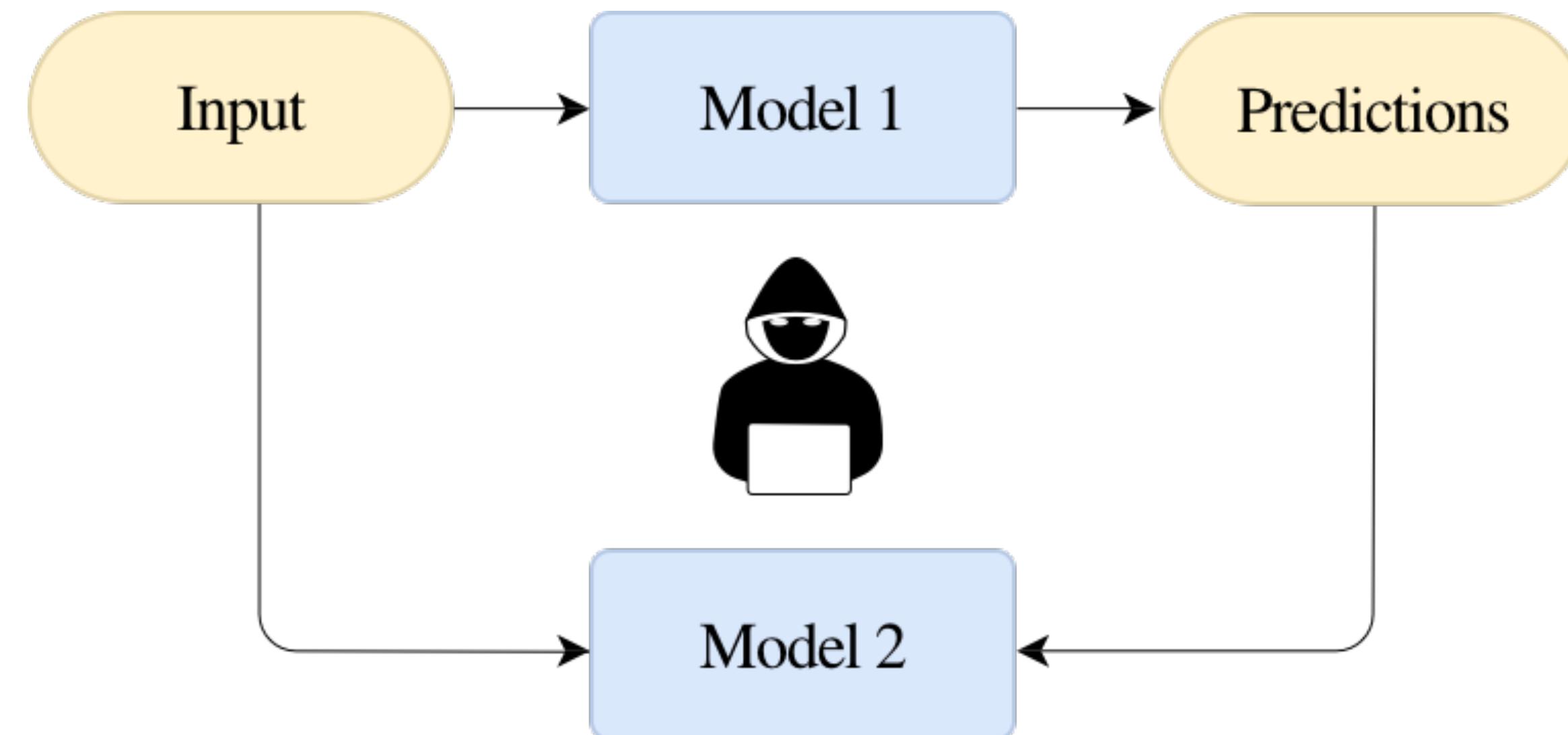


SHIP
AIRPLANE(88.2%)

Threat models

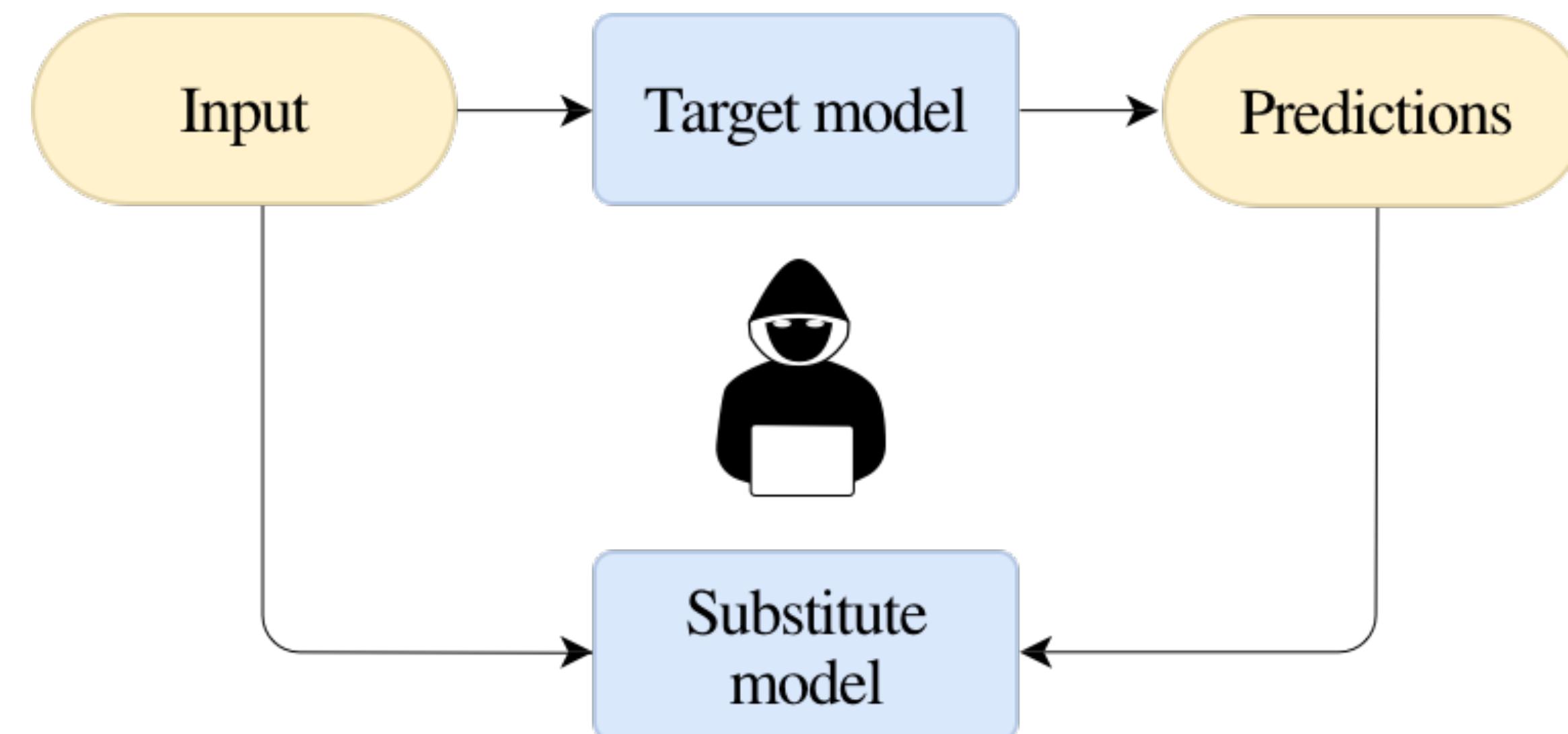
Threat models

- Black-box



Threat models

- Black-box



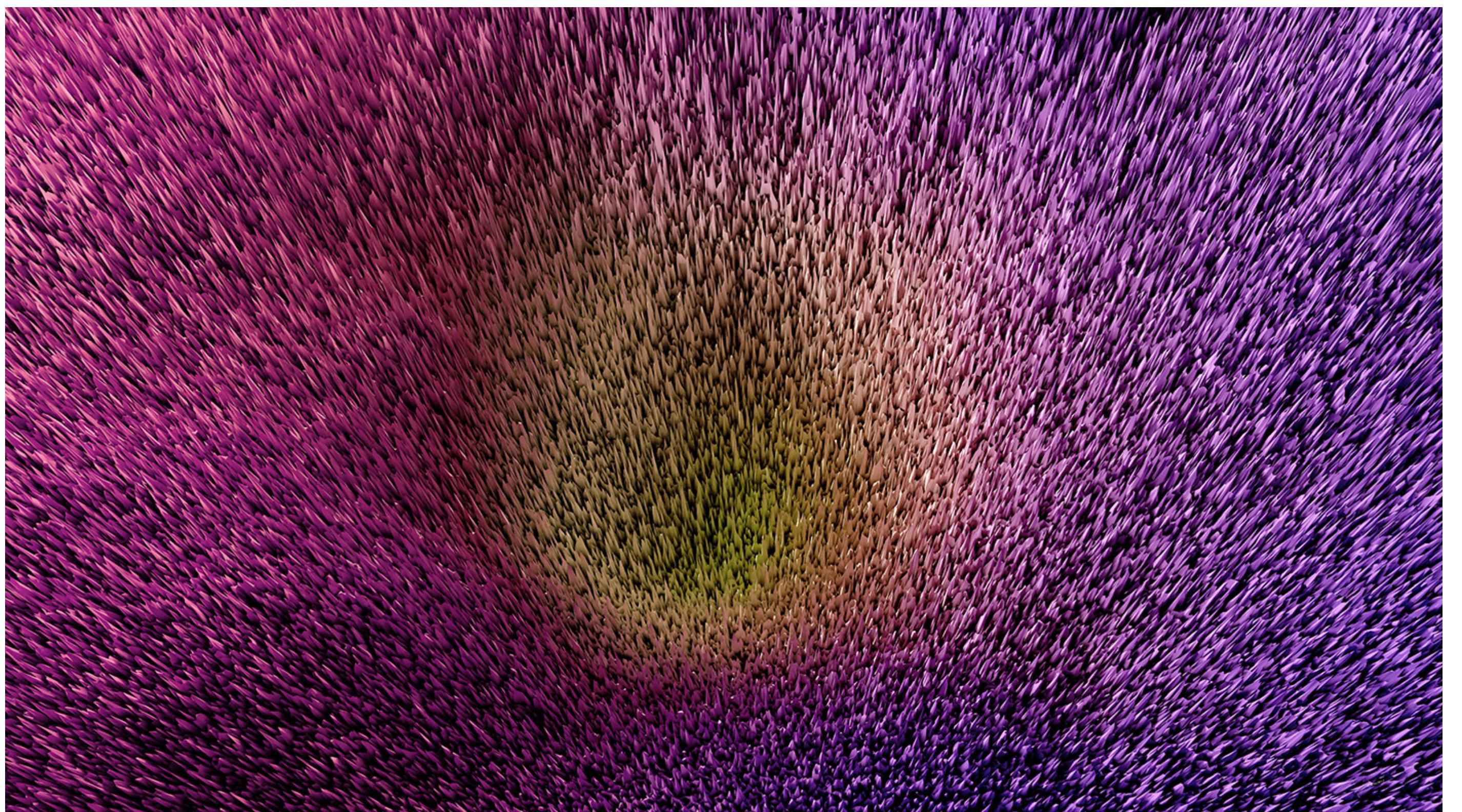
Threat models

- White-box
 - ▶ Training data, hyper-parameters, model architecture

Defenses

Defenses

- Gradient Masking
 - ▶ Hide gradient information
 - ▶ Discarded



Defenses

- Adversarial Training
 - ▶ Most successful

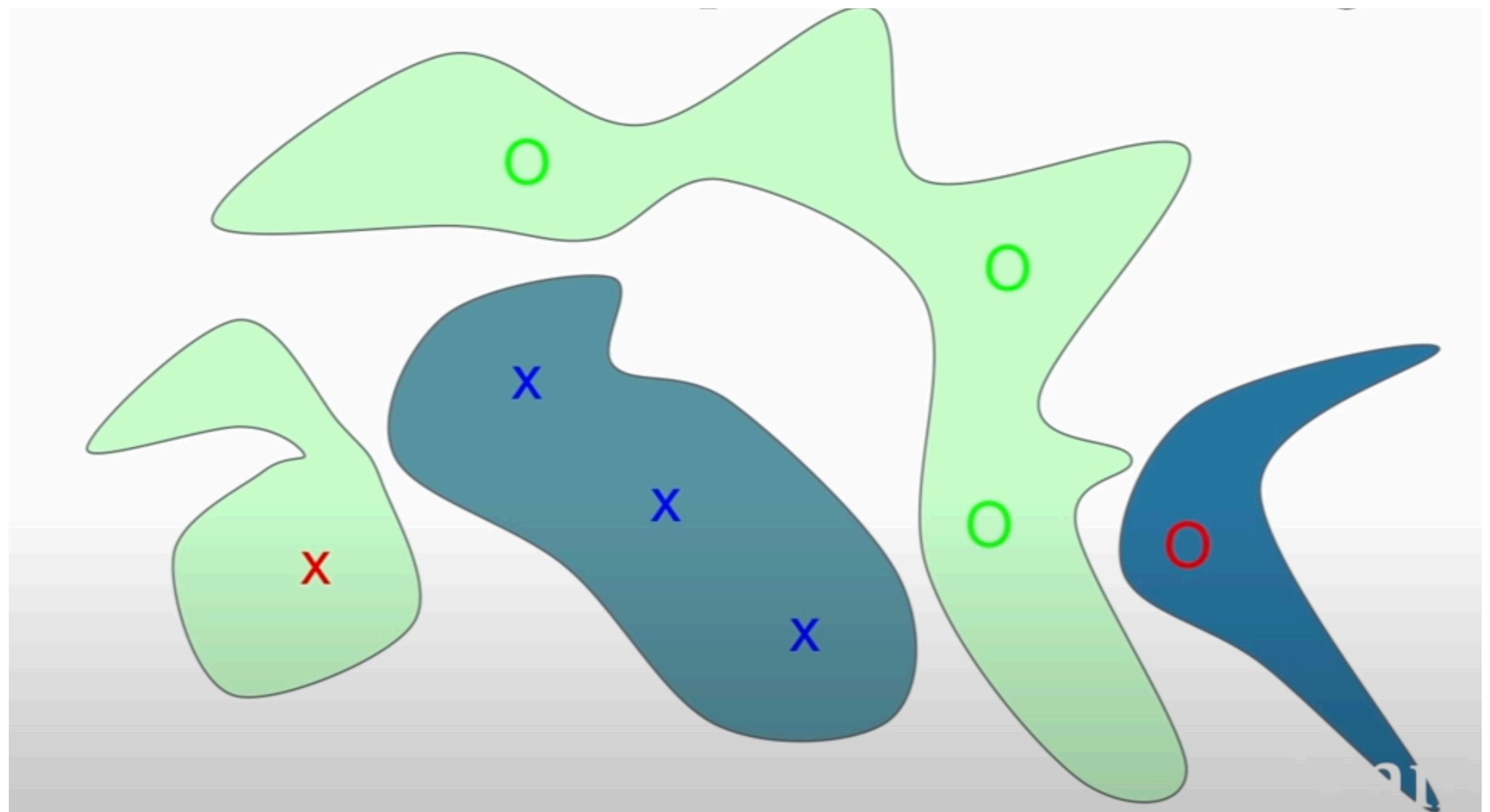
Theories

Theories

- Intuitively make sense but discarded
 - ▶ Overfitting

Theories

- Intuitively make sense but discarded
 - ▶ Overfitting



Theories

- Intuitively make sense but discarded
 - ▶ Excessive linearity

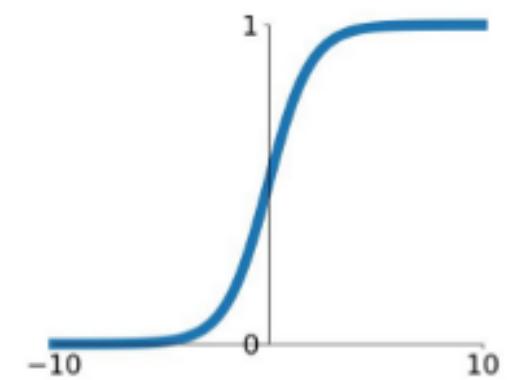
Theories

- Intuitively make sense but discarded
 - ▶ Excessive linearity

Activation Functions

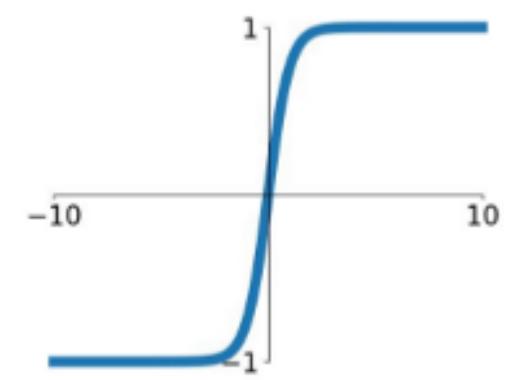
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



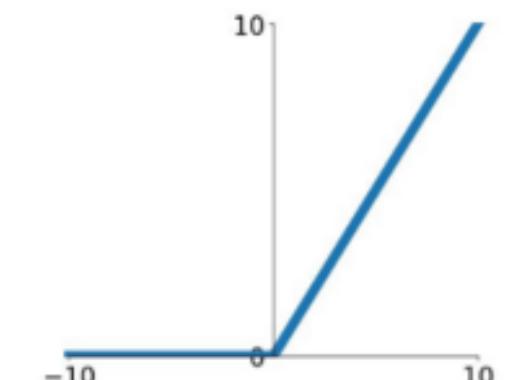
tanh

$$\tanh(x)$$



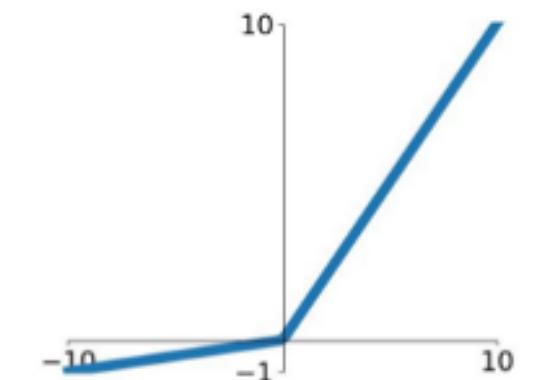
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

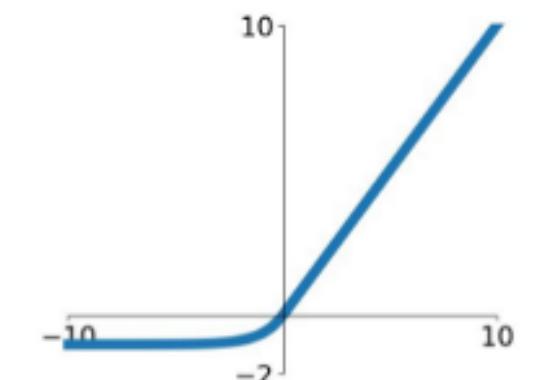


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

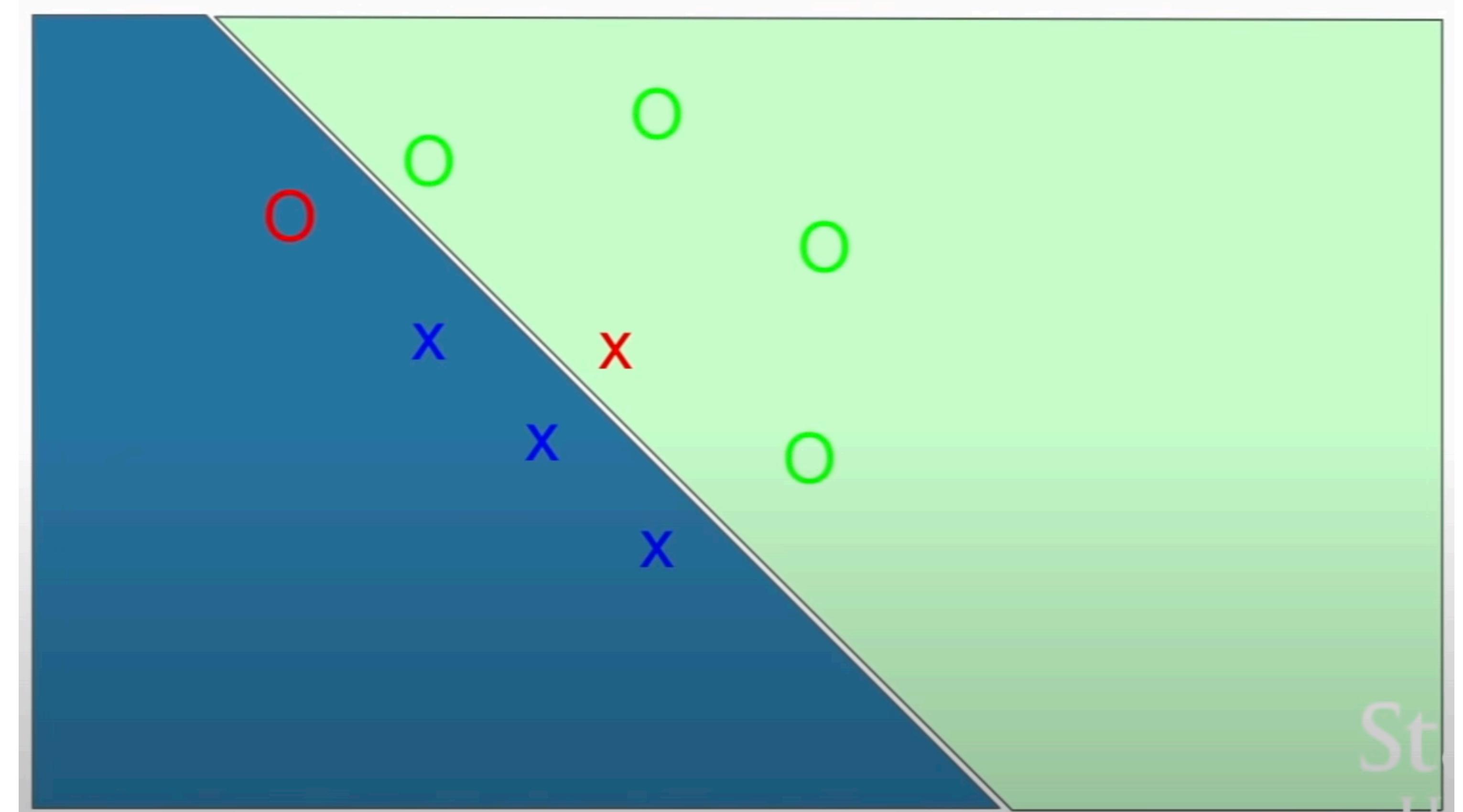
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Theories

- Intuitively make sense but discarded
 - ▶ Excessive linearity

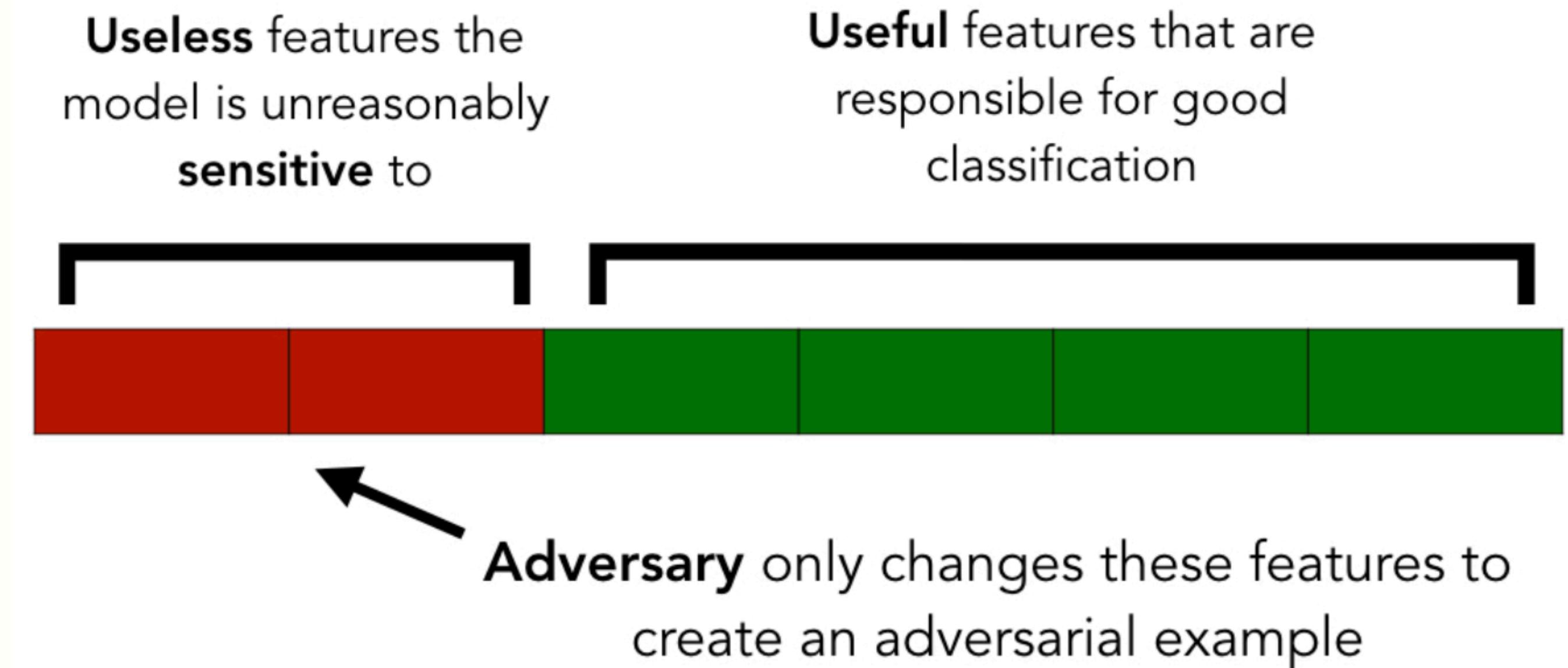


Theories

- Intuitively make sense but discarded
 - ▶ Adversarial examples are bugs

Theories

- Intuitively make sense but discarded
 - ▶ Adversarial examples are bugs



Theories

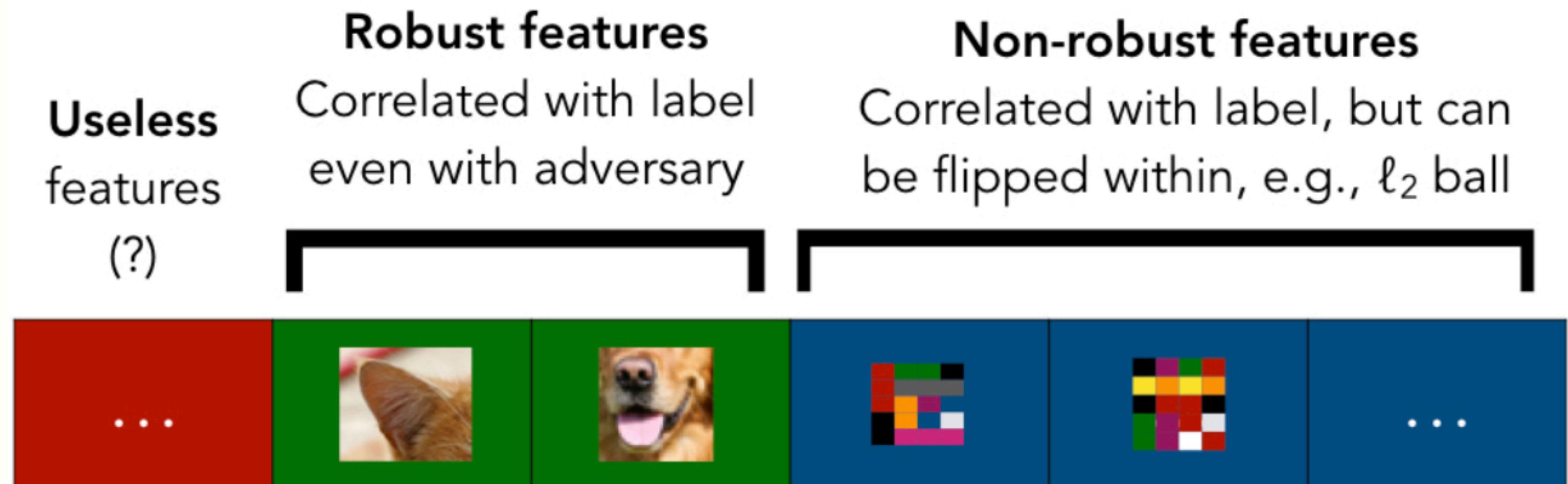
- Widely accepted

Theories

- Widely accepted
 - ▶ “Adversarial examples are not bugs, they are features” Illyas et. al. 2017

Theories

- Widely accepted
 - ▶ “Adversarial examples are not bugs, they are features” Illyas et. al. 2017



A more fundamental question

A more fundamental question

- Do our models really “learn”?

A more fundamental question

- Do our models really “learn”?
- Does the industry care about AEs? [Video](#)

Thank You!

Bibek Poudel
bpoudel@memphis.edu

But wait... there's more...

AI Camera Ruins Soccer Game For Fans After Mistaking Referee's Bald Head For Ball

69.7K
SHARES

[f Share on Facebook](#)

[Share on Twitter](#)

+

