香港城市大學
City University of Hong Kong

CS 6290

Privacy-enhancing Technologies

Department of Computer Science

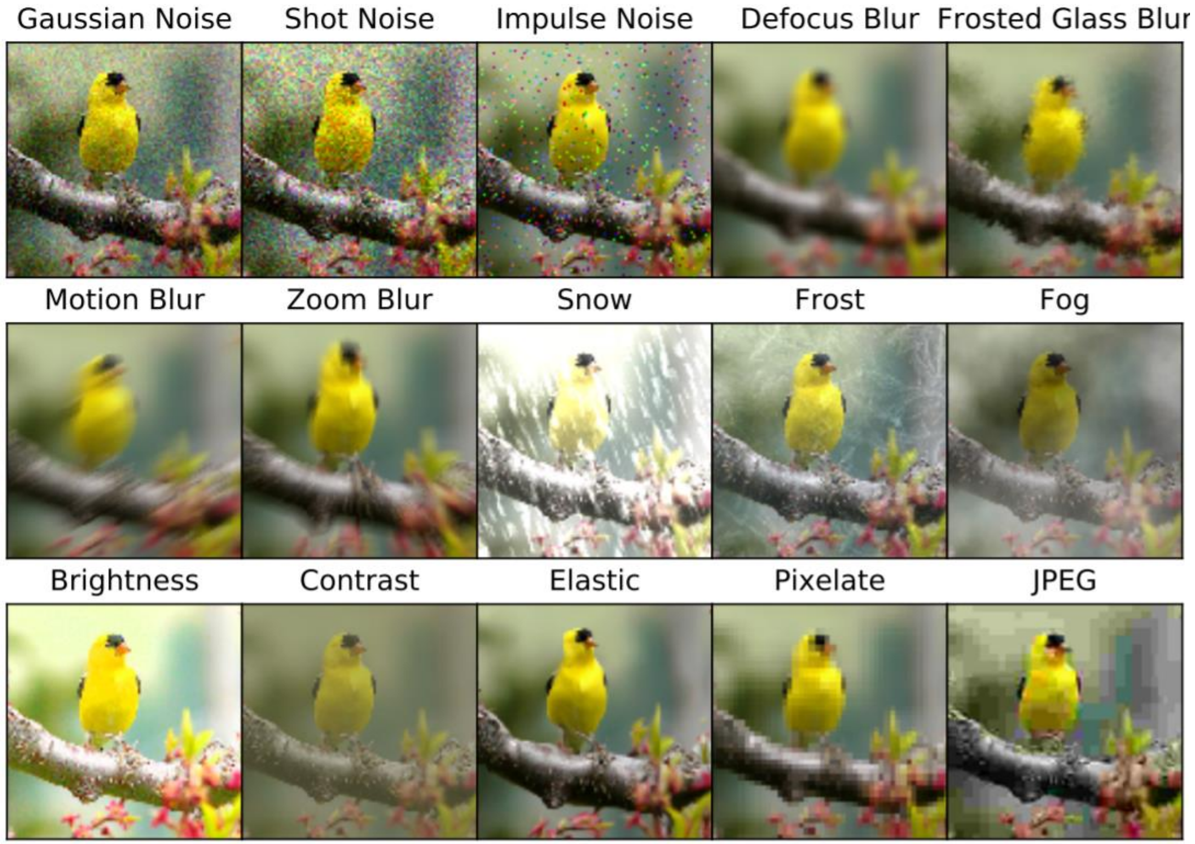
Slides credit in part from Zhengyu Zhao, Zico Kolter

Tutorial 11 – Adversarial Examples and Defenses

Yufei CHEN

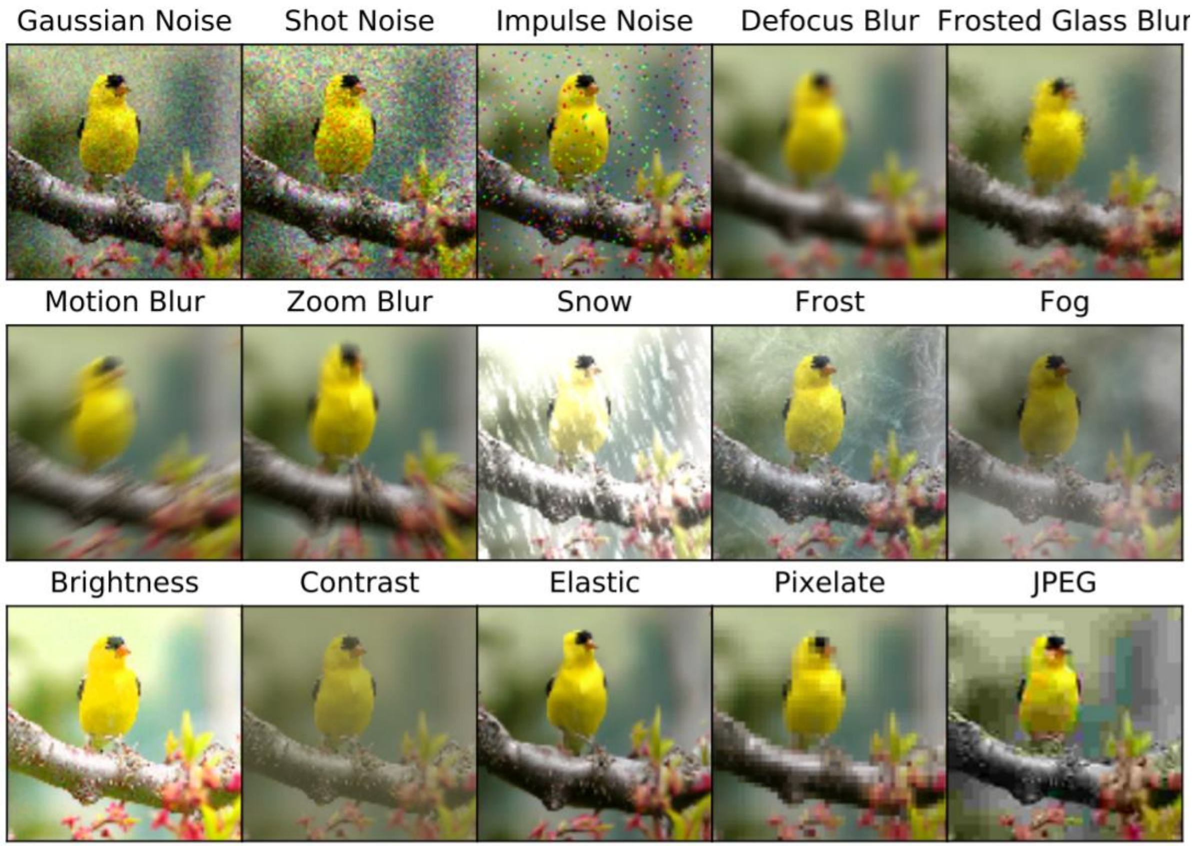
CS Department
City University of Hong Kong

Noisy Examples



Average-case

Noisy Examples → Adversarial Examples

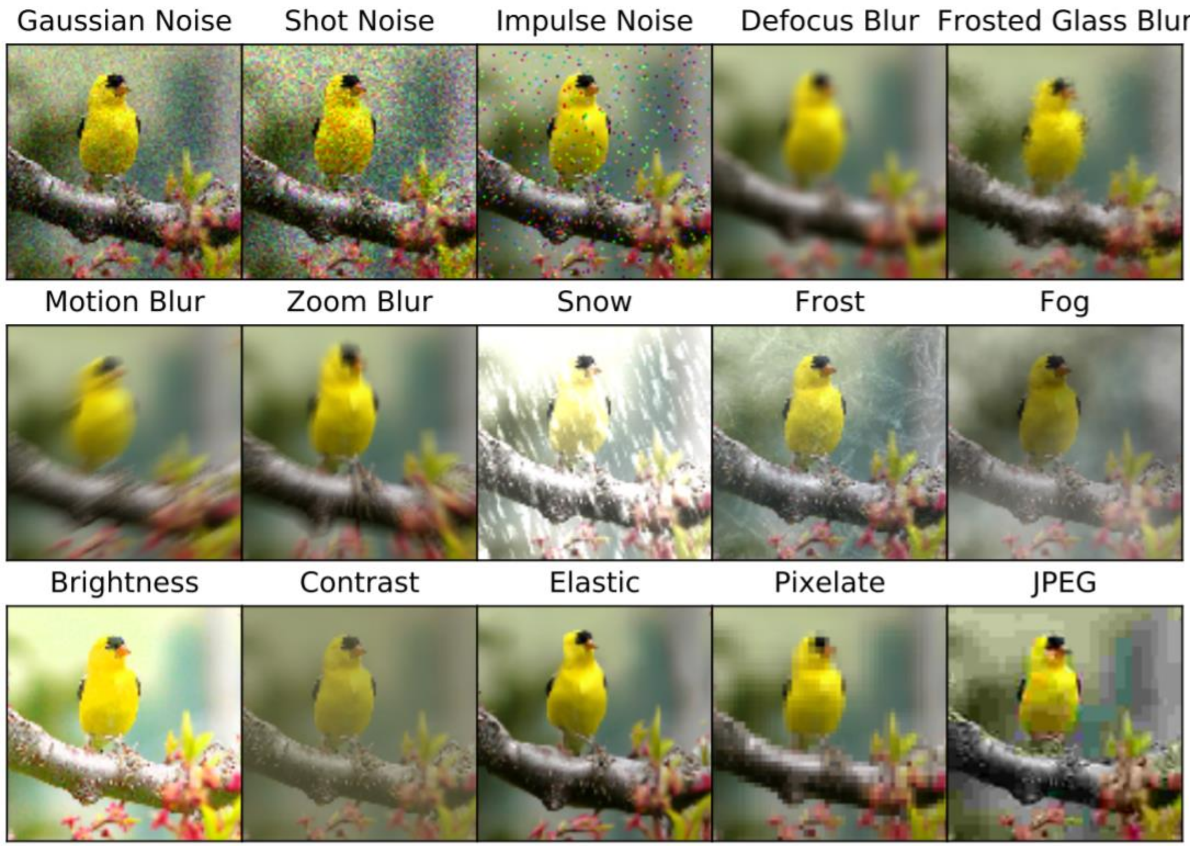


Average-case

→

Worst-case
(optimized)

Noisy Examples → Adversarial Examples



Average-case

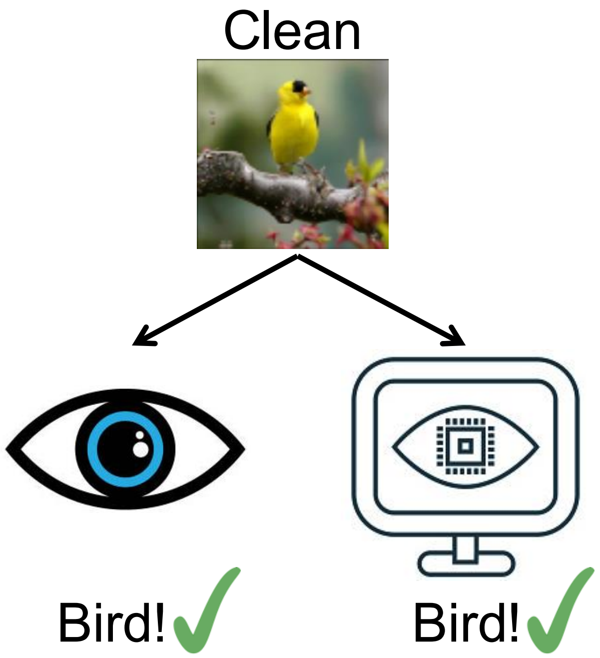
2013
Christian Szegedy, Ian Goodfellow
Intriguing properties of neural networks

2014
Ian Goodfellow
Explaining and harnessing adversarial examples

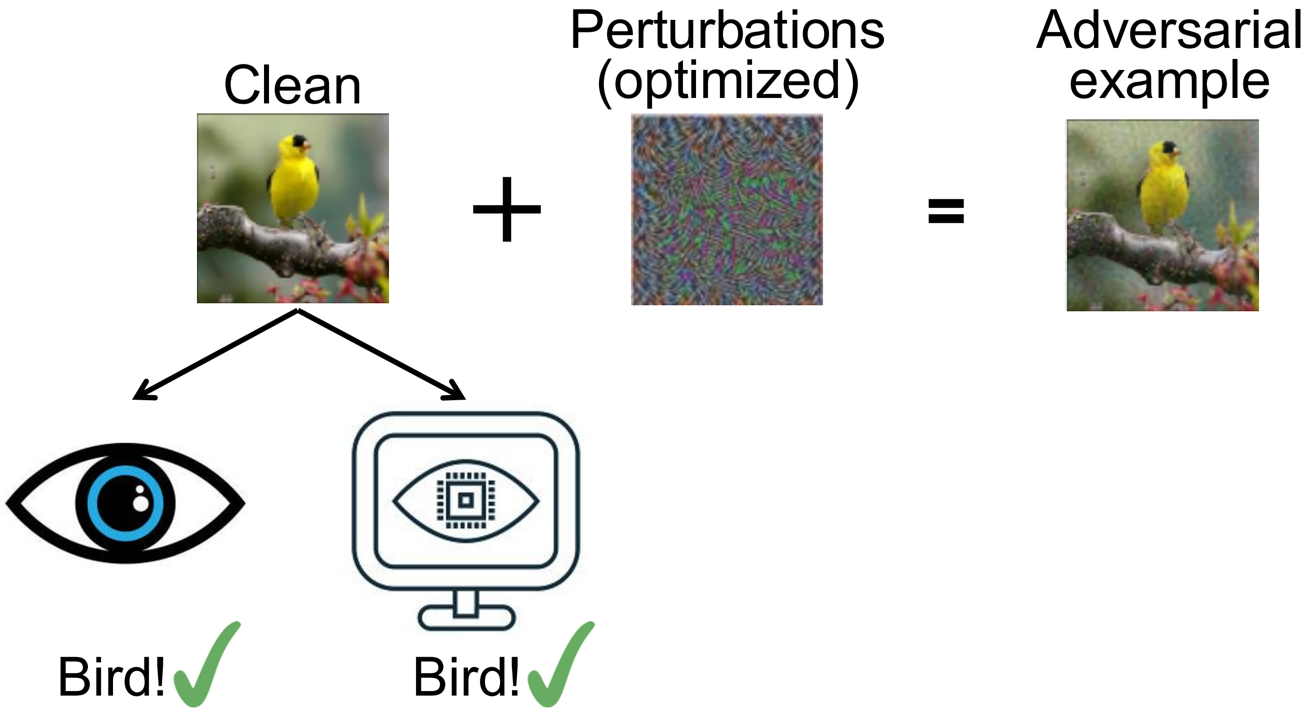
→

Worst-case (optimized)

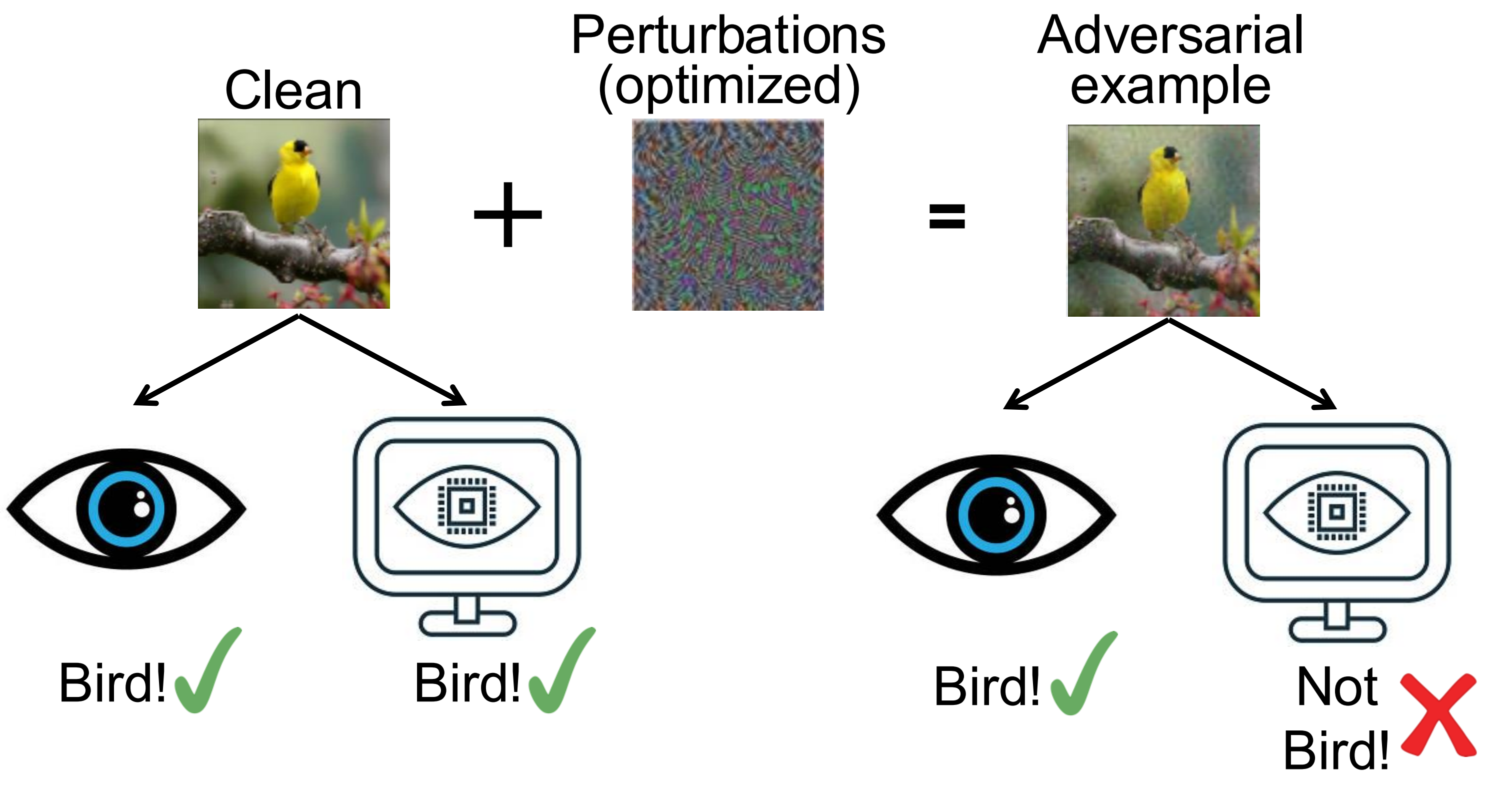
Definition



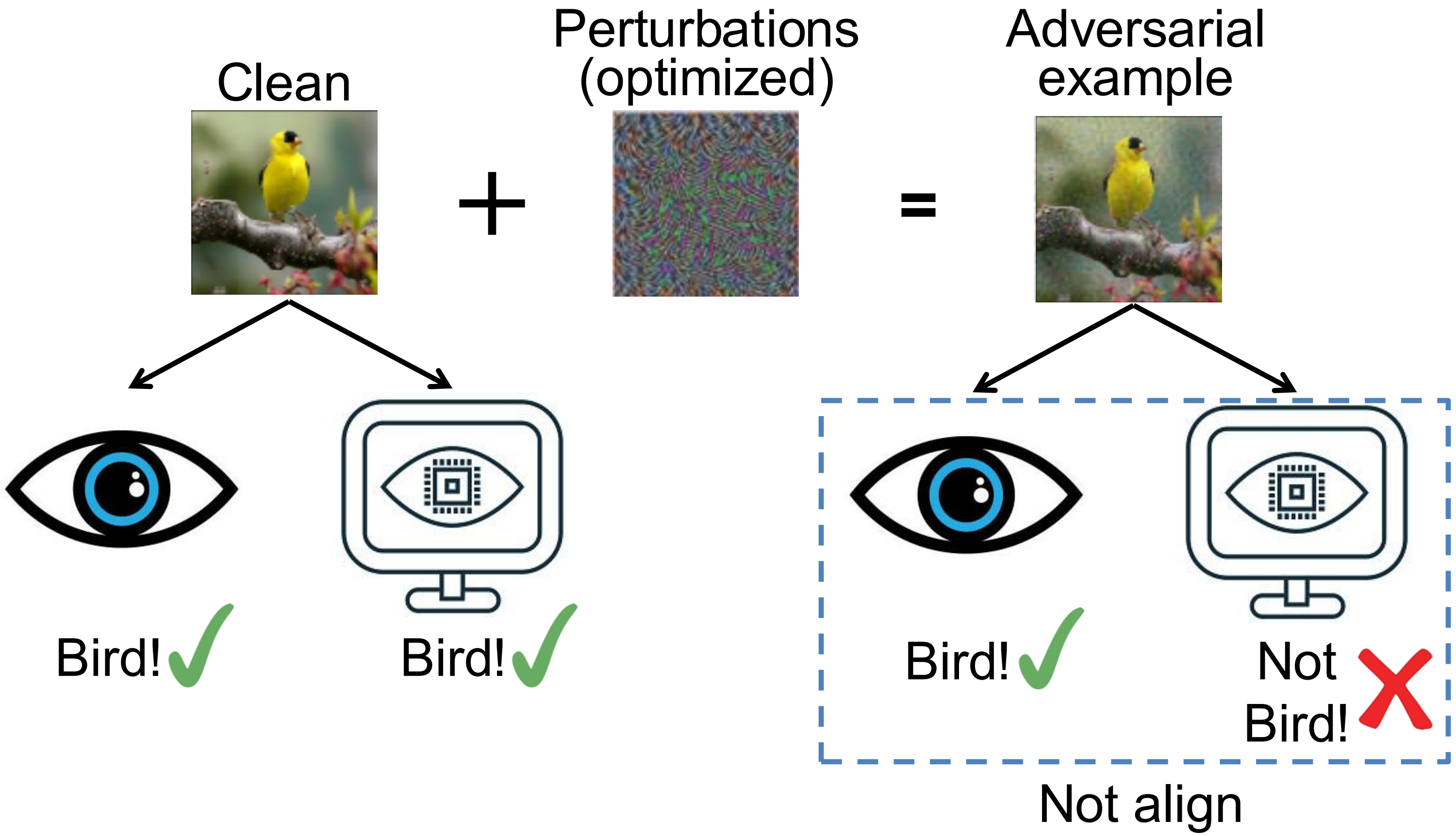
Definition


Clean
Perturbations
(optimized)
+
=
Adversarial
example
Bird!
Bird!

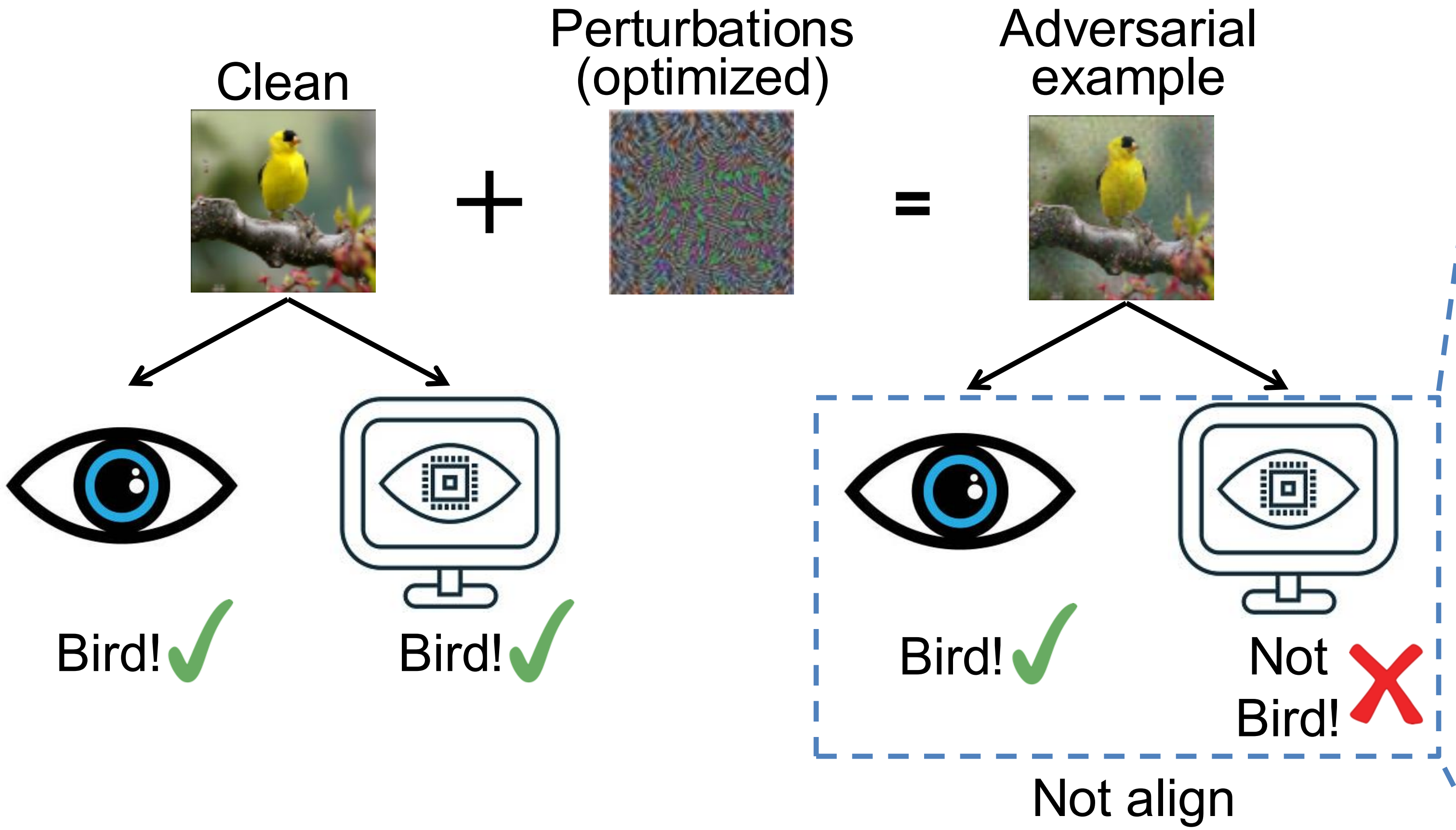
Definition


Clean
Perturbations
(optimized)
Adversarial
example
+
=
Bird!
Bird!
Bird!
Not
Bird!

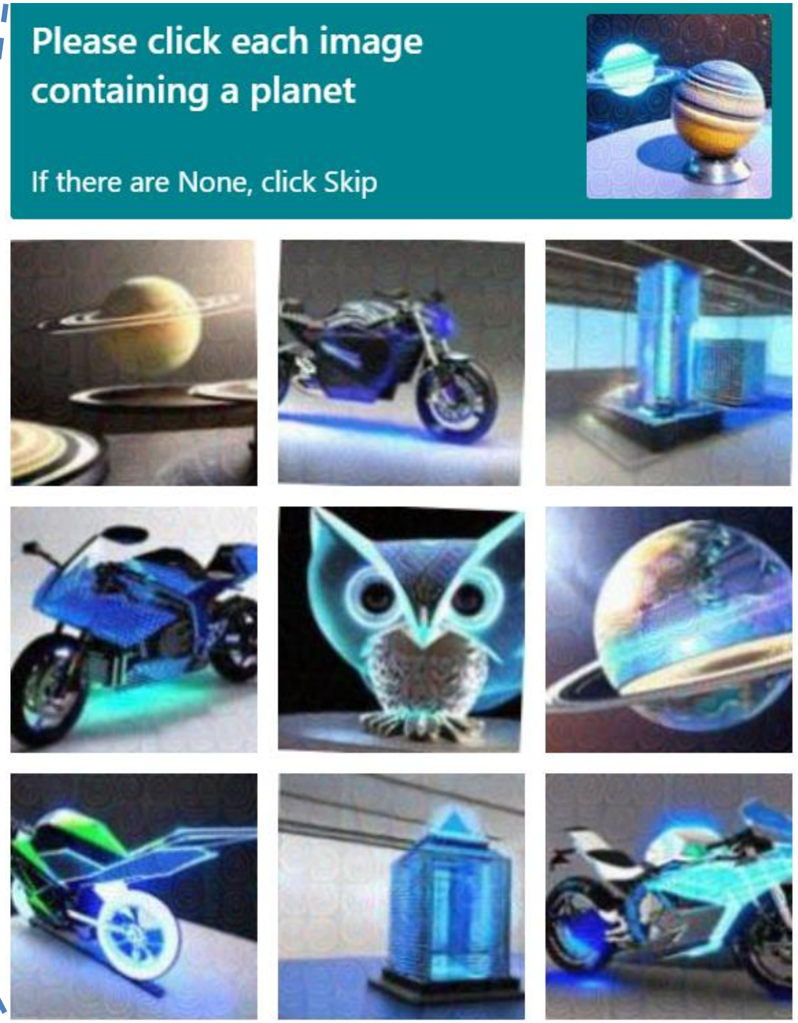
Definition


Clean
+
Perturbations
(optimized)
=
Adversarial
example
Bird!
Bird!
Bird!
Not
Bird!
Not align

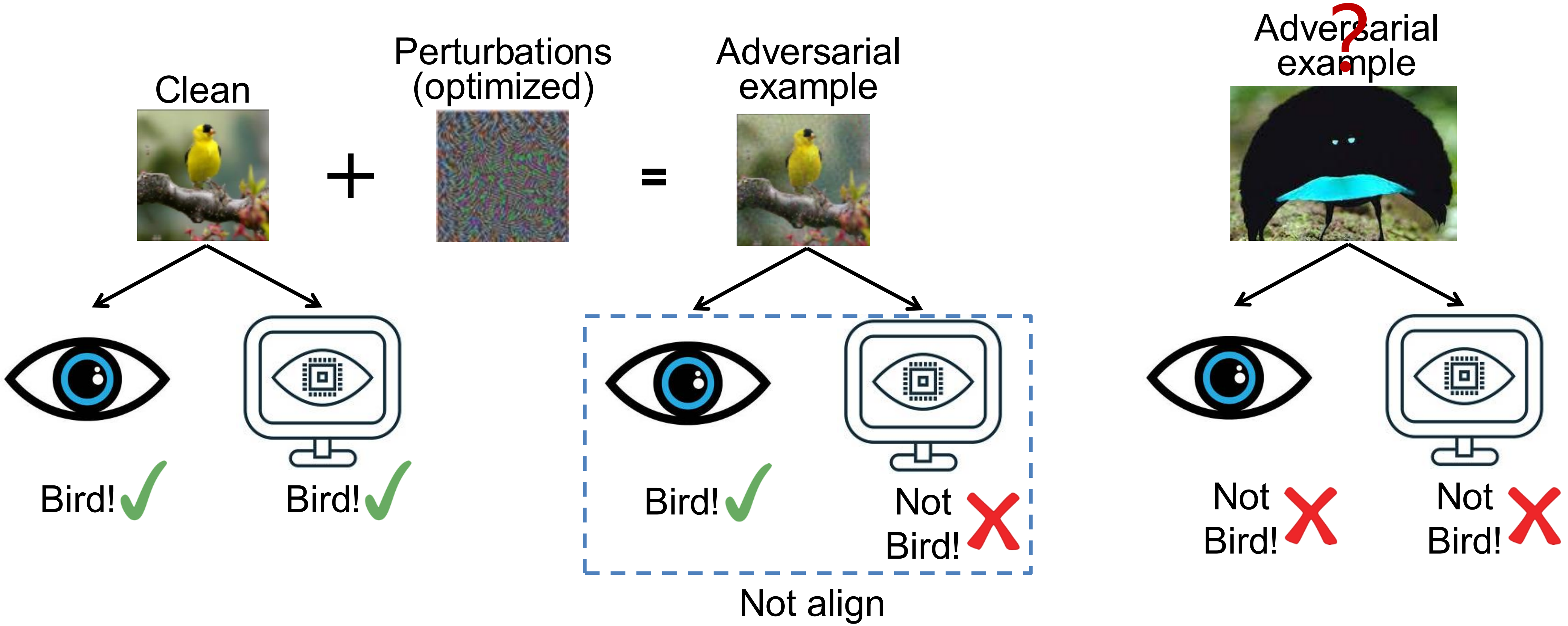
Definition



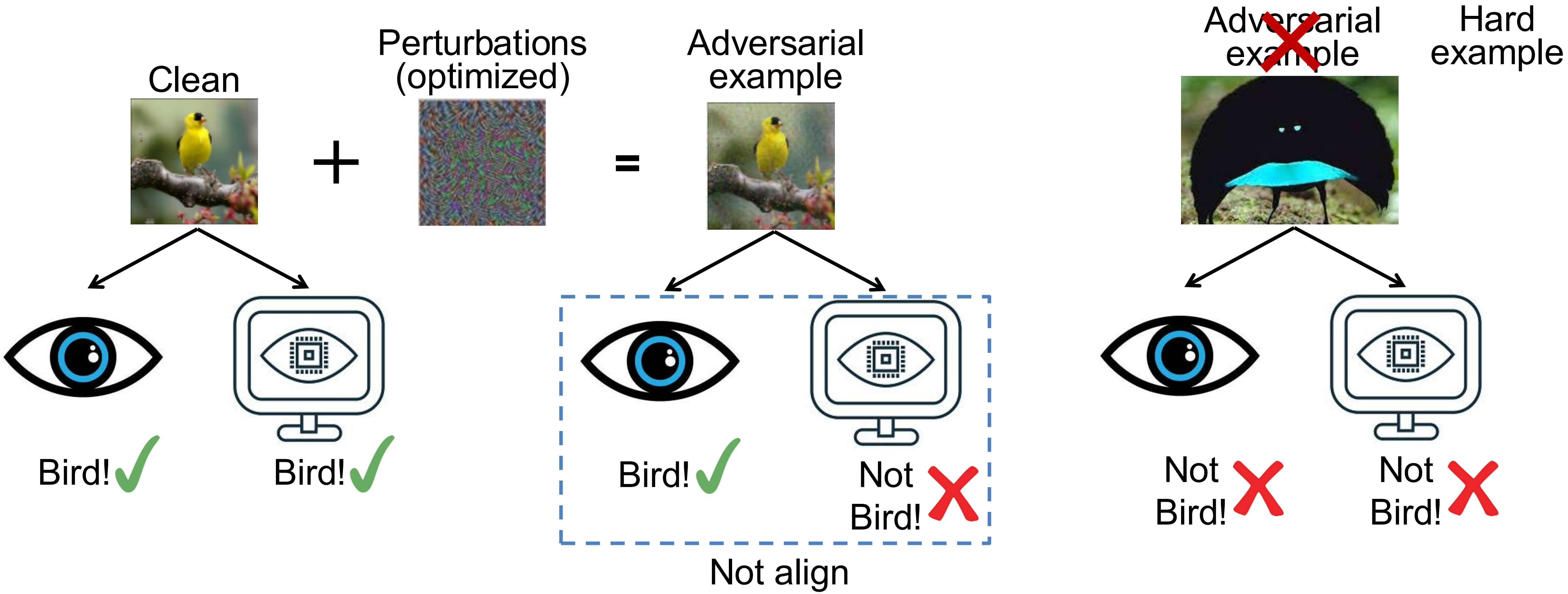
CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart)



Definition


Clean
+
Perturbations
(optimized)
=
Adversarial
example
?
Adversarial
example
Bird!
Bird!
Bird!
Not
Bird!
Not align
Not
Bird!
Not
Bird!

Definition


Clean
+
Perturbations
(optimized)
=
Adversarial
example
~~Adversarial example~~
Hard
example
Bird!
Bird!
Bird!
Not
Bird!
Not align
Not
Bird!
Not
Bird!

Why do Adversarial Examples Exist?

Train





Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

Why do Adversarial Examples Exist?

Train






Test


Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

Why do Adversarial Examples Exist?

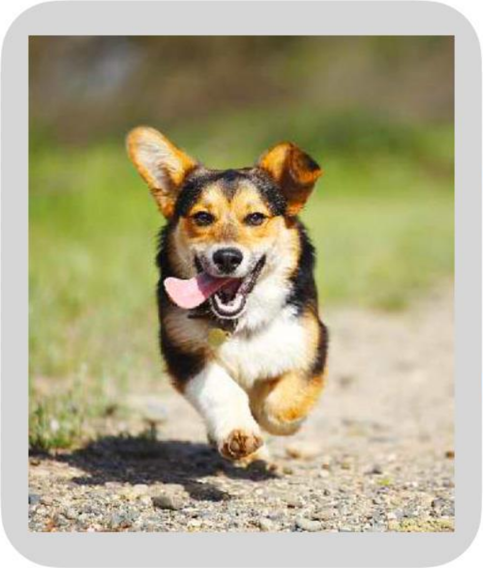
Train

Test

Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

Why do Adversarial Examples Exist?

Training data
“dog”



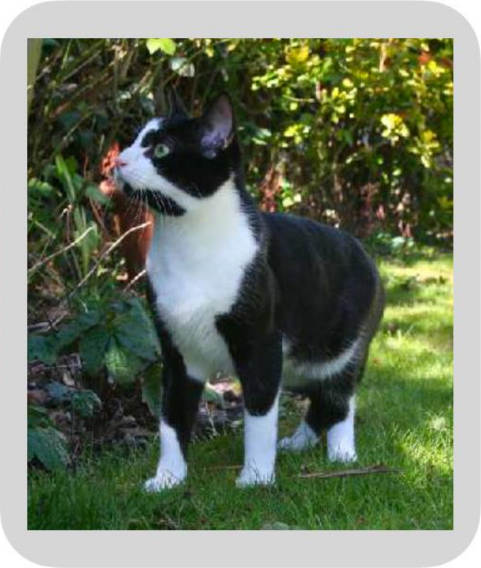
features

- sunny
- dog
- moving

Training data
“cat”

features

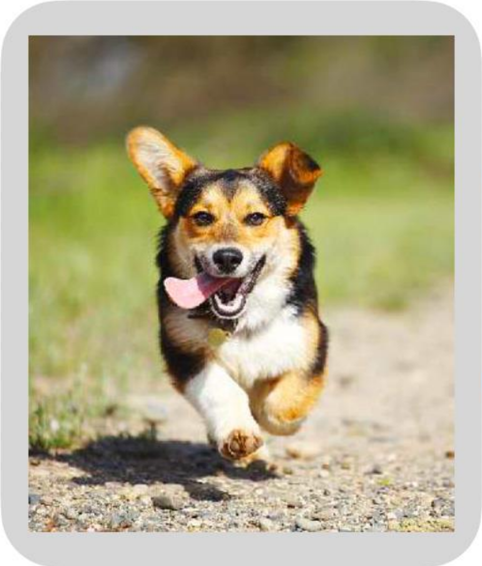
- shady
- cat
- still



Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

Why do Adversarial Examples Exist?

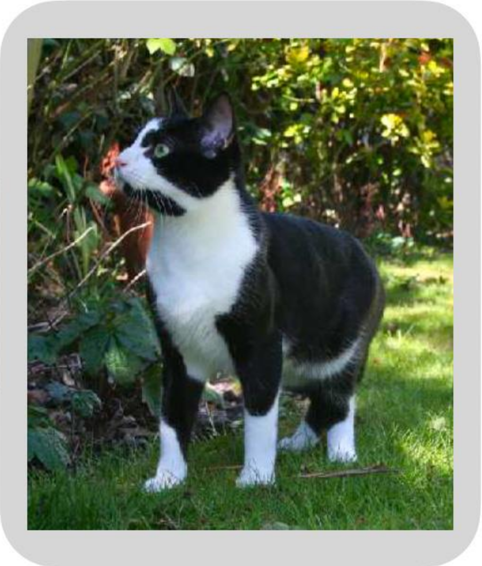
Training data
“dog”



features
- sunny
- dog
- moving

Training data
“cat”

features
- shady
- cat
- still

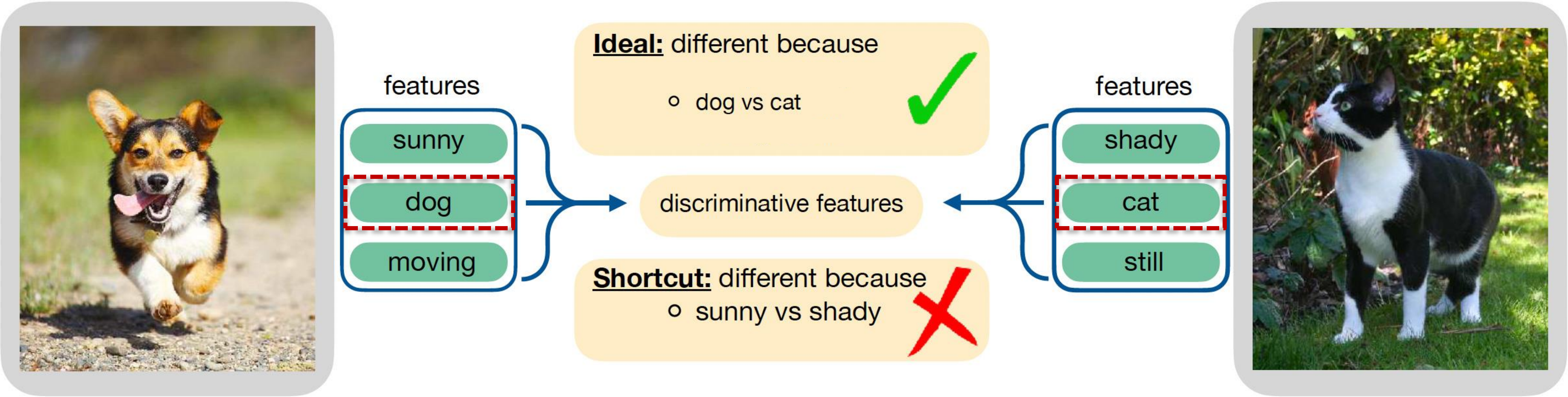


Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

Why do Adversarial Examples Exist?

Training data
“dog”

Training data
“cat”



Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

Why do Adversarial Examples Exist?

Train

Test

?

Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

Why do Adversarial Examples Exist?

Train






Test

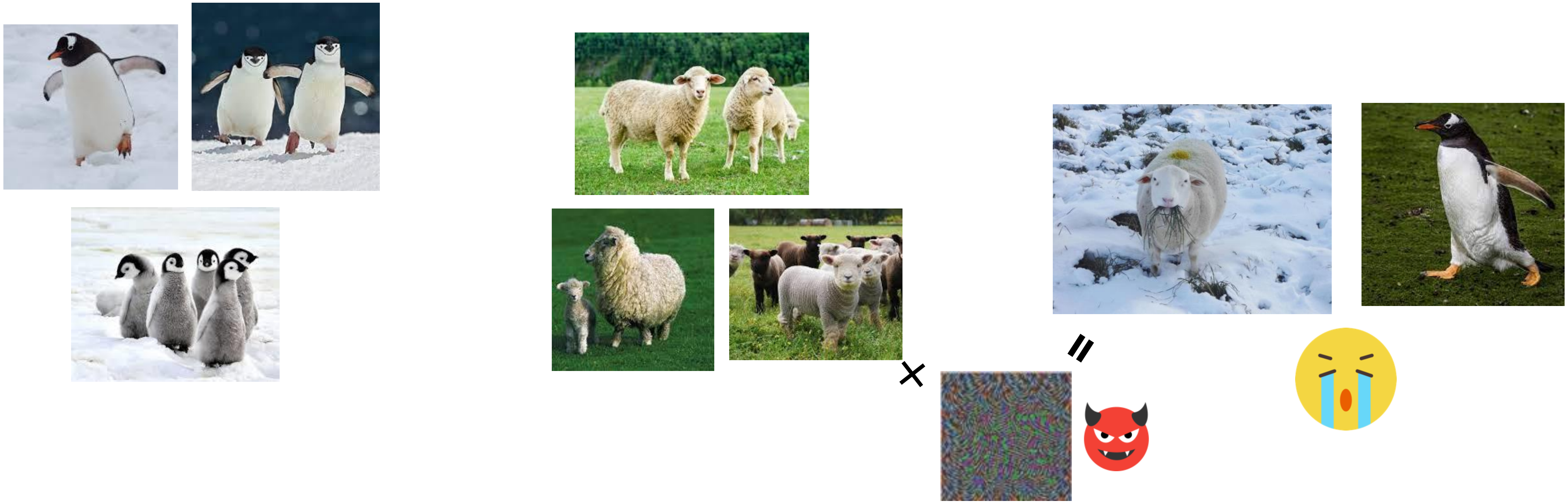


Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

Why do Adversarial Examples Exist?

Train

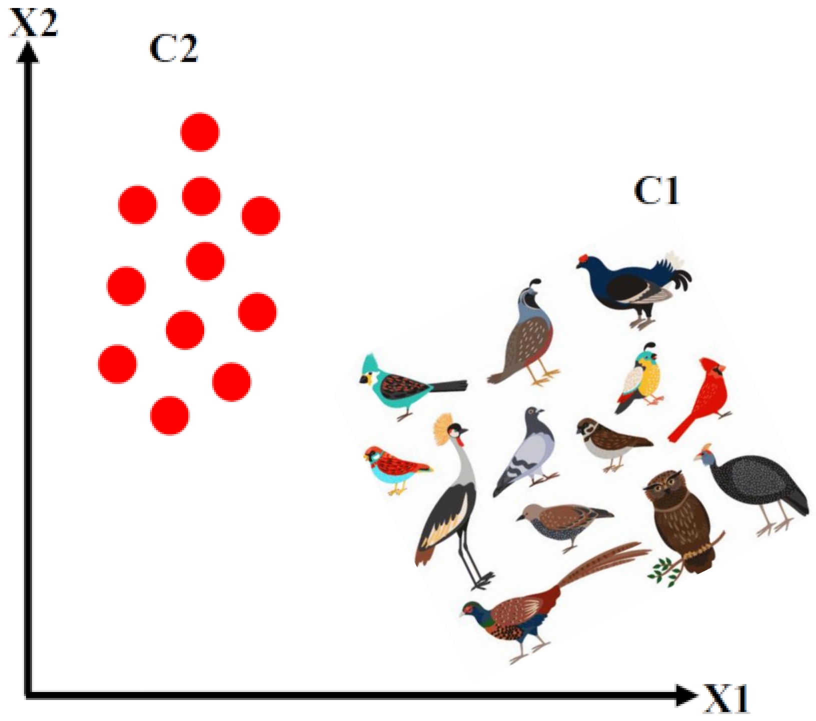
Test



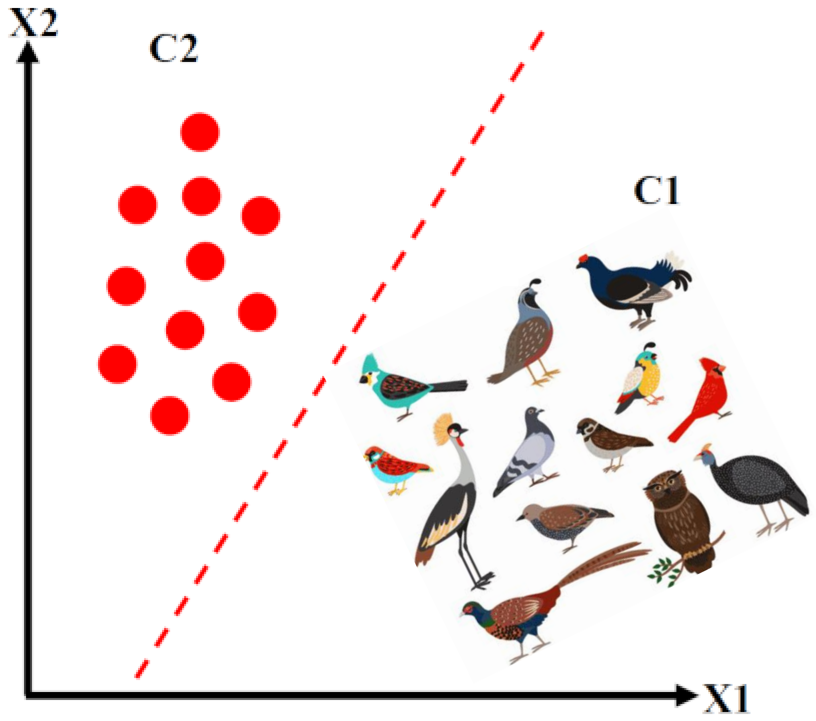
Noise
(optimized)

Shortcut Learning in Deep Neural Networks. Geirhos et al. Nature Machine Intelligence.

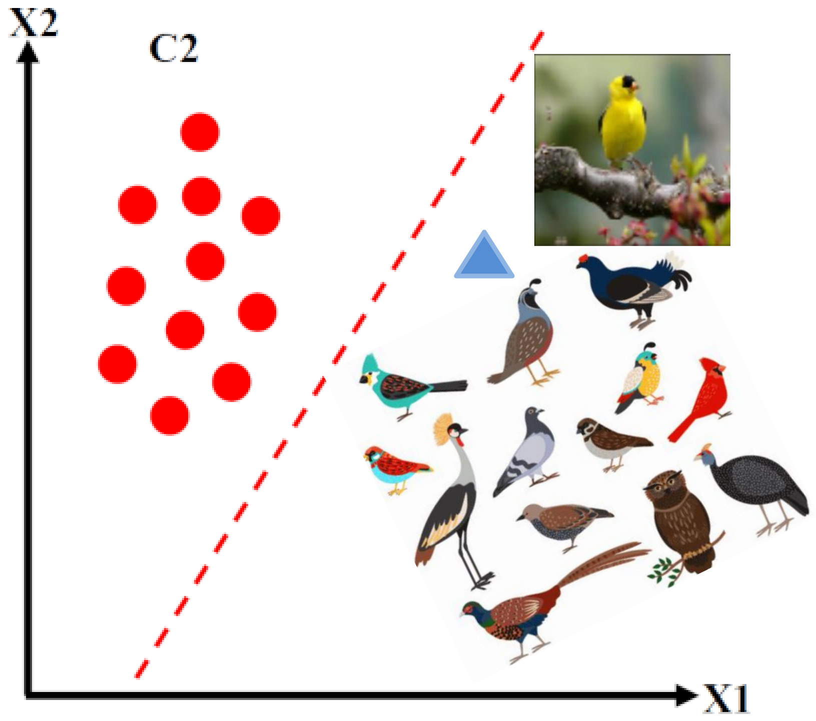
How Can We Generate Adversarial Examples?



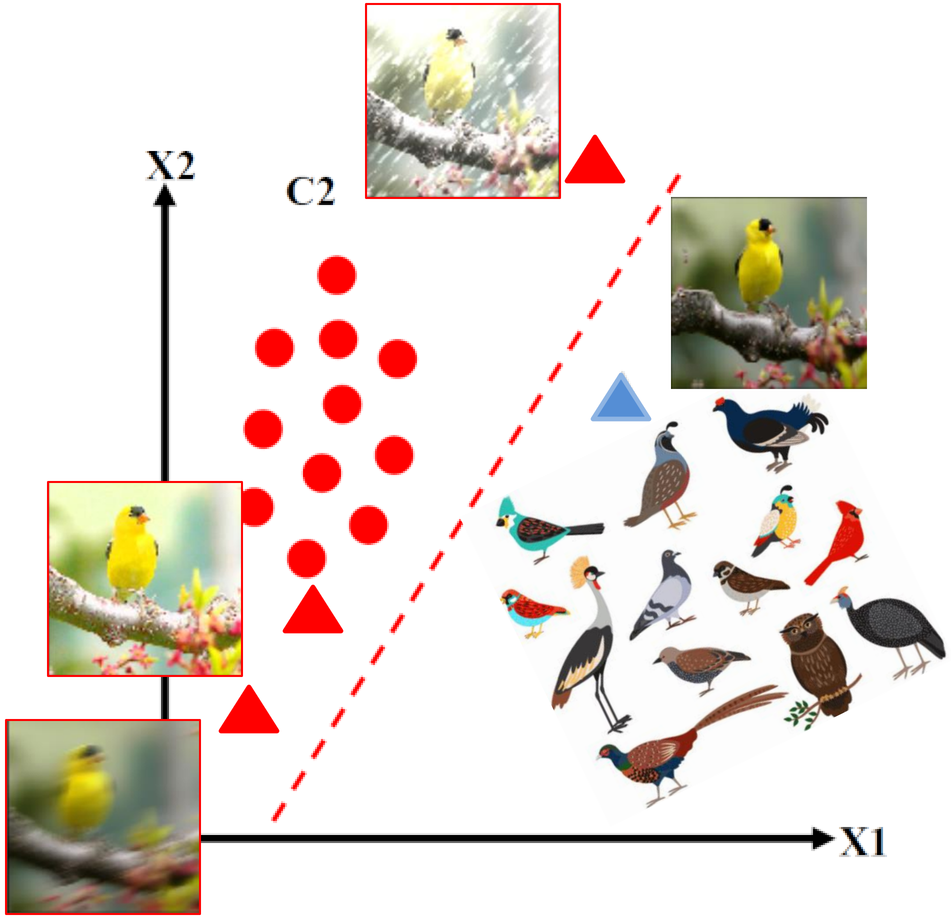
How Can We Generate Adversarial Examples?



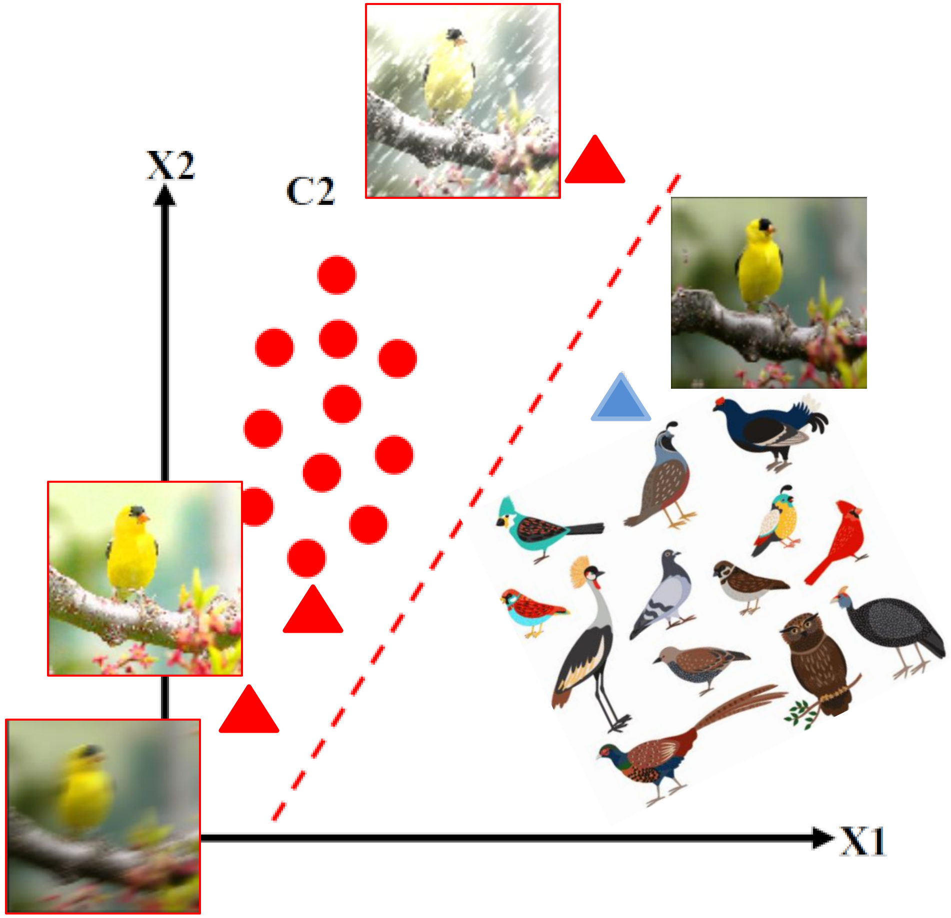
How Can We Generate Adversarial Examples?



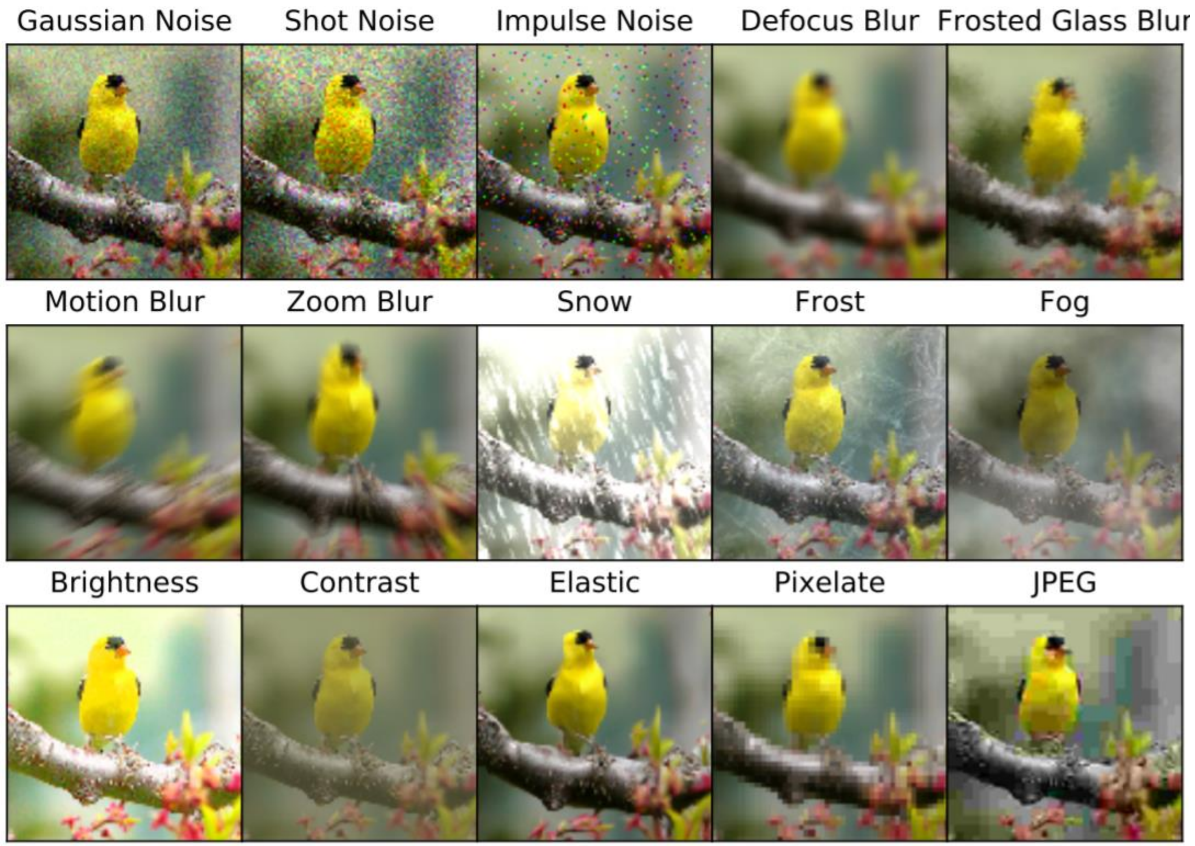
How Can We Generate Adversarial Examples?



How Can We Generate Adversarial Examples?



Noisy Examples → Adversarial Examples

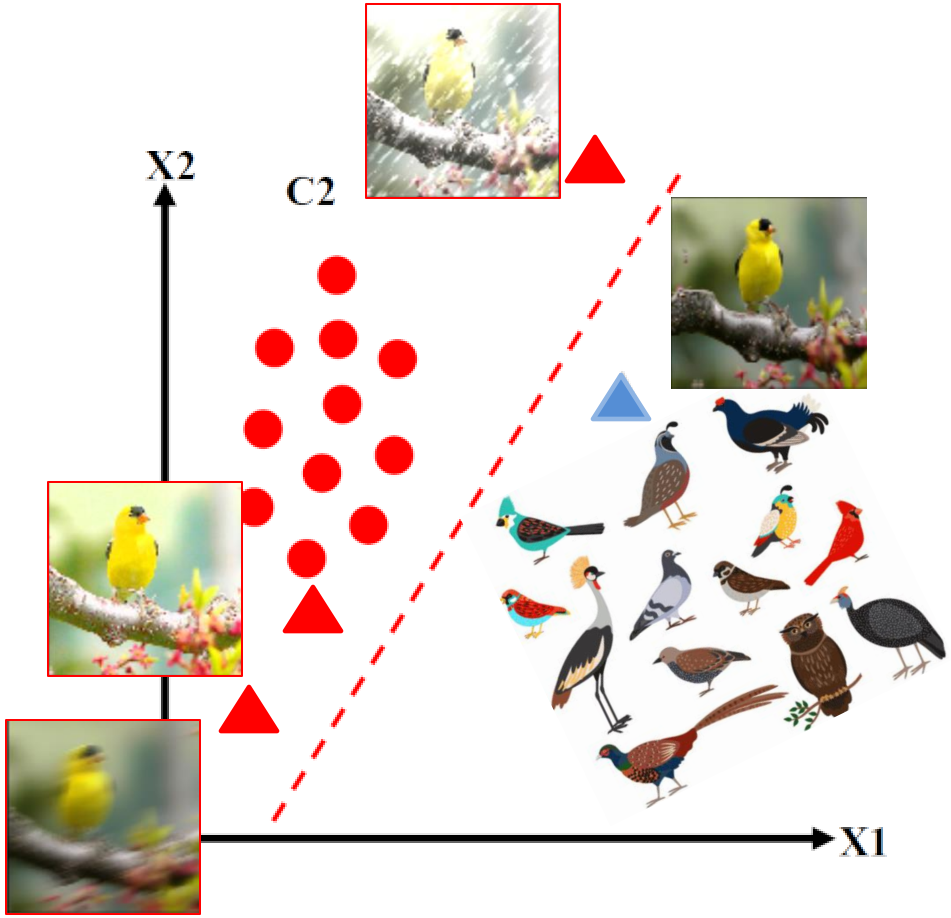


Average-case

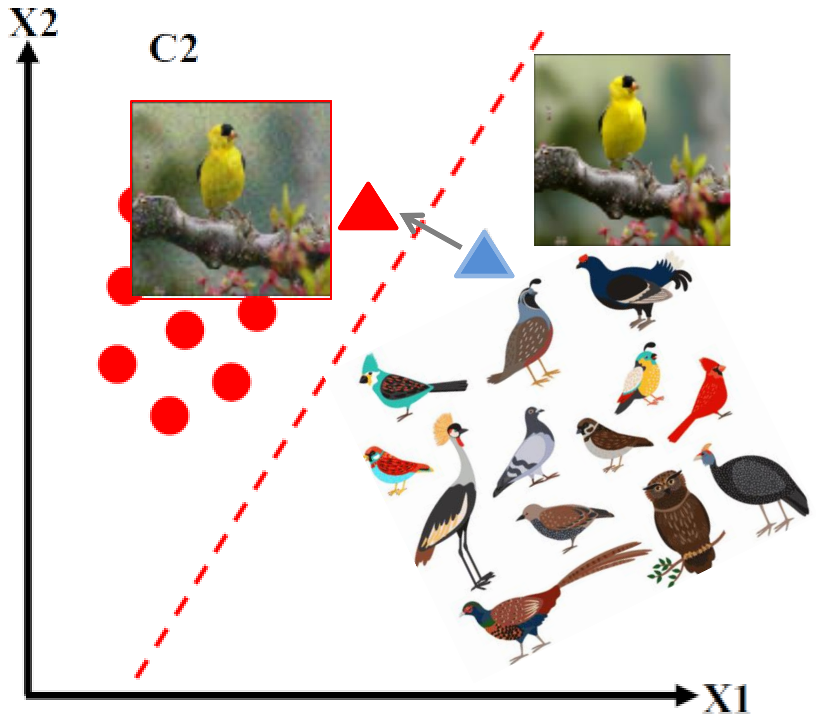
→

Worst-case
(optimized)

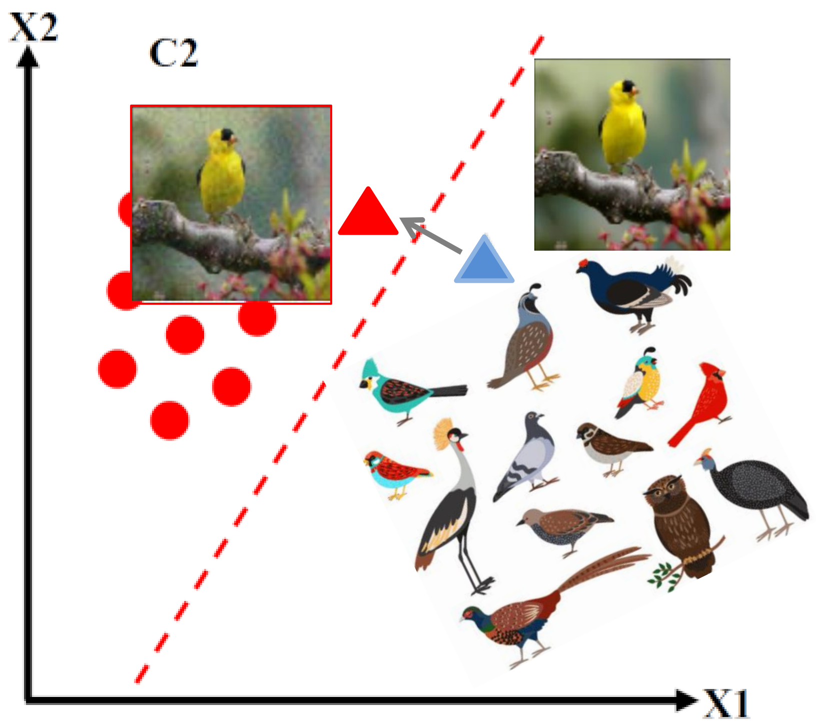
How Can We Generate Adversarial Examples?

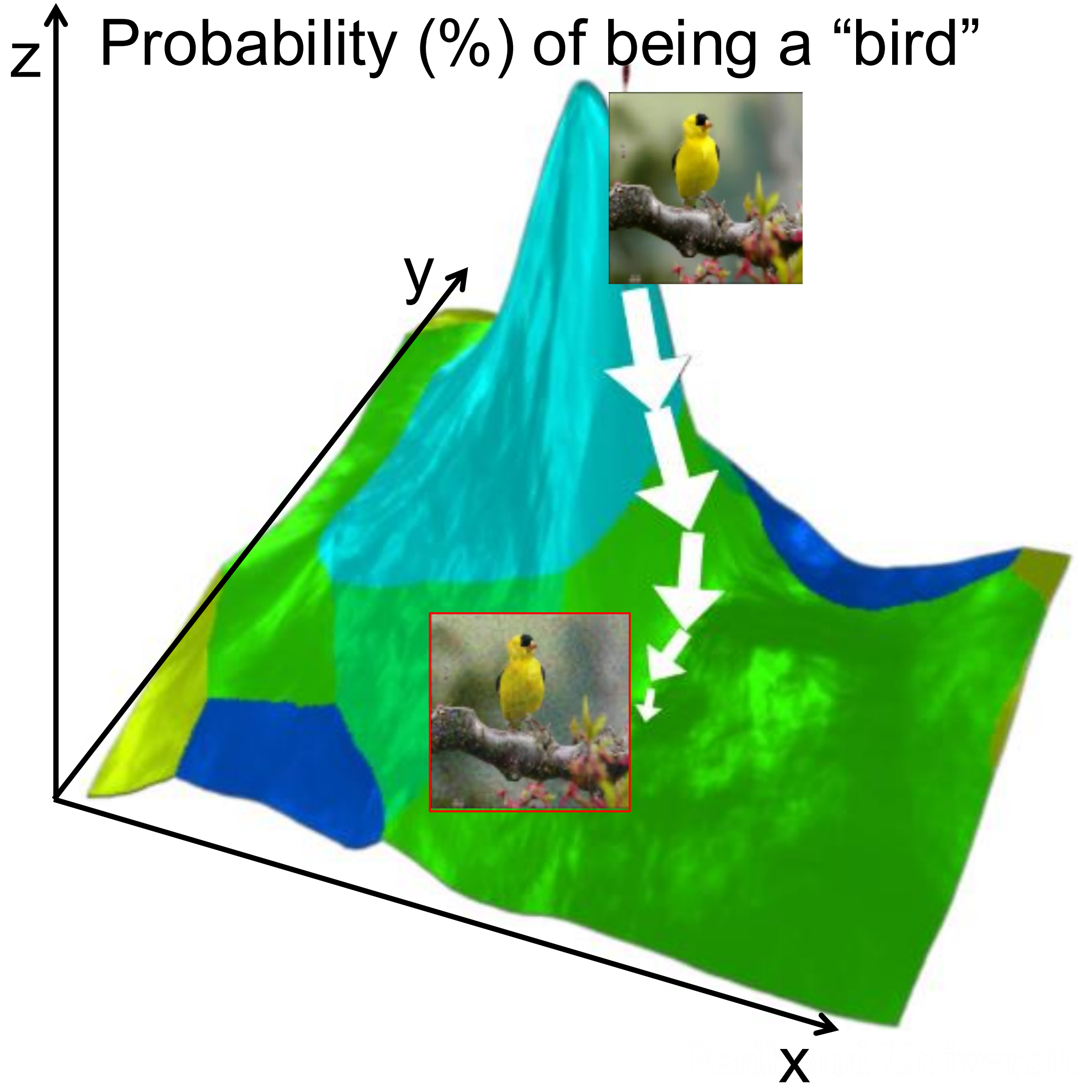


How Can We Generate Adversarial Examples?

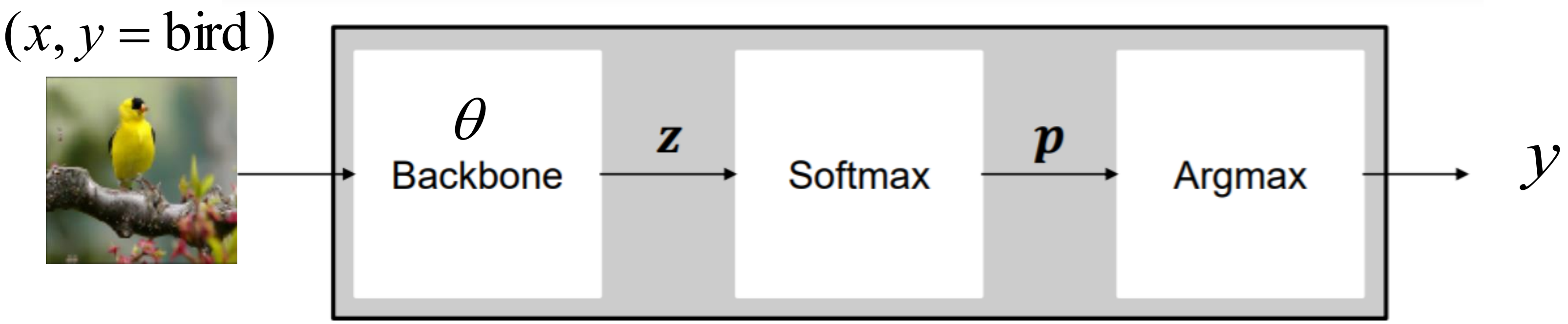


How Can We Generate Adversarial Examples?

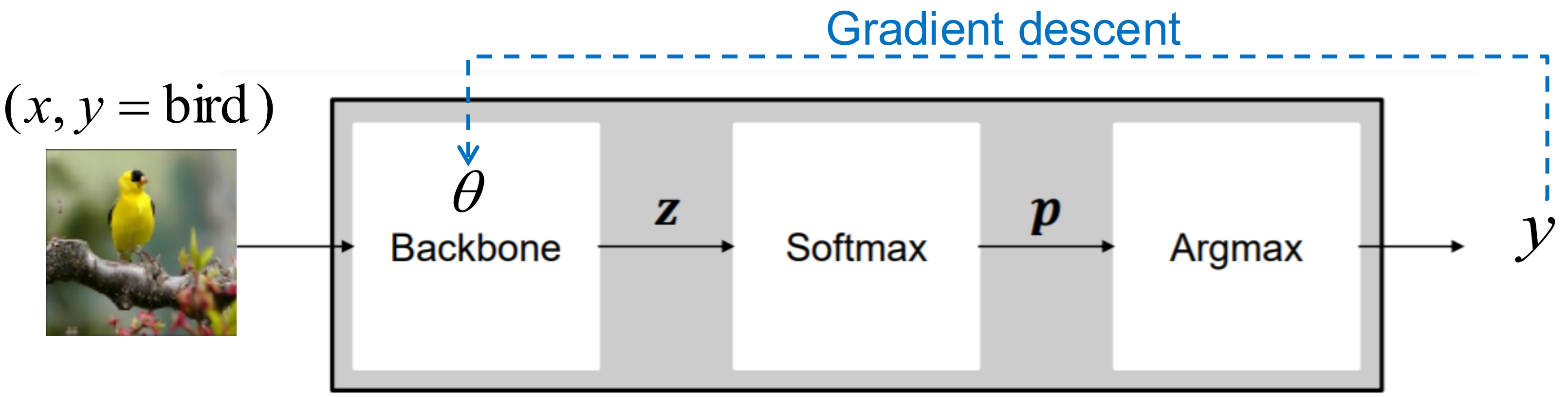




Optimization: Formulation



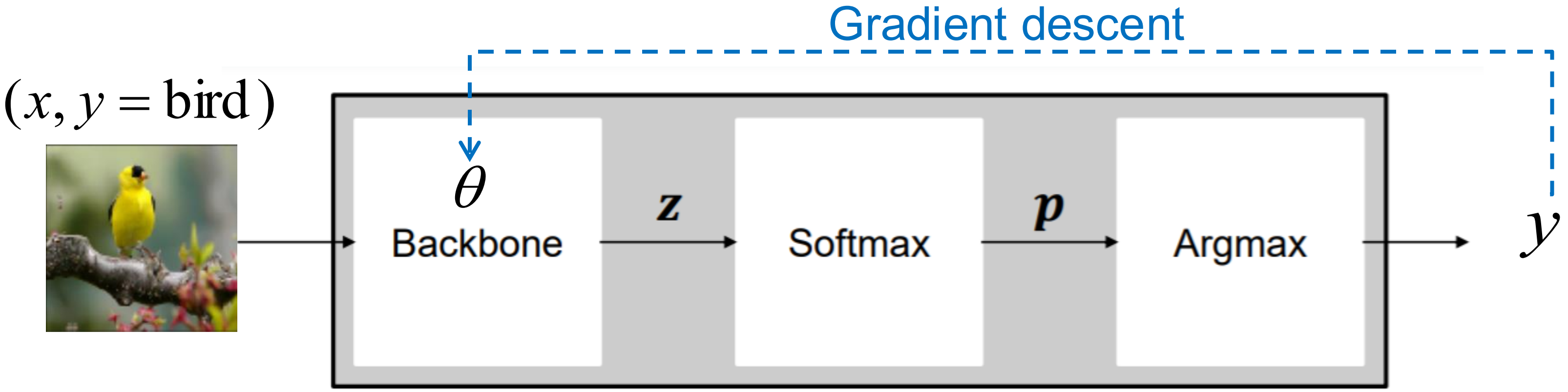
Optimization: Formulation



Model training

$$\theta' = \arg\min_{\theta} \, d(y, y_{\text{bird}})$$

Optimization: Formulation

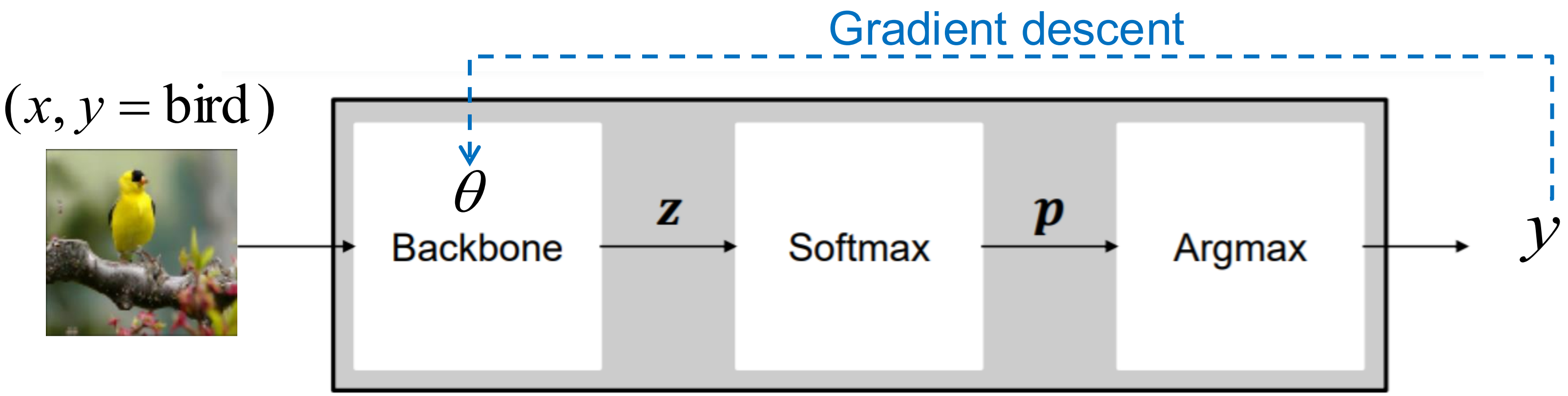


Model training

$$\boxed{\theta' = \arg\min_{\theta} d(y, y_{\text{bird}})}$$

$$d(y, y_{\text{bird}}) = -y_{\text{bird}} \log(p)$$

Optimization: Formulation



Model training

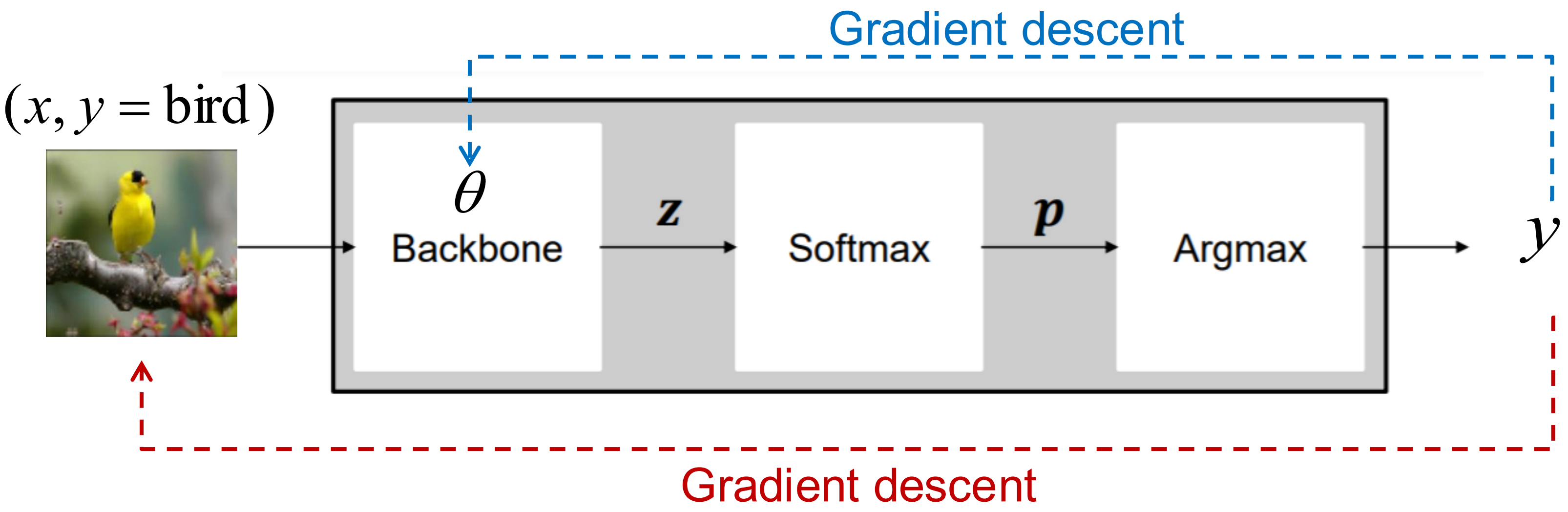
$$\theta' = \arg\min_{\theta} d(y, y_{\text{bird}})$$

Adversarial example

$$x' = \arg\max_{x} d(y, y_{\text{bird}})$$

$$d(y, y_{\text{bird}}) = -y_{\text{bird}} \log(p)$$

Optimization: Formulation



Model training

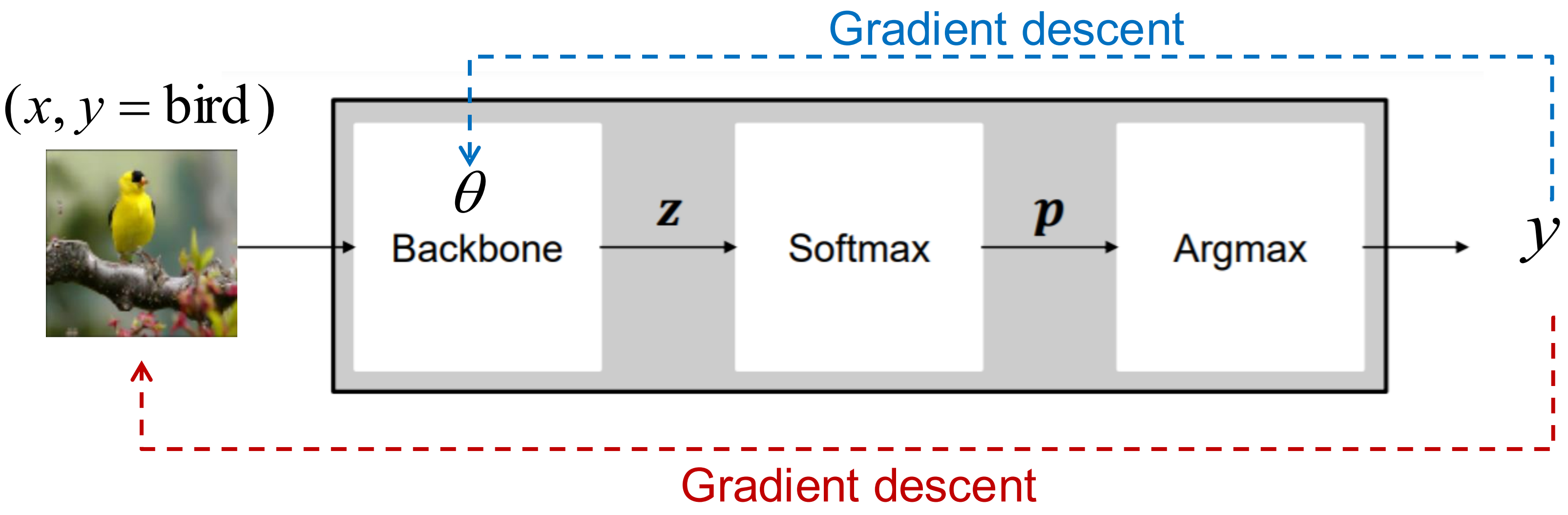
$$\theta' = \arg\min_{\theta} d(y, y_{\text{bird}})$$

Adversarial example

$$x' = \arg\max_{x} d(y, y_{\text{bird}})$$

$$d(y, y_{\text{bird}}) = -y_{\text{bird}} \log(p)$$

Optimization: Formulation



Model training

$$\theta' = \arg\min_{\theta} d(y, y_{\text{bird}})$$

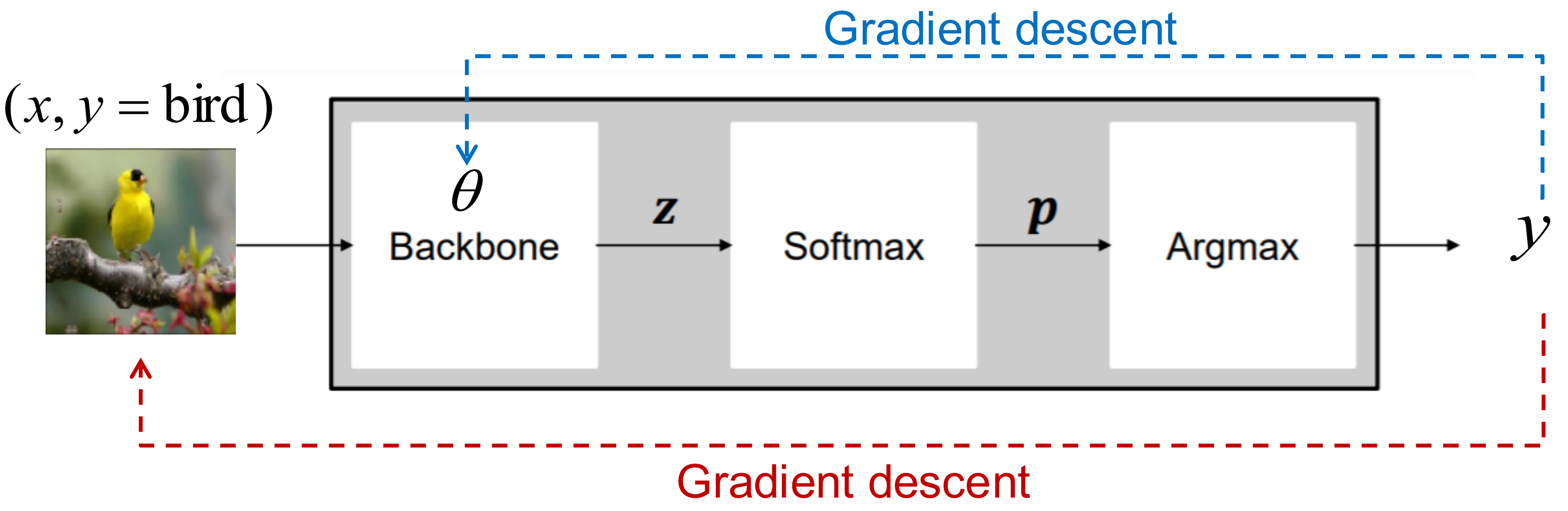
$$d(y, y_{\text{bird}}) = -y_{\text{bird}} \log(p)$$

Adversarial example

$$x' = \arg\max_{x} d(y, y_{\text{bird}})$$

Non-targeted Attack

Optimization: Formulation



Model training

$$\boxed{\theta' = \arg\min_{\theta} d(y, y_{\text{bird}})}$$

$$d(y, y_{\text{bird}}) = -y_{\text{bird}} \log(p)$$

Adversarial example

$$\boxed{x' = \arg\max_{x} d(y, y_{\text{bird}})}$$

Non-targeted Attack

$$x' = \arg\min_{x} d(y, y_{\text{tar}})$$

Targeted Attack

Optimization: Perturbation Constraint ε

Objective: $x' = \arg\max_{x} d(y, y_{\text{bird}})$ s.t. $\Big\|$  $-$  $\Big\|_p \le \varepsilon$

x' x

Optimization: Perturbation Constraint ε

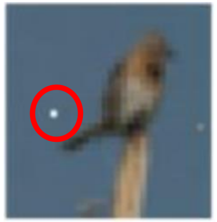
Objective: $x' = \arg\max_{x} d(y, y_{\text{bird}})$ s.t. $\left\| x' - x \right\|_p \le \varepsilon$

x'

x

L0-norm :

$d = \text{num}(\Delta x_n)$; number



Bird (Airplane)

L1-norm :

$d = |\Delta x_1| + |\Delta x_2| + \cdots$; total value

L2-norm :

$d = {\Delta x_1}^2 + {\Delta x_2}^2 + \cdots$; total value

L∞-norm:

$d = \max(\Delta x_1, \Delta x_2, \ldots)$; max value

Optimization: Three Methods

Objective: $x' = \arg\max_{x} d(y, y_{\text{bird}})$ s.t. $\Big\|$  $-$  $\Big\|_p \le \varepsilon$

x' x

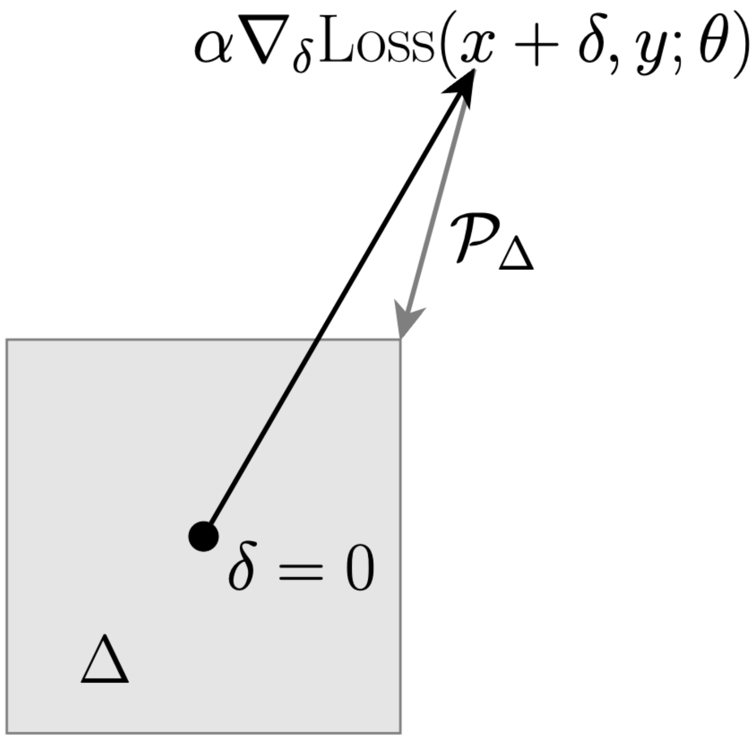
Method 1: Fast Gradient Sign Method (FGSM)

To be more concrete, take Δ to be the ℓ_∞ ball, $\Delta = \{\delta : \|\delta\|_\infty \leq \epsilon\}$, so projection takes the form

$$P_\Delta(\delta) = \mathrm{Clip}(\delta, [-\epsilon, \epsilon])$$

As $\alpha \to \infty$, we always reach "corner" of the box, called fast gradient sign method (FGSM) [Goodfellow et al., 2014]

$$\delta = \epsilon \cdot \mathrm{sign}\big(\nabla_\delta \mathrm{Loss}(x + \delta, y; \theta)\big)$$



Method 2: Projected Gradient Descent (PGD)

Objective: $x' = \arg\max_{x} d(y, y_{\text{bird}})$ s.t. $\|$  $-$  $\|_p \le \varepsilon$

x' x

Method 2: Projected Gradient Descent (PGD)

Objective: $x' = \arg\max_{x} d(y, y_{\text{bird}})$ s.t. $\Big\|$  $-$  $\Big\|_p \leq \varepsilon$

x' x

$$x'_0 = x,$$
$$x'_{i+1} = x'_i - \alpha(\nabla_x d(y, y_{\text{bird}}))$$

Explaining and Harnessing Adversarial Examples. Goodfellow et al. ICLR 2015
Adversarial Examples in the Physical World. Kurakin et al. ICLR workshop 2017

Method 2: Projected Gradient Descent (PGD)

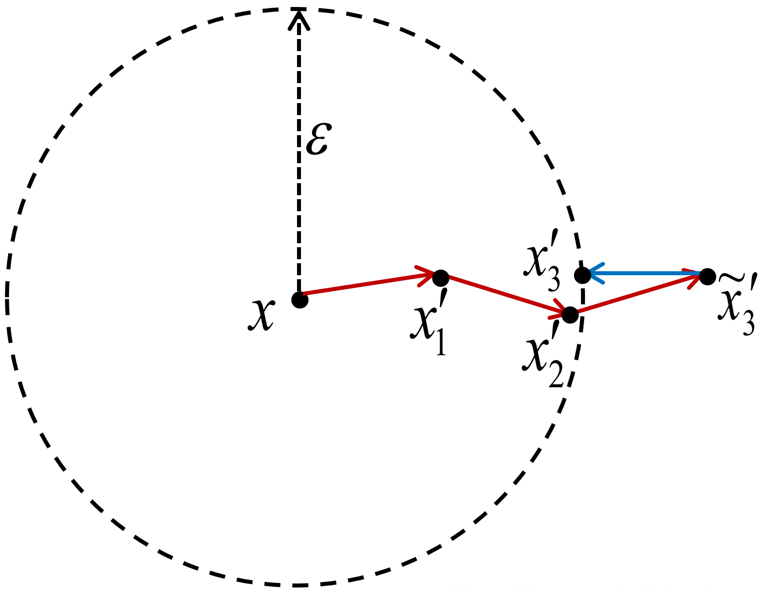
Objective: $x' = \arg\max_{x} d(y, y_{\text{bird}})$ s.t. $\Big\|$  $-$  $\Big\|_p \le \varepsilon$

x' $\quad\quad$ x

$$x'_0 = x,$$

$$x'_{i+1} = x'_i - \alpha(\nabla_x d(y, y_{\text{bird}}))$$

$x' \leftarrow$ project$(x' - x, -\varepsilon, \varepsilon)$



Explaining and Harnessing Adversarial Examples. Goodfellow et al. ICLR 2015
Adversarial Examples in the Physical World. Kurakin et al. ICLR workshop 2017

Method 2: Projected Gradient Descent (PGD)

```
# PGD Attack
# MNIST init
def pgd_attack(model, images, labels, eps=0.3, alpha=2/255, iters=40) :
    images = images.to(device)
    labels = labels.to(device)
    loss = nn.CrossEntropyLoss()

    ori_images = images.data

    for i in range(iters) :
        images.requires_grad = True
        outputs = model(images)

        model.zero_grad()
        cost = loss(outputs, labels).to(device)
        cost.backward()

        adv_images = images + alpha*images.grad.sign()
        eta = torch.clamp(adv_images - ori_images, min=-eps, max=eps)
        images = torch.clamp(ori_images + eta, min=0, max=1).detach_()

    return images
```

https://github.com/Harry24k/PGD-pytorch/blob/master/PGD.ipynb

Method 3: Joint Optimization

Objective: $x' = \arg\max_{x} d(y, y_{\text{bird}})$ s.t. $\Big\|$  $-$  $\Big\|_p \le \varepsilon$

x' x

$$x' = \arg\max_{x} d(y, y_{\text{bird}}) - \lambda \cdot \|x' - x\|_p$$

Intriguing Properties of Neural Networks. Szegedy et al. ICLR 2014
Towards Evaluating the Robustness of Neural Networks. Carlini and Wagner. S&P 2017

Method 3: Joint Optimization

Objective: $x' = \underset{x}{\arg\max}\, d(y, y_{\text{bird}})$ s.t. $\Big\|$  $-$  $\Big\|_p \leq \varepsilon$

x' x

$$x' = \underset{x}{\arg\max}\, d(y, y_{\text{bird}}) - \lambda \cdot \|x' - x\|_p$$

$$\|x' - x\|_{p=2} = \sqrt{\sum_{k=1}^{n} (x'_k - x_k)^2}$$

Intriguing Properties of Neural Networks. Szegedy et al. ICLR 2014
Towards Evaluating the Robustness of Neural Networks. Carlini and Wagner. S&P 2017

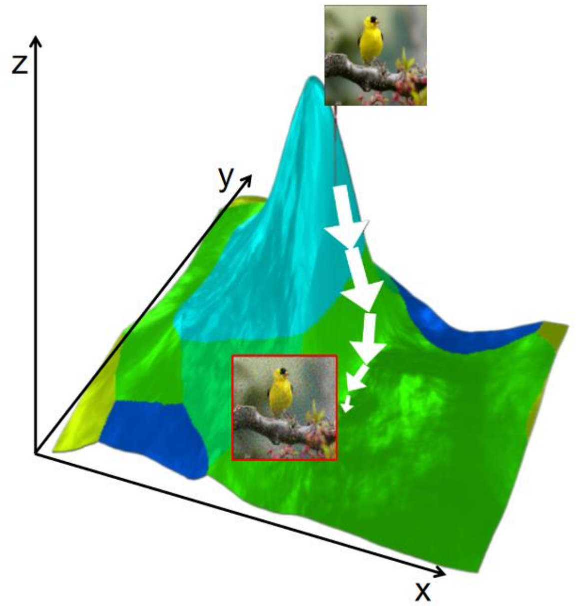
Method 3: Joint Optimization

Objective: $x' = \arg\max_x d(y, y_{\text{bird}})$ s.t. $\Big\|$  $-$  $\Big\|_p \le \varepsilon$

x' x

$$x' = \arg\max_x \underbrace{d(y, y_{\text{bird}}) - \lambda \cdot \|x' - x\|_p}_{L_{\text{joint}}}$$

$$\|x' - x\|_{p=2} = \sqrt{\sum_{k=1}^{n} (x'_k - x_k)^2}$$



$$x'_0 = x,$$

$$x'_{i+1} = x'_i - \alpha \cdot (\nabla_x L_{\text{joint}})$$

Intriguing Properties of Neural Networks. Szegedy et al. ICLR 2014
Towards Evaluating the Robustness of Neural Networks. Carlini and Wagner. S&P 2017

(Empirical) Defenses

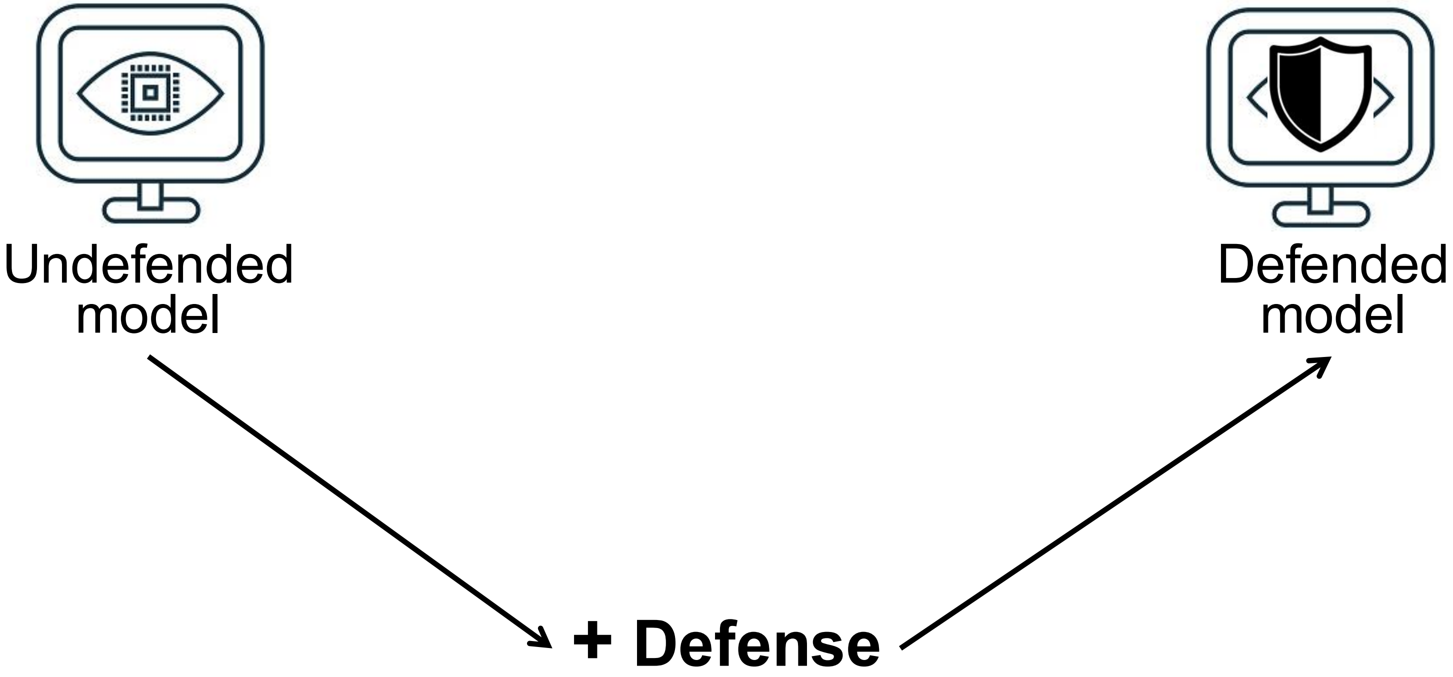
- Black-box:
  Attack doesn’t know defense

- White-box:
  Attack knows and can be adapted to defense

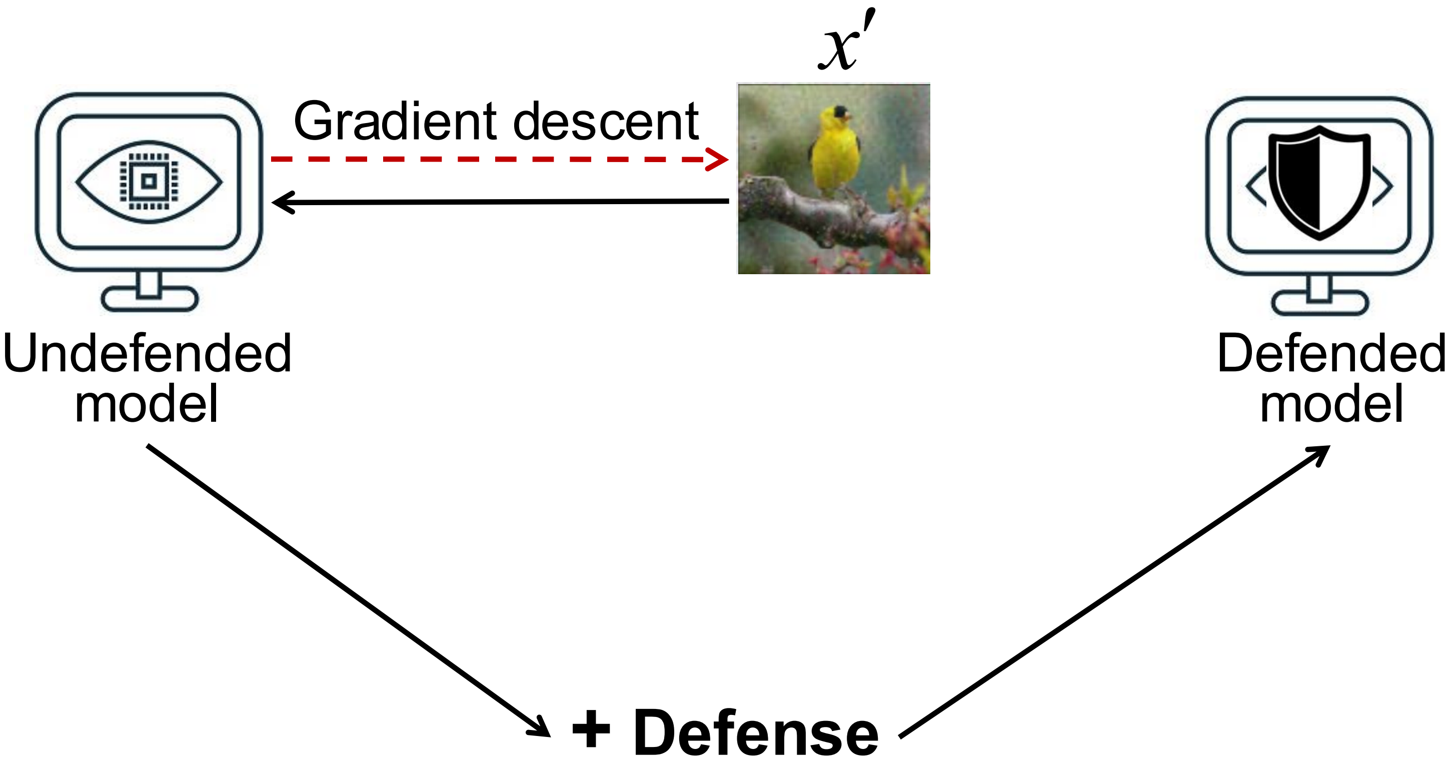
(Empirical) Defenses

- **Black-box:**
  Attack doesn't know defense

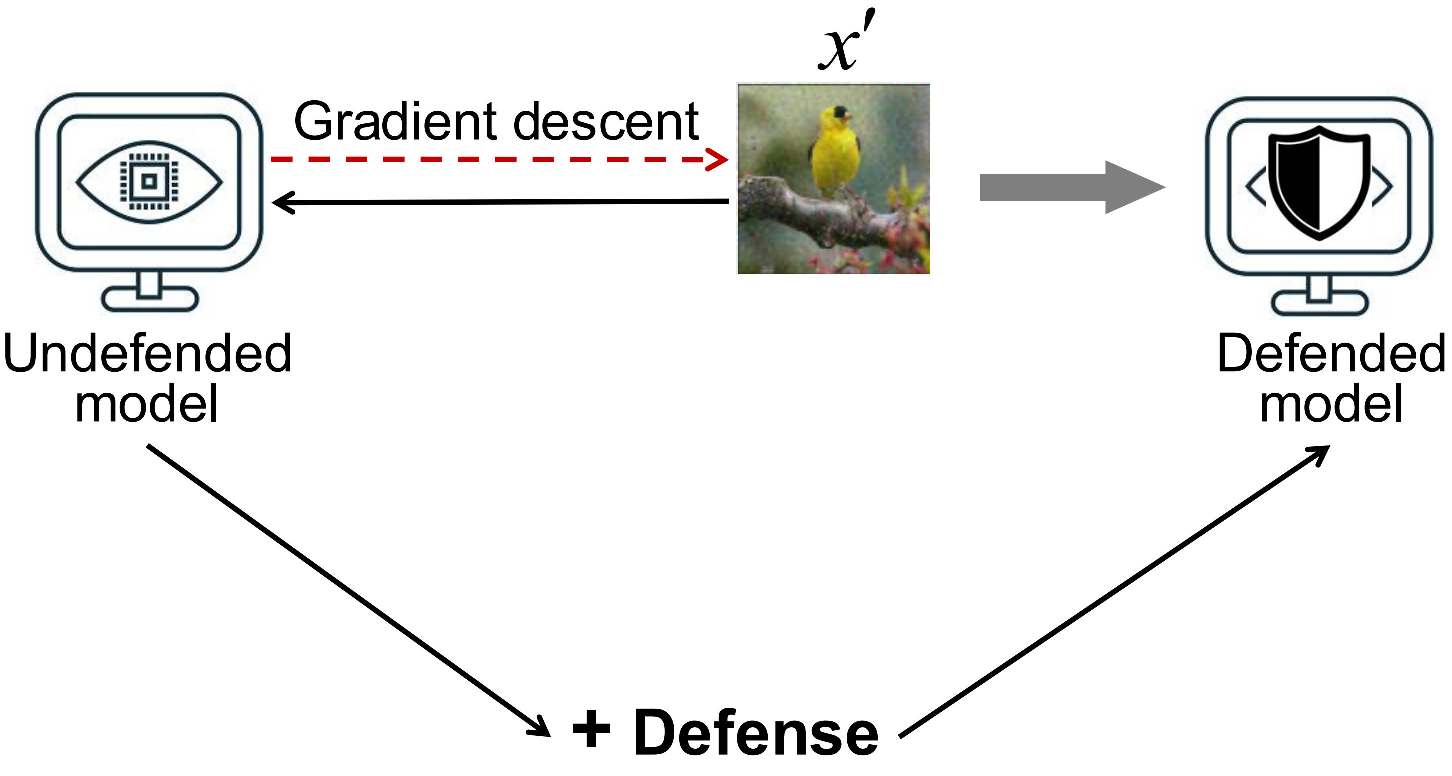
- **White-box:**
  Attack knows and can be adapted to defense


Undefended
model
+ Defense
Defended
model

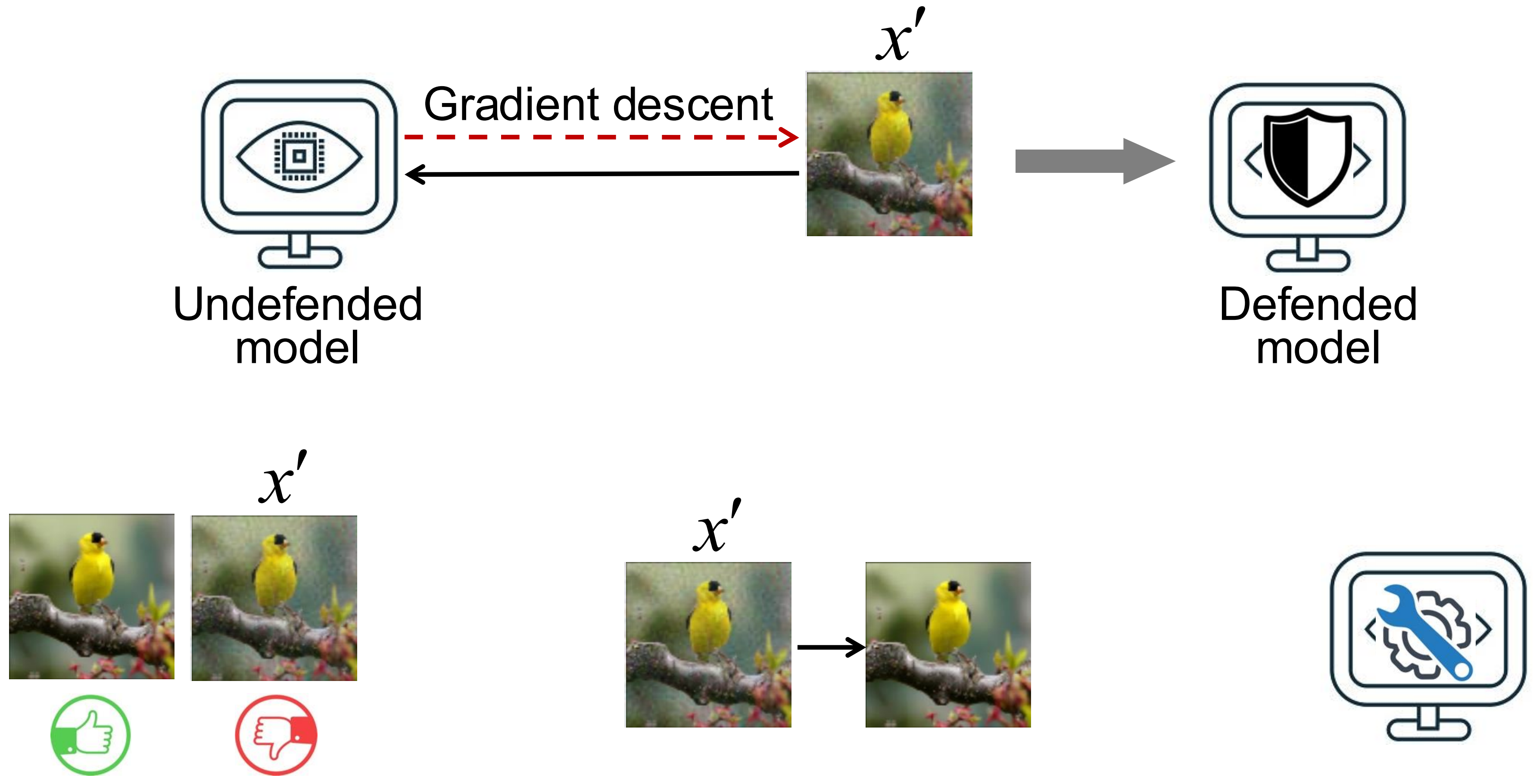
Black-box Defense



Black-box Defense



Black-box Defense

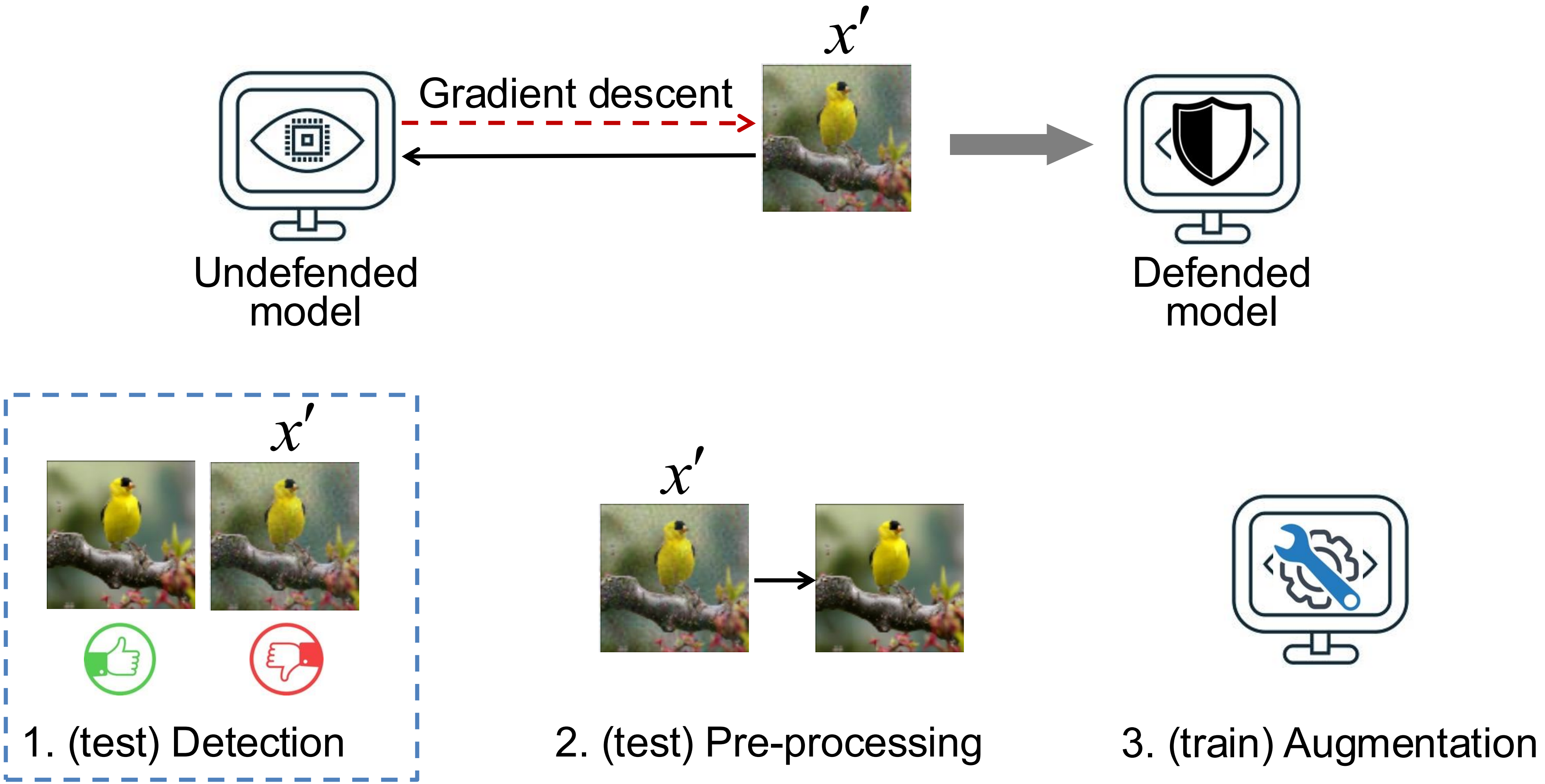


1. (test) Detection

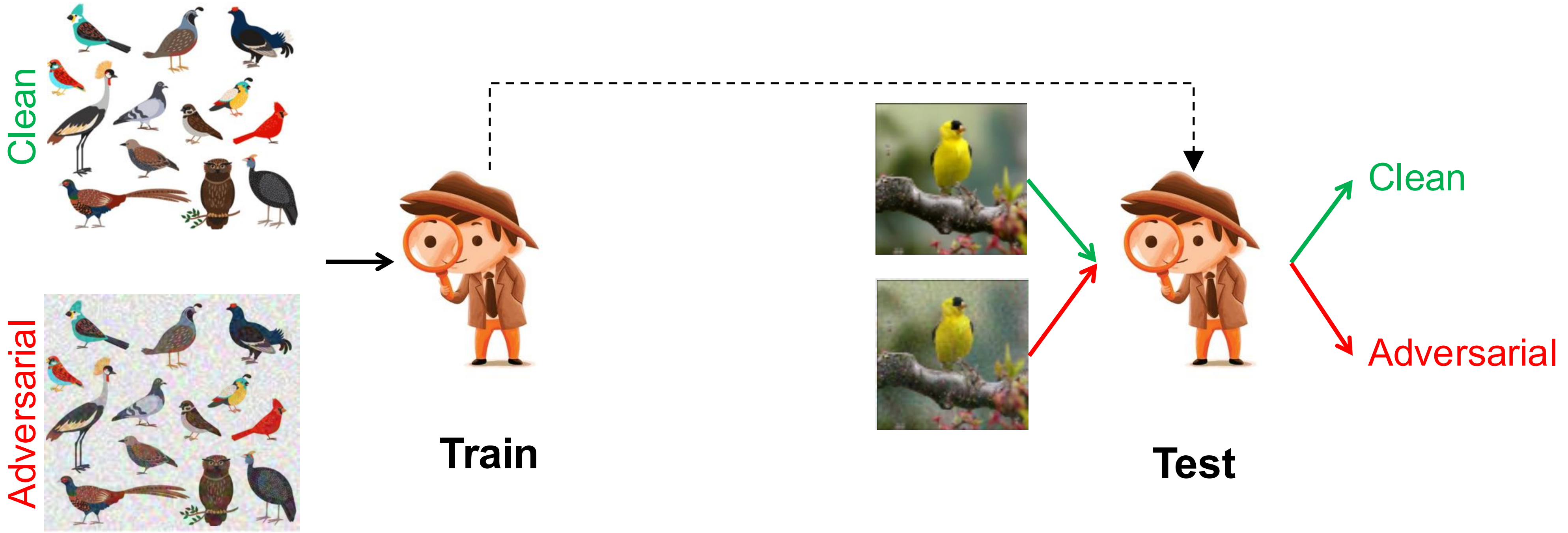
2. (test) Pre-processing

3. (train) Augmentation

Black-box Defense

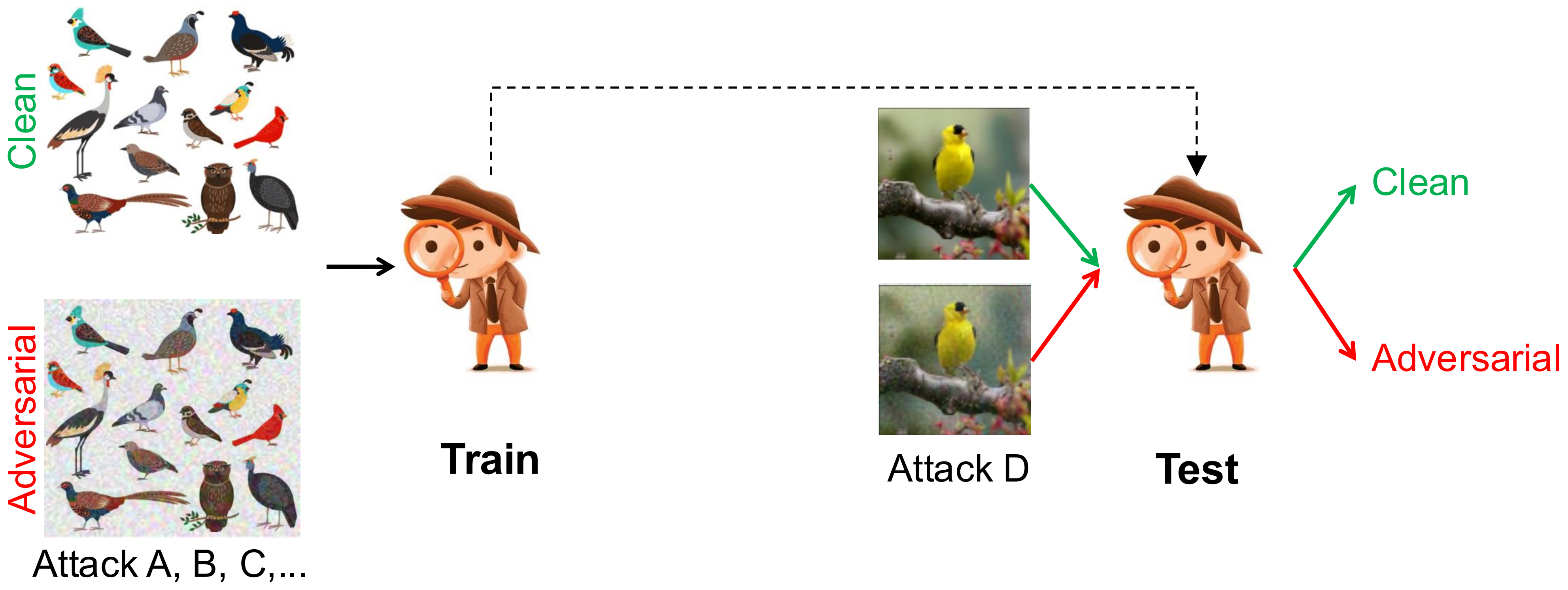


Black-box Defense: Detection



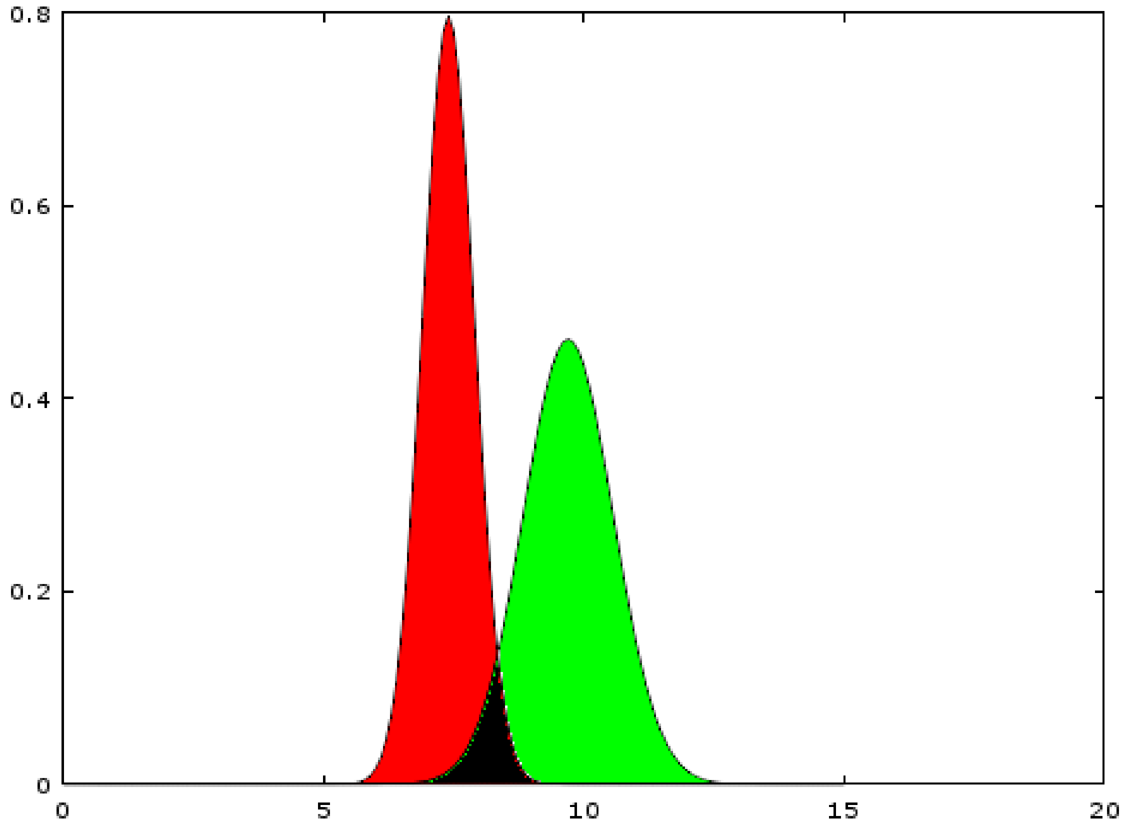
Adversarial and Clean Data Are Not Twins. Gong et al. arXiv 2017

Black-box Defense: Detection

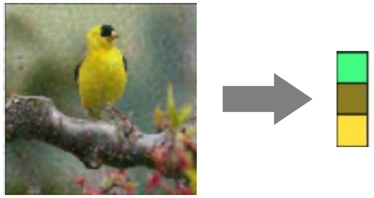


Adversarial and Clean Data Are Not Twins. Gong et al. arXiv 2017

Black-box Defense: Detection: Statistical Features

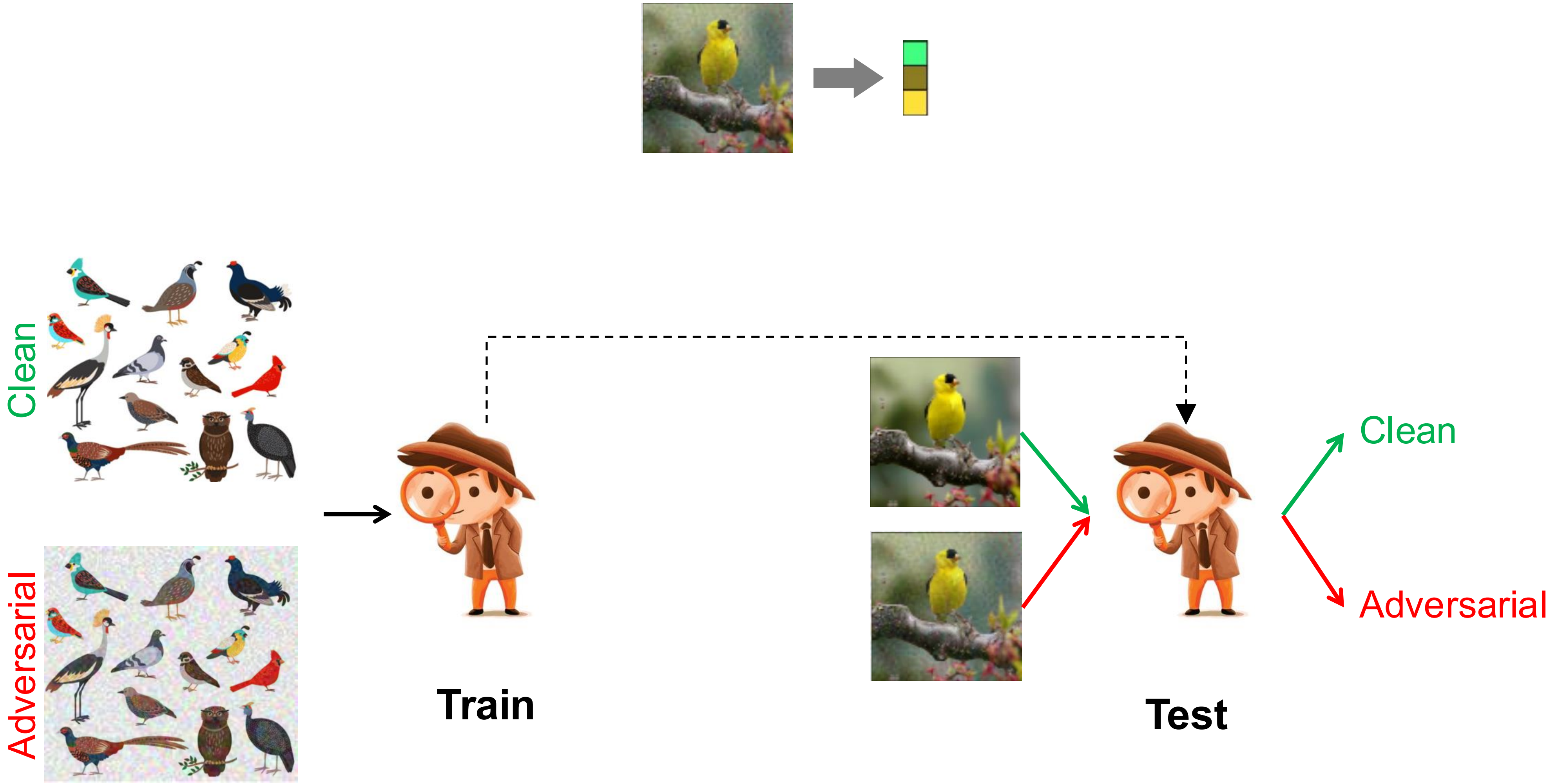


Black-box Defense: Detection: Statistical Features



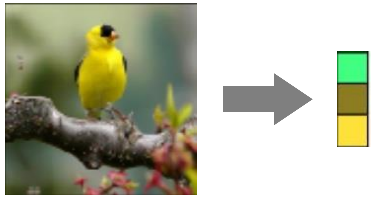
On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

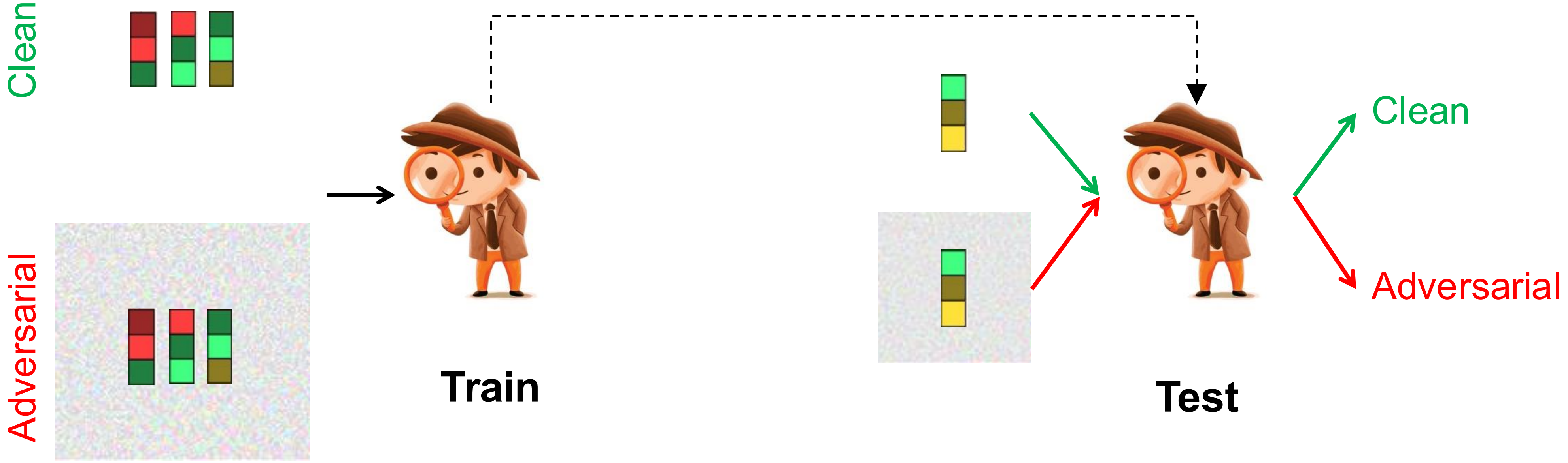
Black-box Defense: Detection: Statistical Features



On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

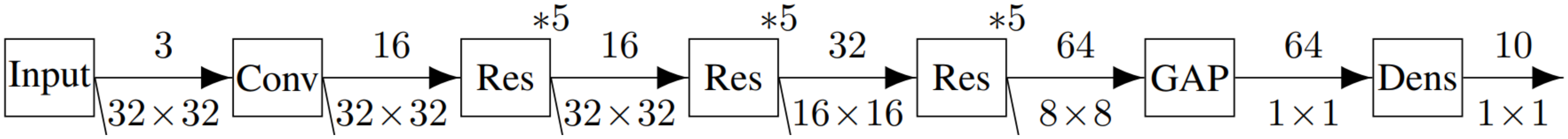
Black-box Defense: Detection: Statistical Features





On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

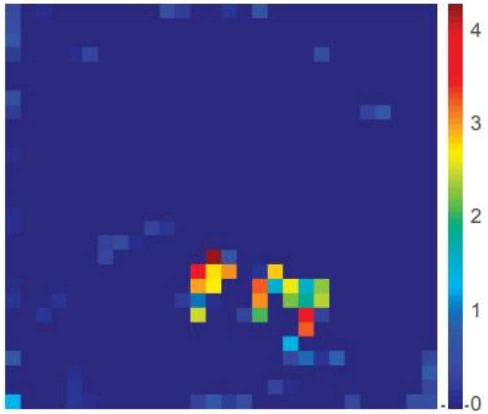
Black-box Defense: Detection: Statistical Features



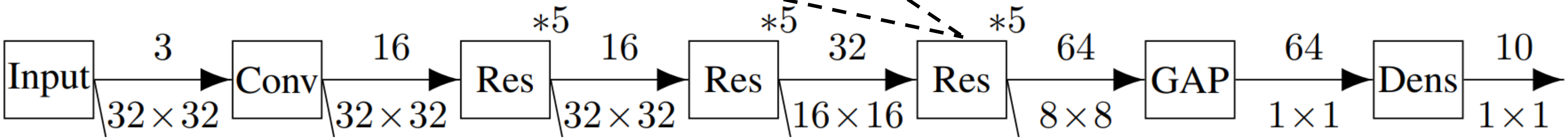
On Detecting Adversarial Perturbations. Metzen et al. ICLR 2017

Black-box Defense: Detection: Statistical Features

Clean



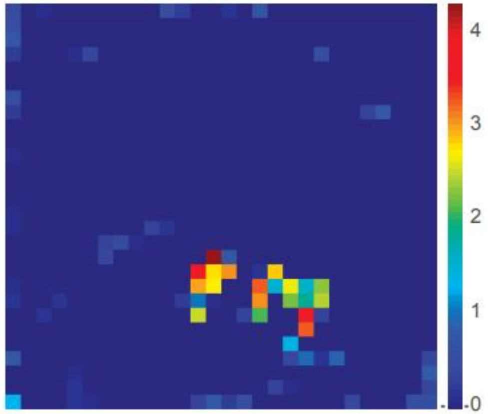
Clean feature map



On Detecting Adversarial Perturbations. Metzen et al. ICLR 2017

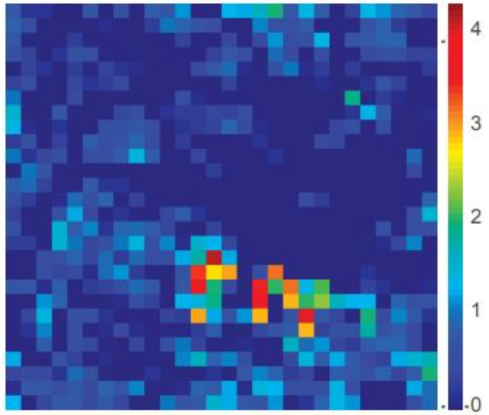
Black-box Defense: Detection: Statistical Features

Clean

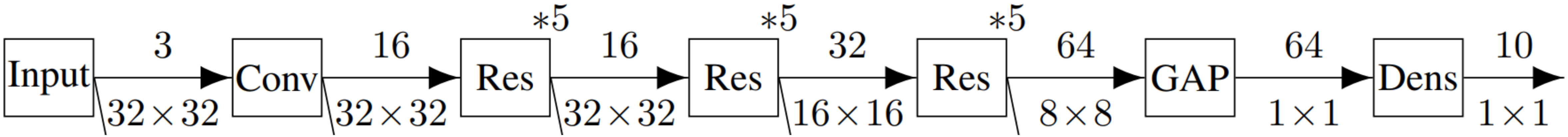


Clean feature map

Adversarial

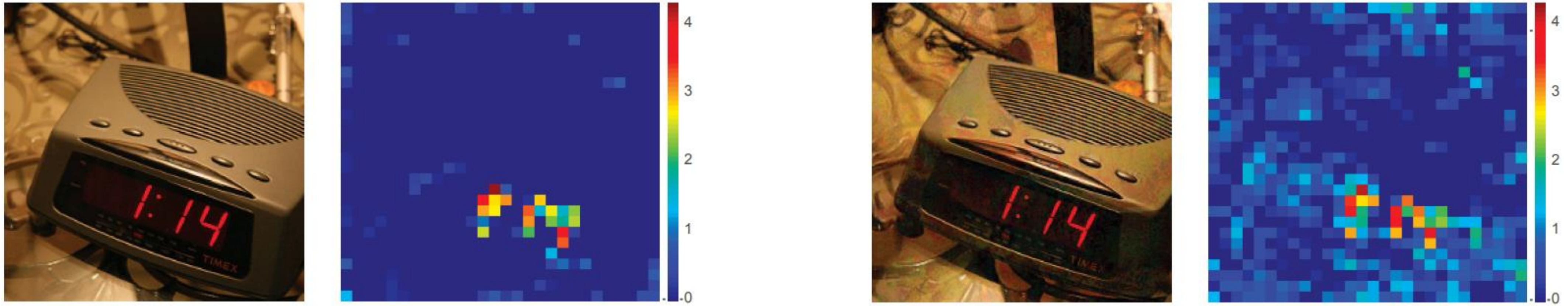


Adversarial feature map

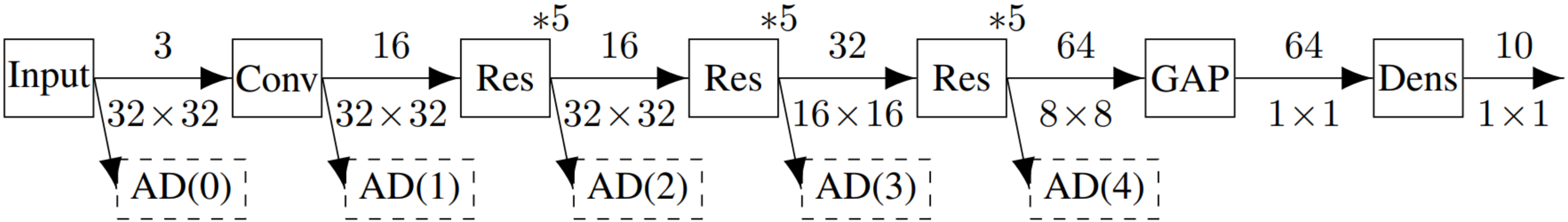


On Detecting Adversarial Perturbations. Metzen et al. ICLR 2017

Black-box Defense: Detection: Statistical Features

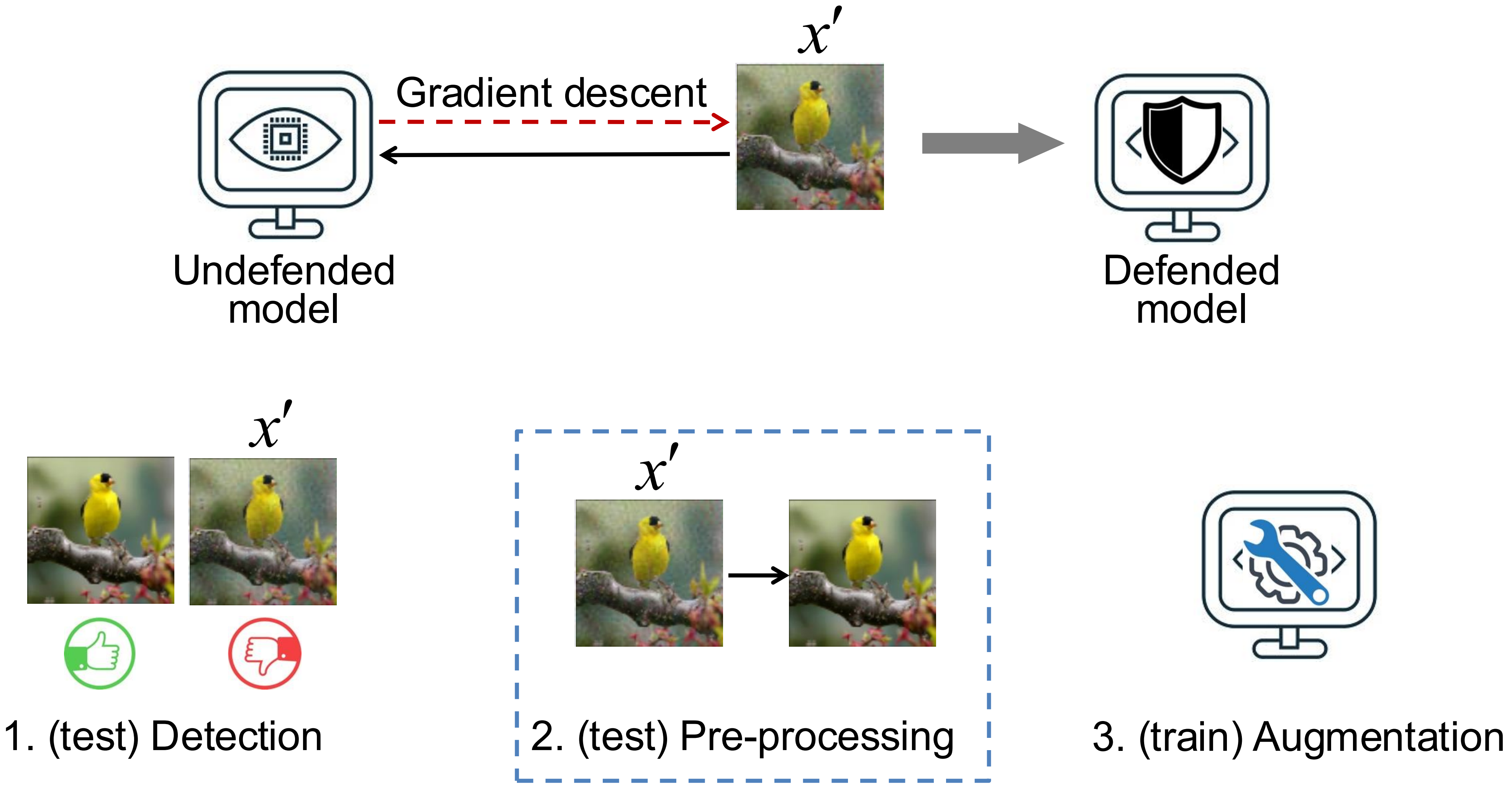


Clean | Clean feature map | Adversarial | Adversarial feature map

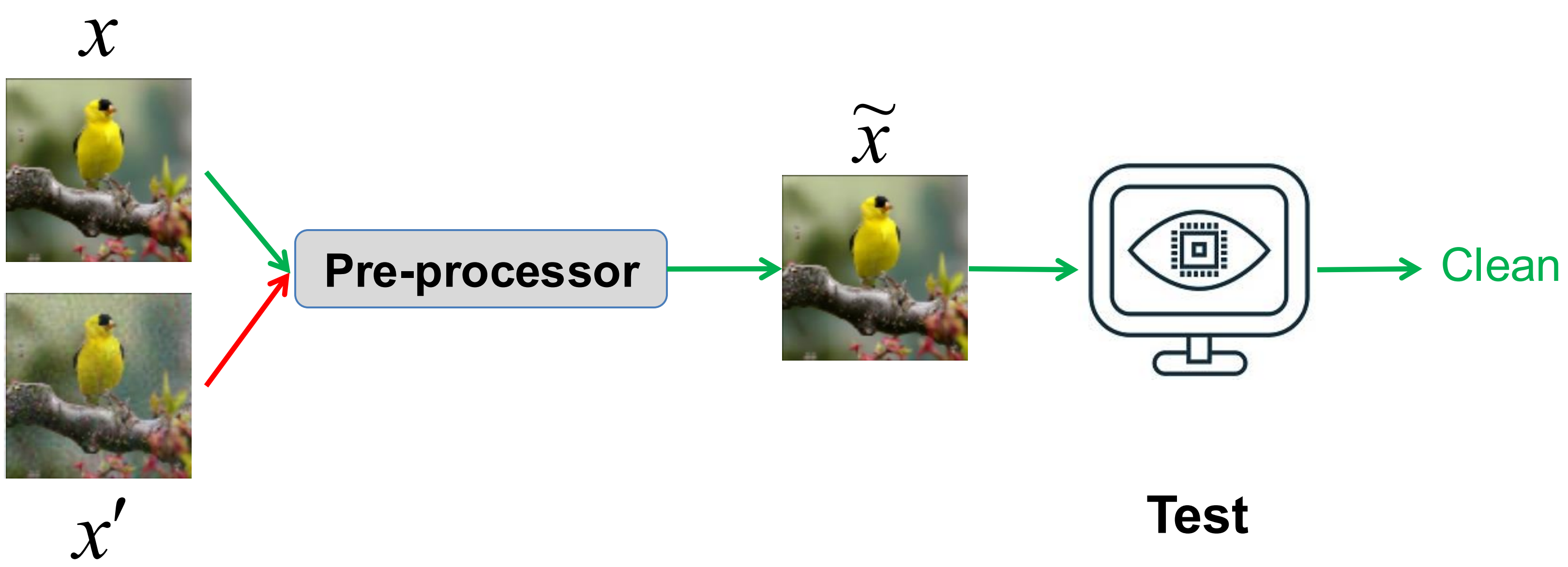


On Detecting Adversarial Perturbations. Metzen et al. ICLR 2017

Black-box Defense

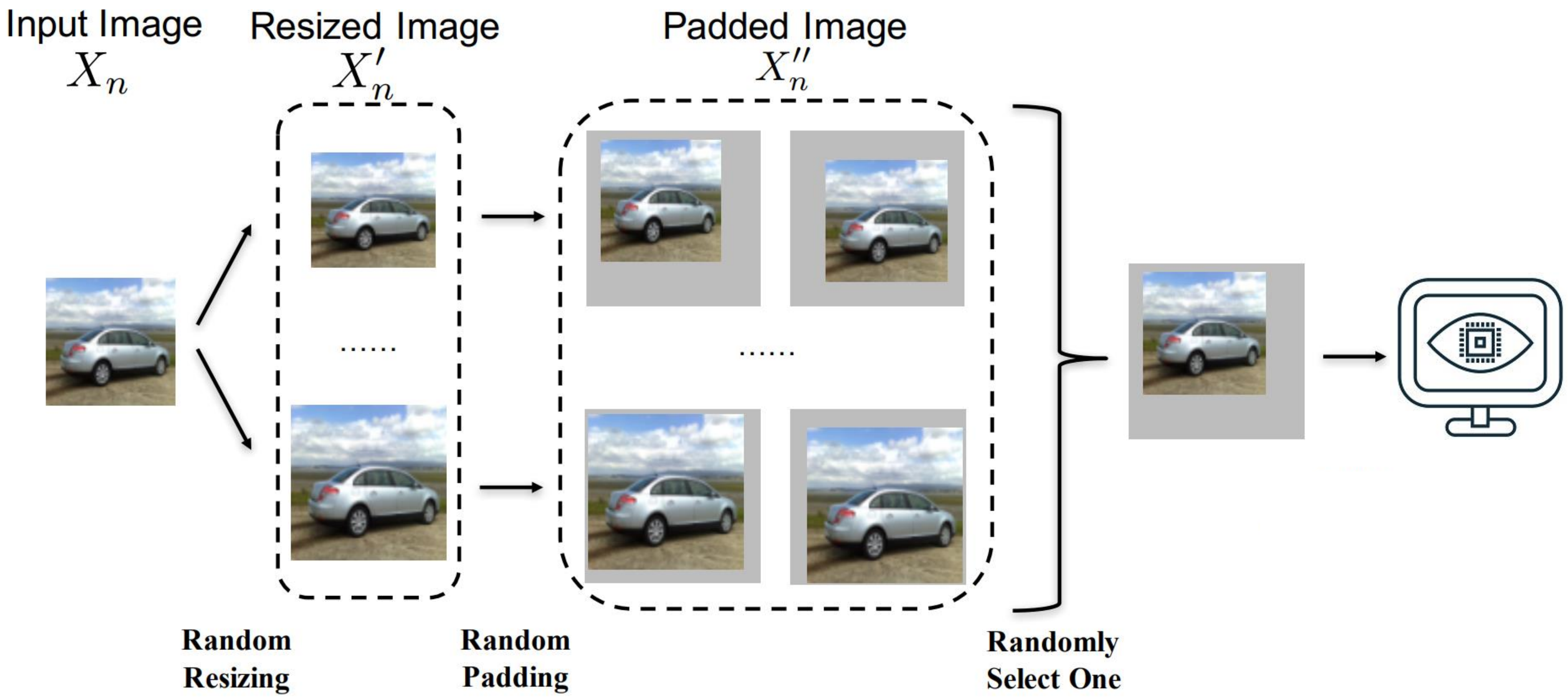


Black-box Defense: Pre-processing



Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. Xu et al. NDSS 2018

Black-box Defense: Pre-processing: Randomization



Mitigating Adversarial Effects Through Randomization. Xie et al. ICLR 2018

Black-box Defense: Pre-processing: Denoising

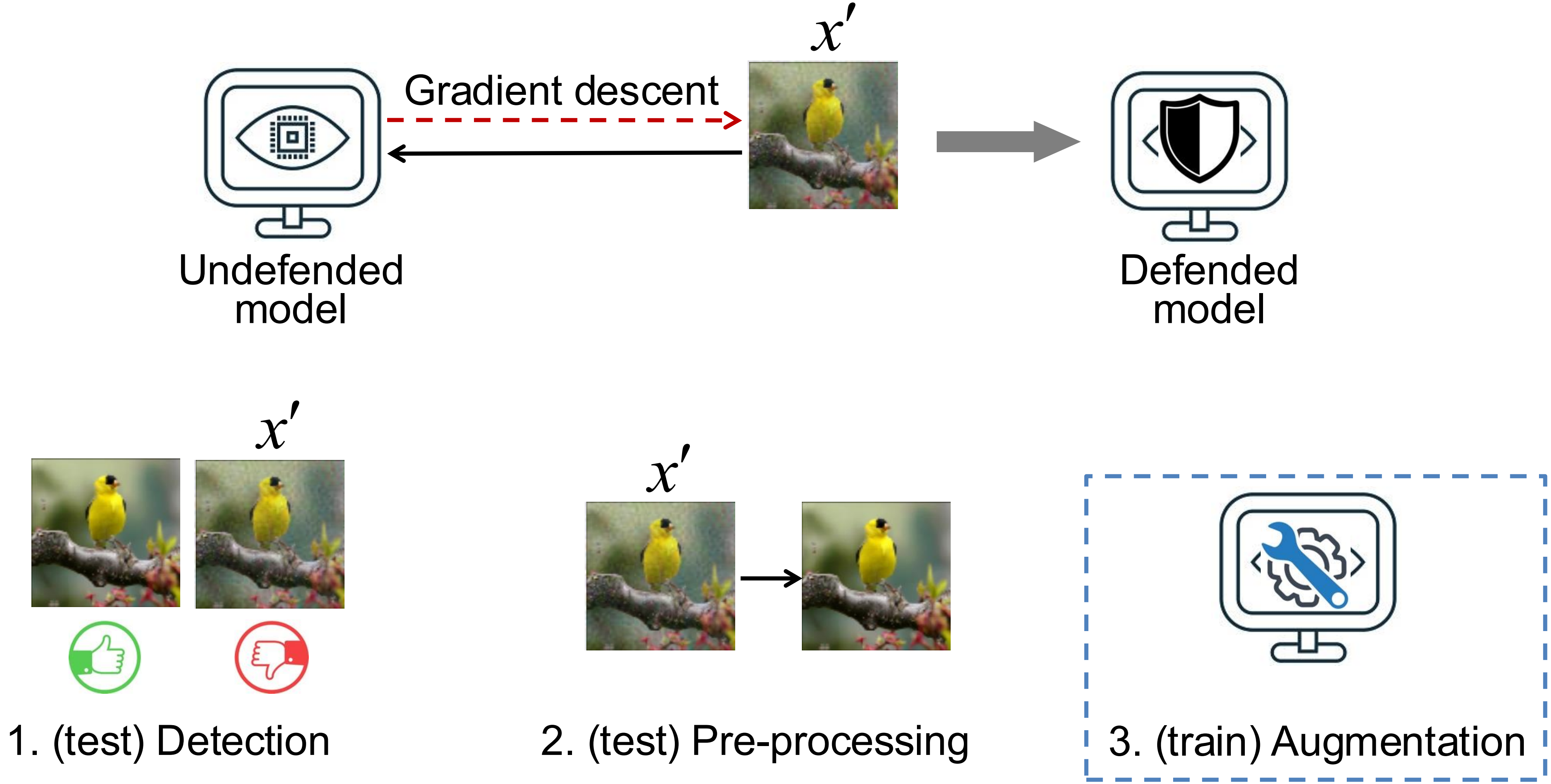
Shield: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression. Das et al. KDD 2018

Black-box Defense: Pre-processing: Denoising

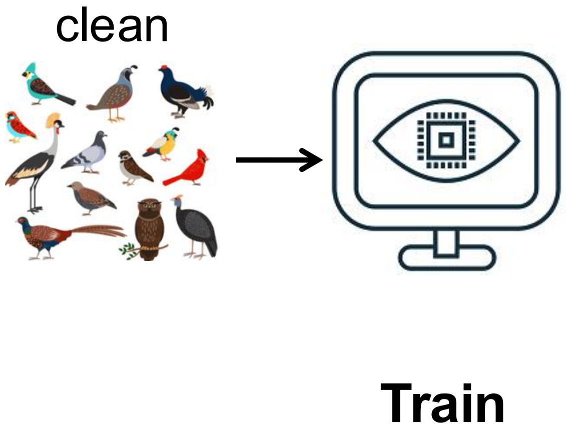
Defense	No Attack $\lvert L_2 \rvert = 0$	**CW-L2** ($\kappa = 0$) $\lvert L_2 \rvert = .0025$
No Defense	*75.59*	*10.29*
JPEG [quality=100]	**74.95**	74.37
JPEG [quality=90]	74.83	**74.43**
JPEG [quality=80]	74.23	73.92
JPEG [quality=70]	73.61	73.11
JPEG [quality=60]	72.97	72.46
JPEG [quality=50]	72.32	71.86
JPEG [quality=40]	71.48	71.03
JPEG [quality=30]	70.08	69.63
JPEG [quality=20]	67.72	67.32

Shield: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression. Das et al. KDD 2018

Black-box Defense

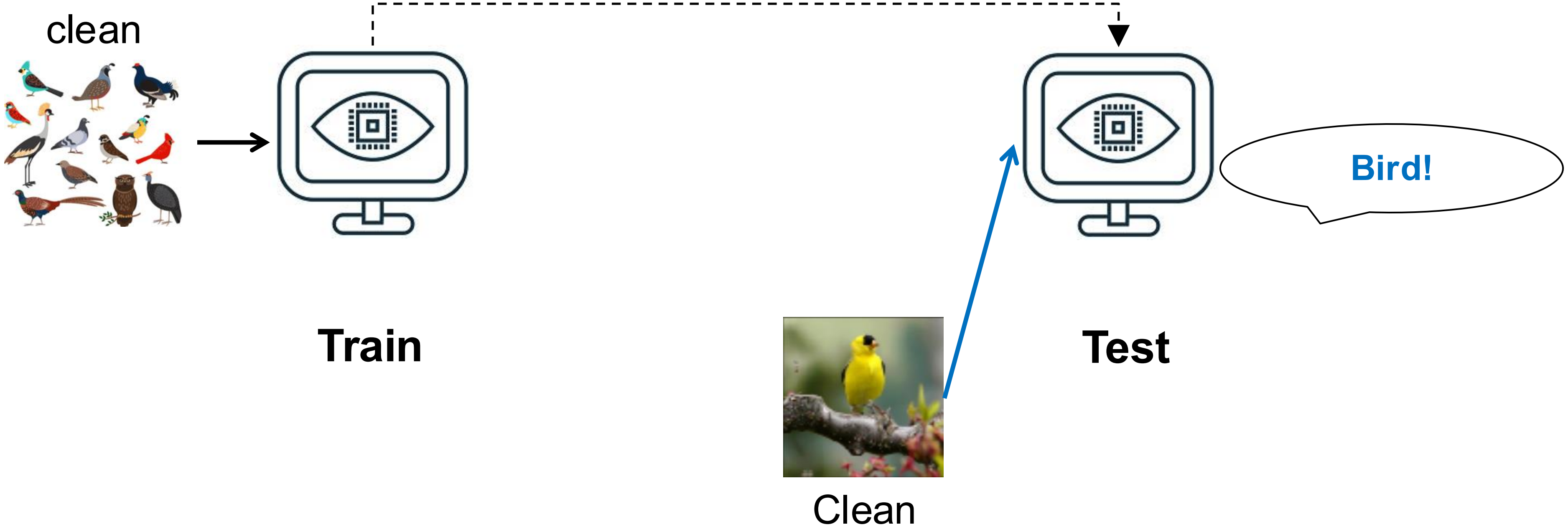


Black-box Defense: Augmentation



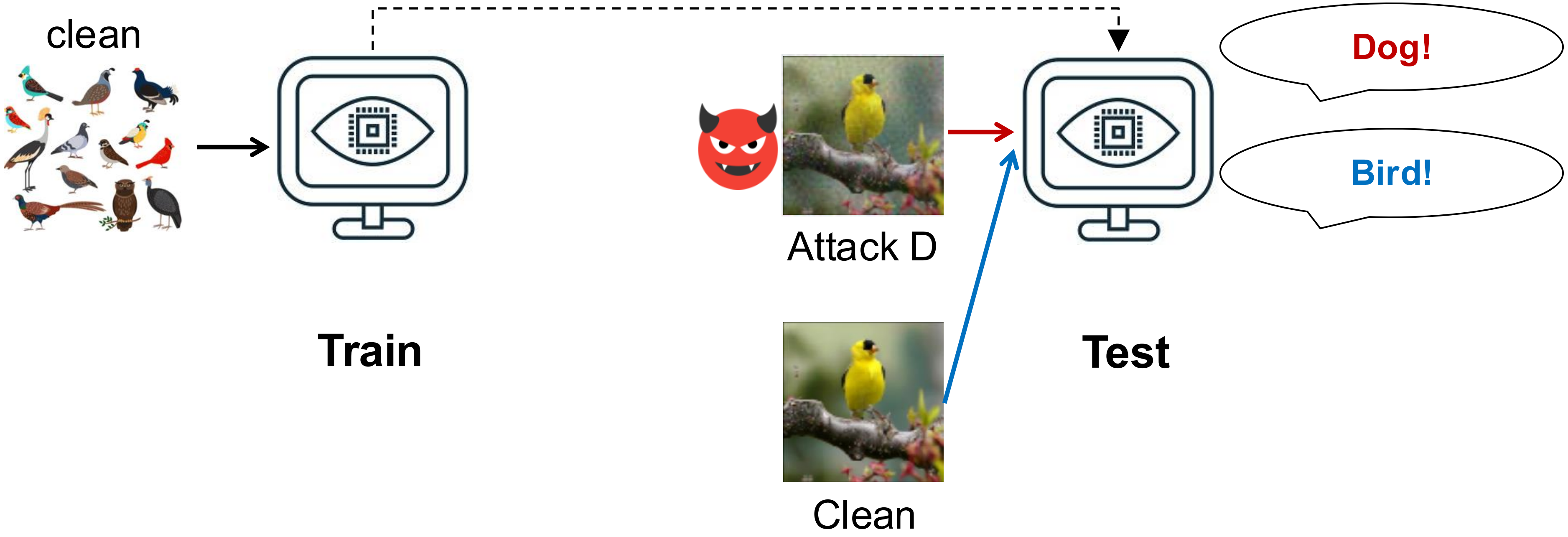
On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

Black-box Defense: Augmentation



On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

Black-box Defense: Augmentation



On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

Black-box Defense: Augmentation

Adversarial

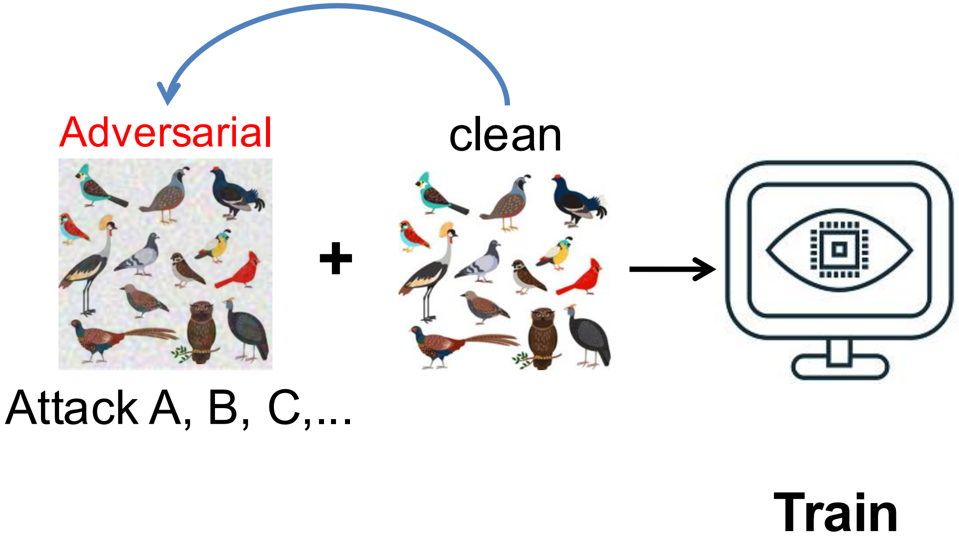
Attack A, B, C,...

clean

Train

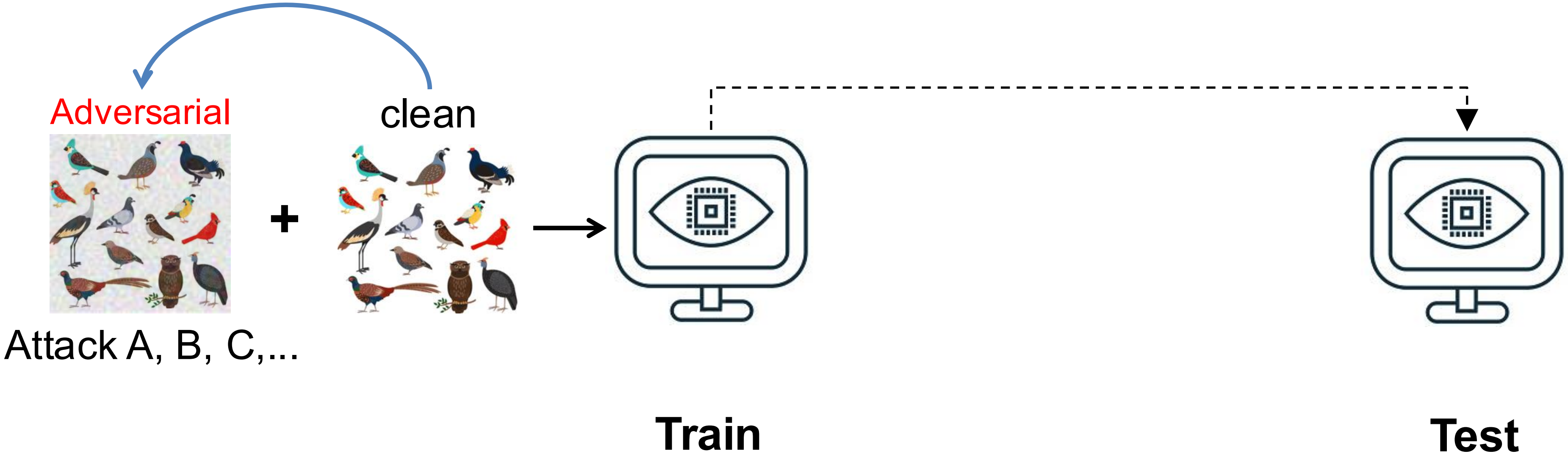
On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

Black-box Defense: Augmentation



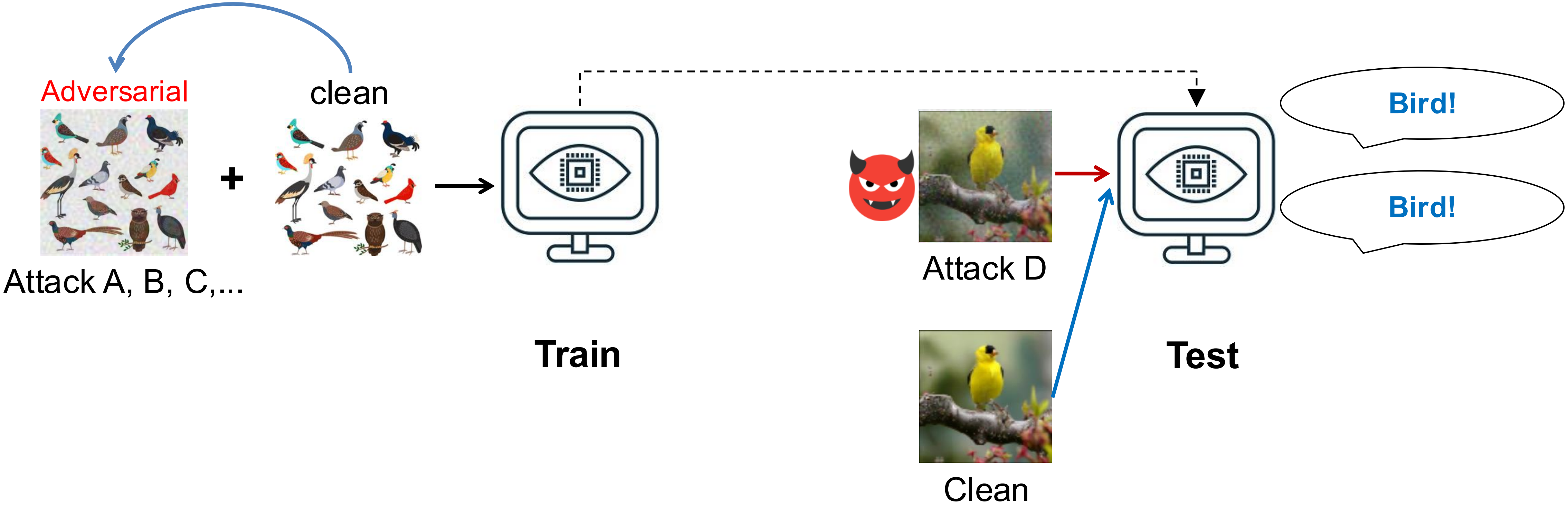
On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

Black-box Defense: Augmentation



On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

Black-box Defense: Augmentation

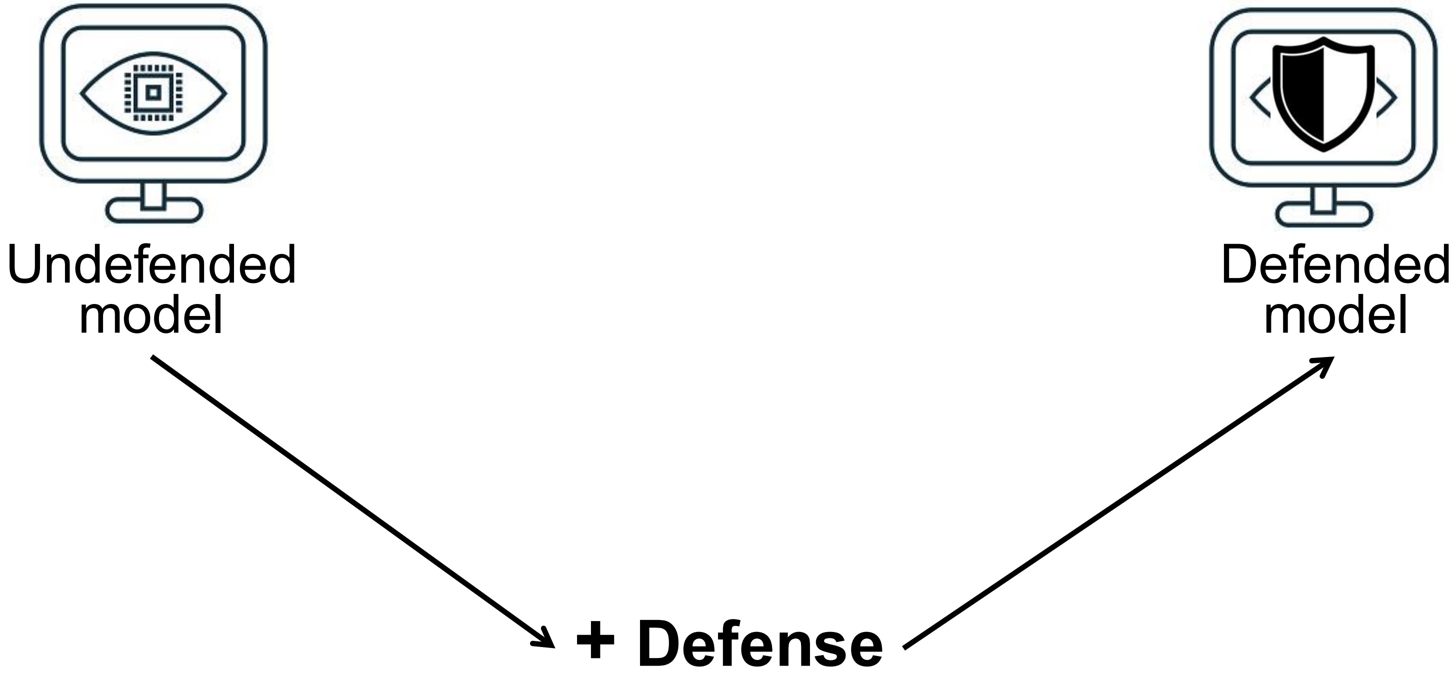


On the (Statistical) Detection of Adversarial Examples. Grosse et al. arXiv 2017

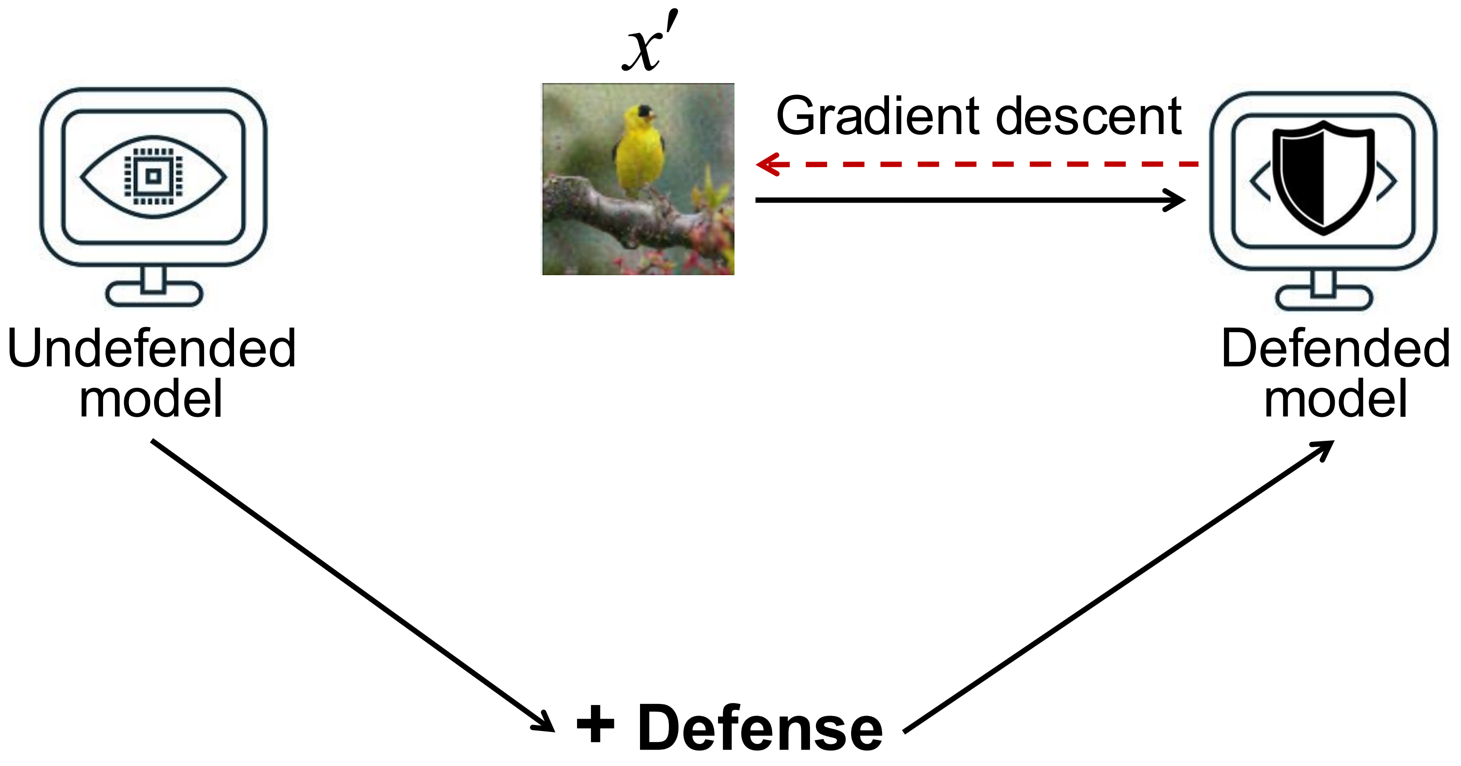
(Empirical) Defenses

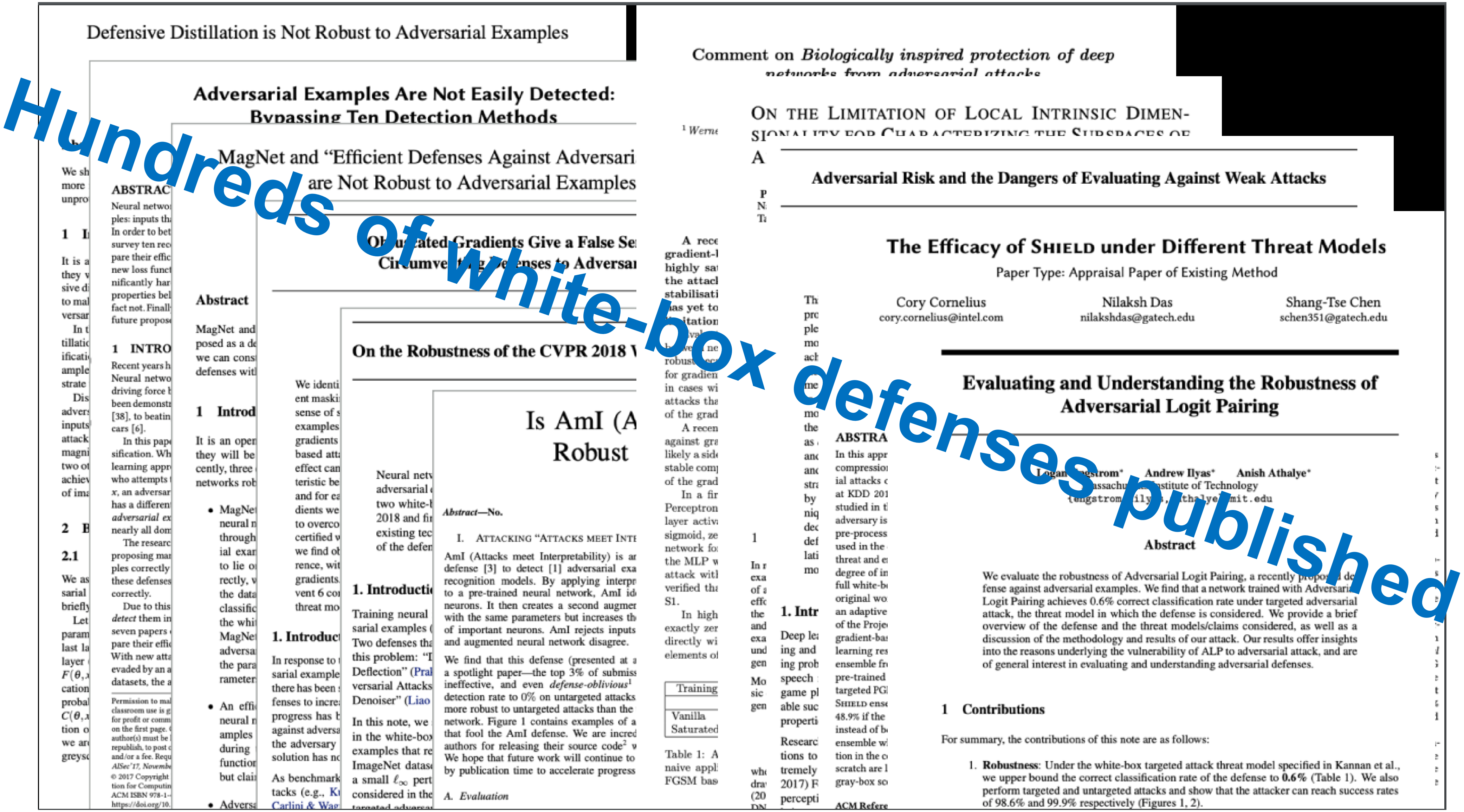
- Black-box:
  Attack doesn't know defense and keeps fixed

- White-box:
  Attack knows and can be adapted to defense


Undefended model
+ Defense
Defended model

White-box Defense





https://nicholas.carlini.com/slides/2021_aiforgood_nonrobustml.pdf

Towards Evaluating the Robustness of Neural Networks

IEEE Symposium on Security and Privacy, 2017. **Best Student Paper.**

Nicholas Carlini and David Wagner

Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods

ACM Workshop on Artificial Intelligence and Security, 2017. **Finalist, Best Paper.**

Nicholas Carlini and David Wagner

All (except one) are broken!

On the Robustness of the CVPR 2018 White-Box Adversarial Example Defenses

Computer Vision: Challenges and Opportunities for Privacy and Security, 2018.

Anish Athalye and **Nicholas Carlini**

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples

International Conference on Machine Learning, 2018. **Best Paper.**

Anish Athalye*, **Nicholas Carlini***, and David Wagner

On Adaptive Attacks to Adversarial Example Defenses

NeurIPS, 2020.

Florian Tramer, **Nicholas Carlini**, Wieland Brendel, Aleksander Madry

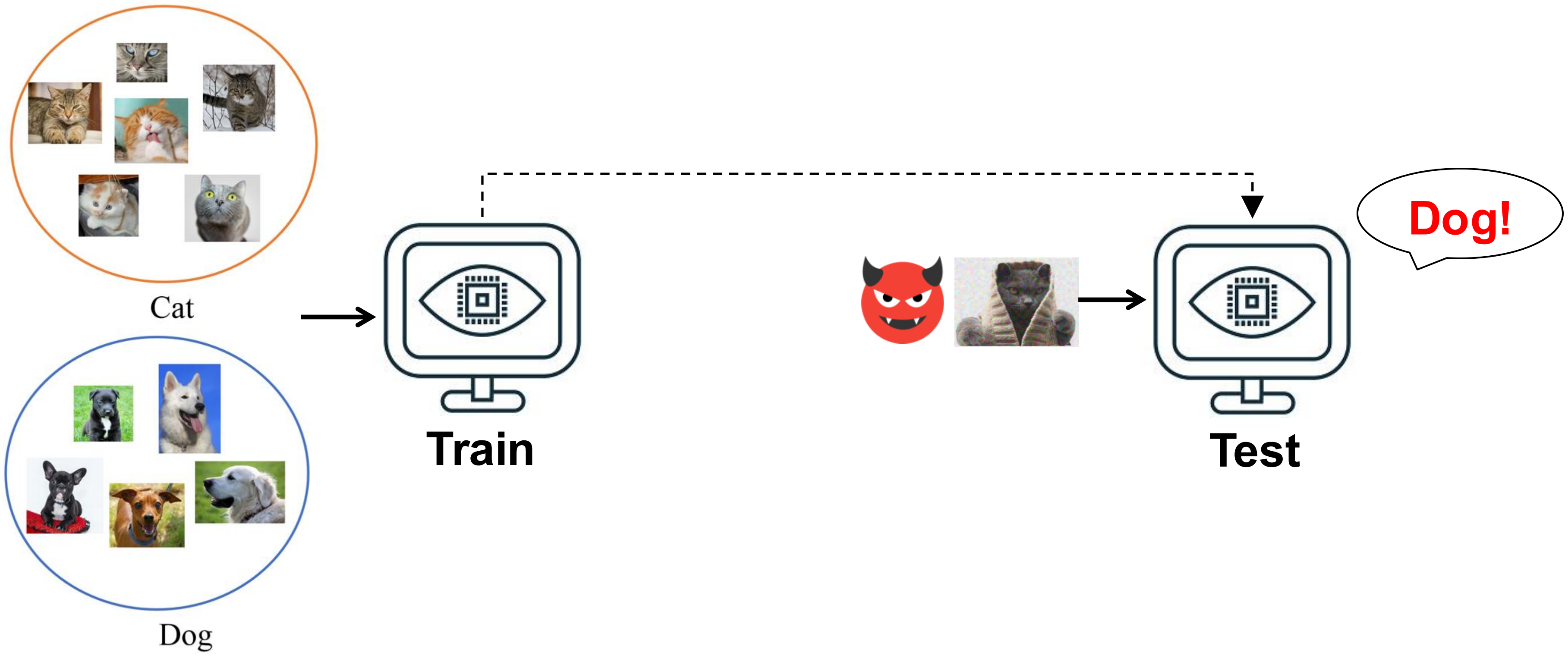
Evading Adversarial Example Detection Defenses with Orthogonal Projected Gradient Descent

ICLR, 2022.

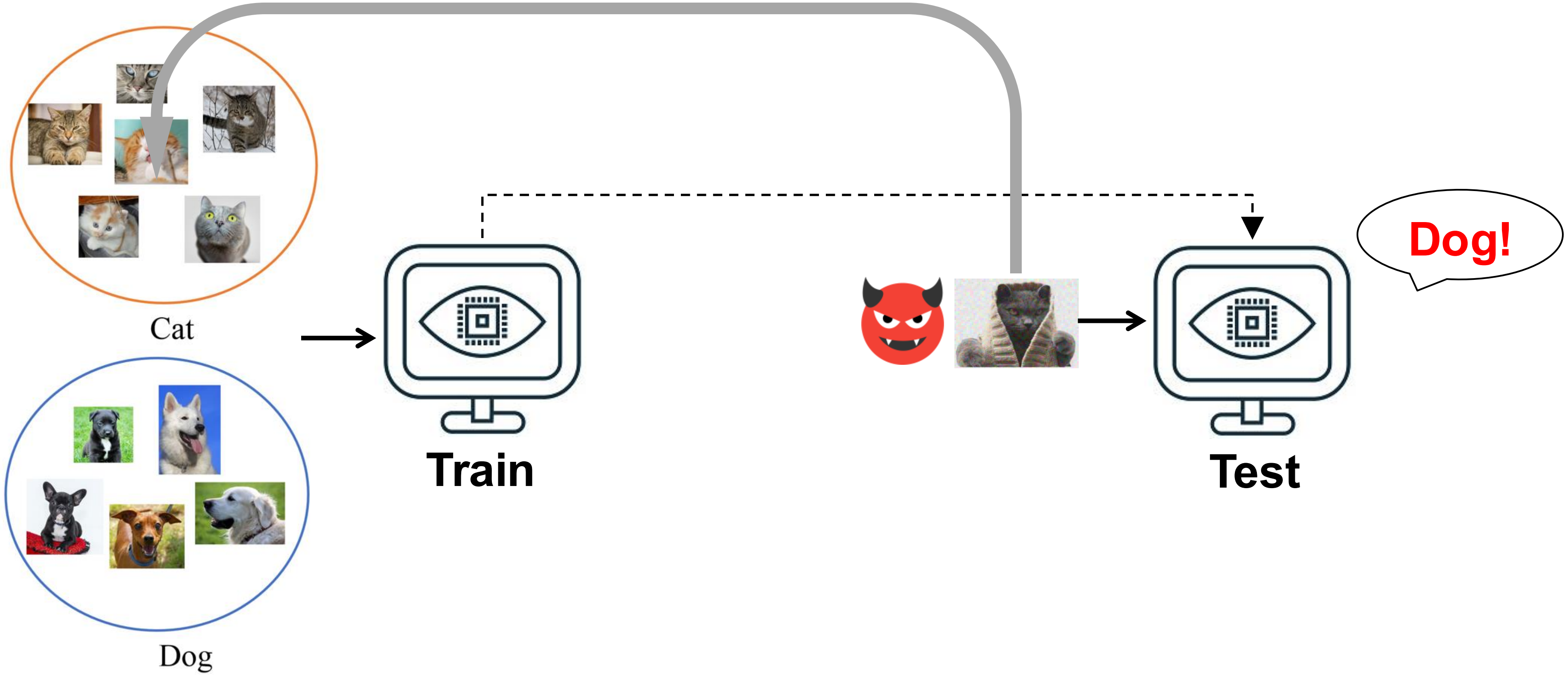
Oliver Bryniarski, Nabeel Hingun, Pedro Pachuca, Vincent Wang, **Nicholas Carlini**

Nicholas Carlini
Research Scientist, Google DeepMind
nicholas [at] carlini [dot] com
GitHub | Google Scholar

That One Defense: Adversarial Training



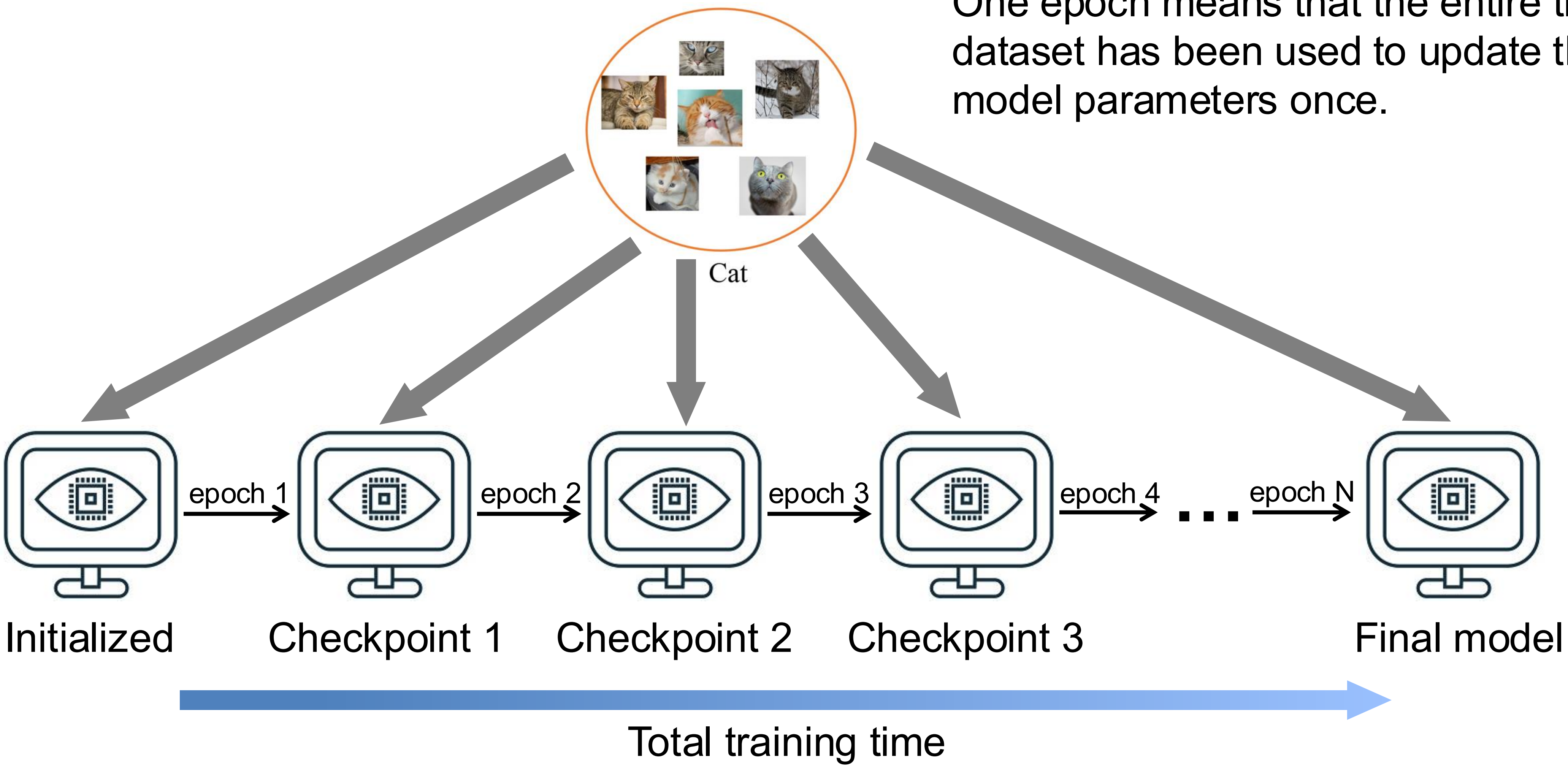
That One Defense: Adversarial Training



Towards Deep Learning Models Resistant to Adversarial Attacks. Madry et al. ICLR 2018

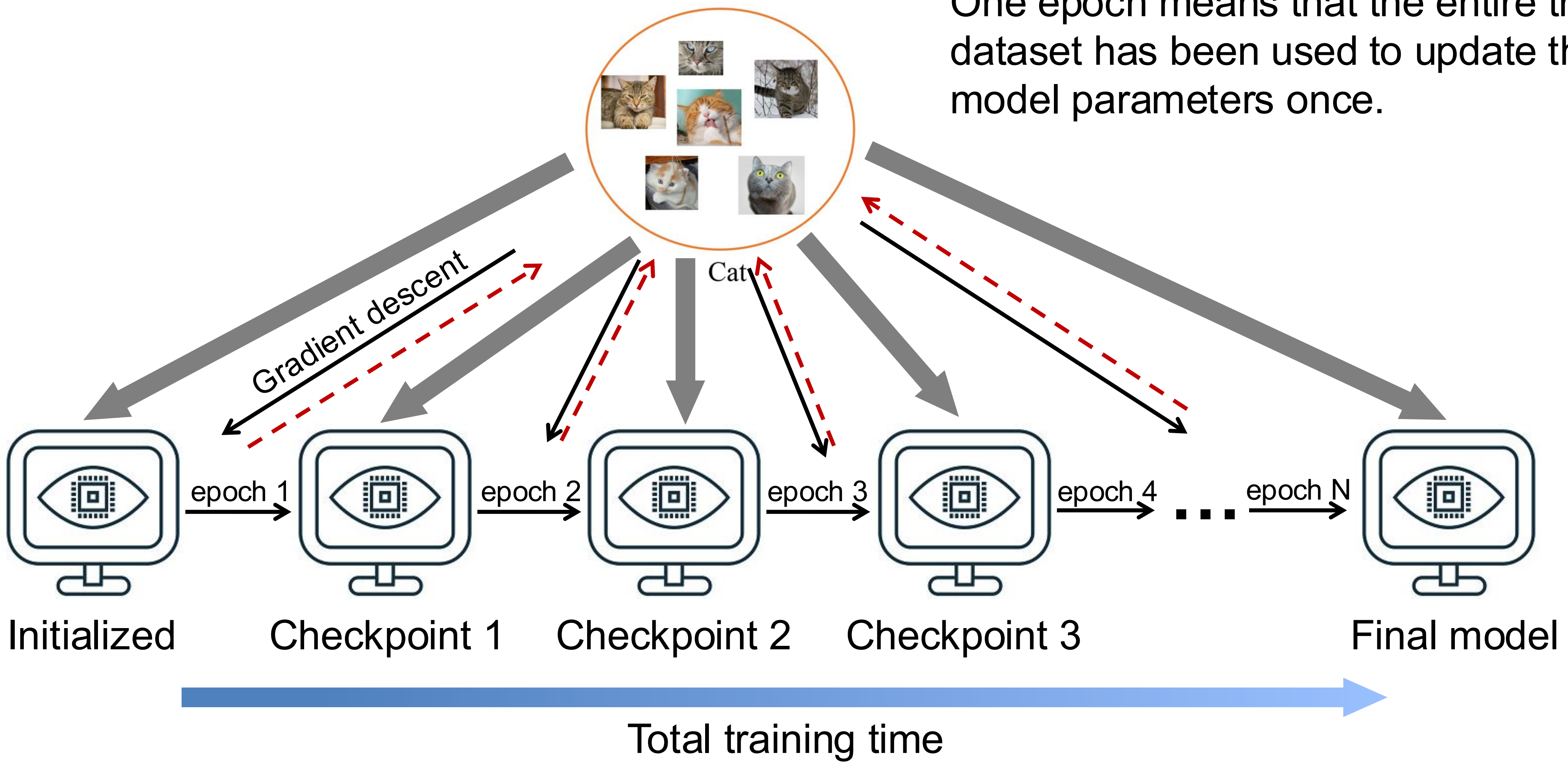
That One Defense: Adversarial Training

One epoch means that the entire training dataset has been used to update the model parameters once.



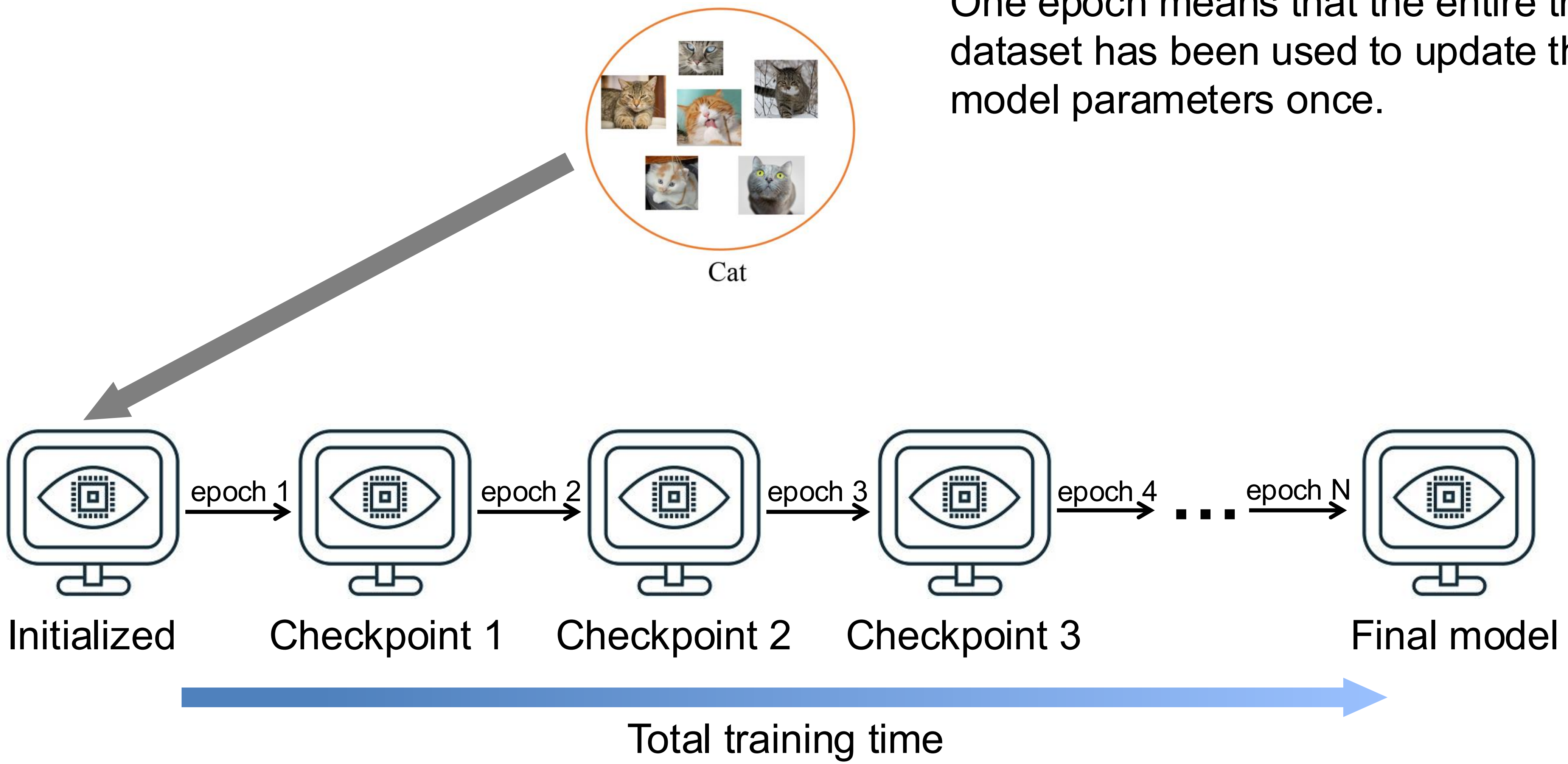
That One Defense: Adversarial Training

One epoch means that the entire training dataset has been used to update the model parameters once.



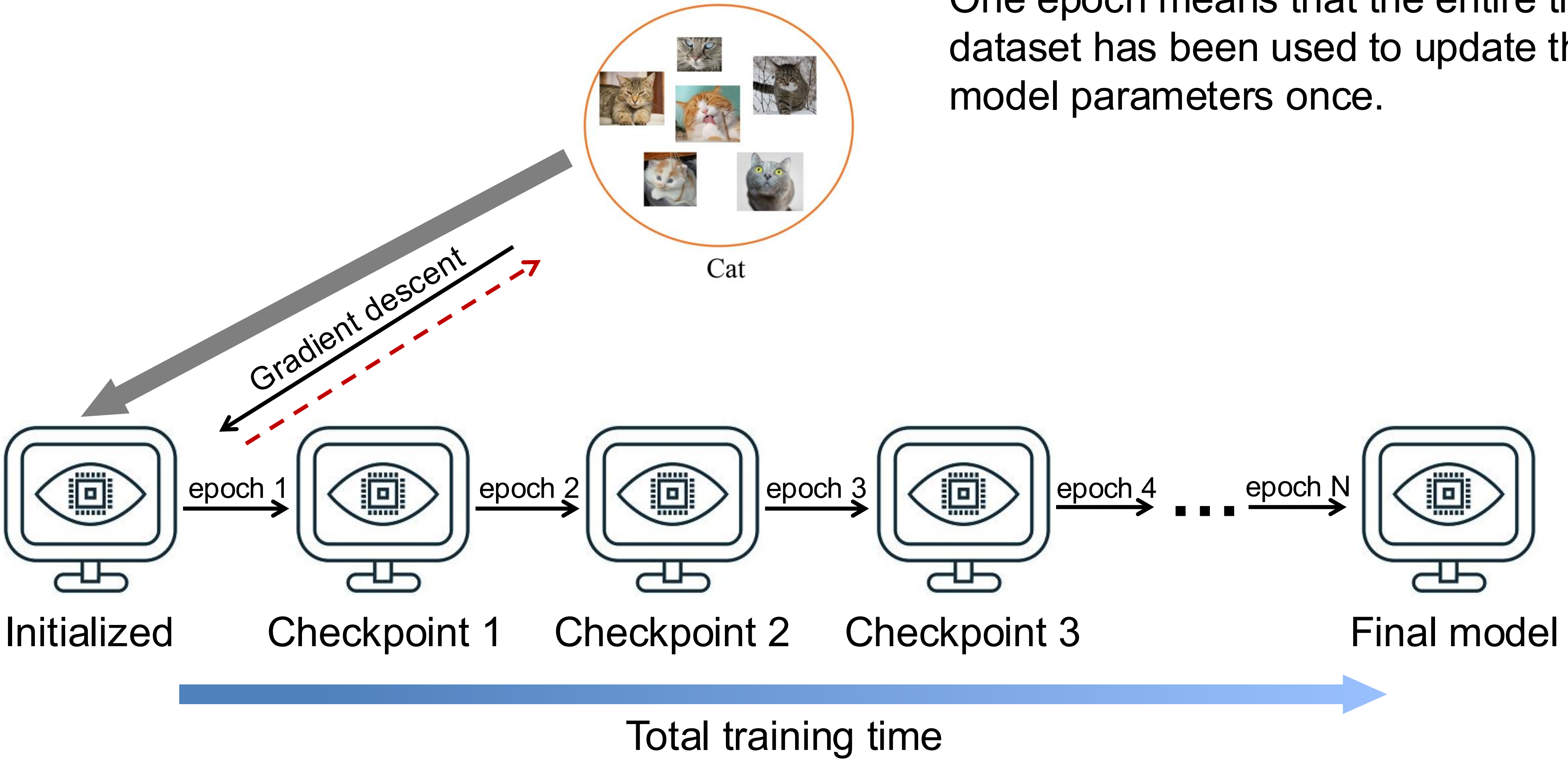
That One Defense: Adversarial Training

One epoch means that the entire training dataset has been used to update the model parameters once.



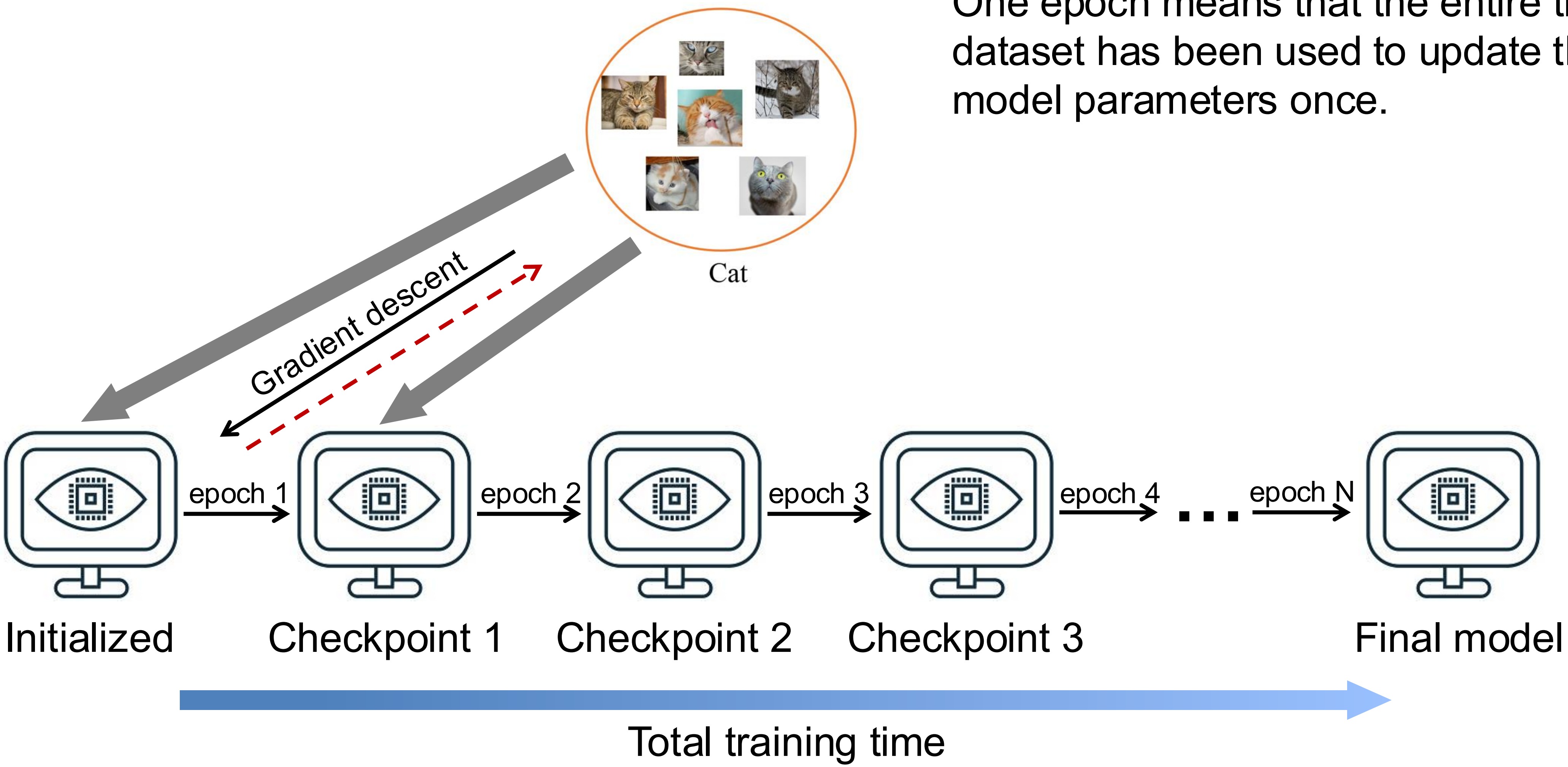
That One Defense: Adversarial Training

One epoch means that the entire training dataset has been used to update the model parameters once.



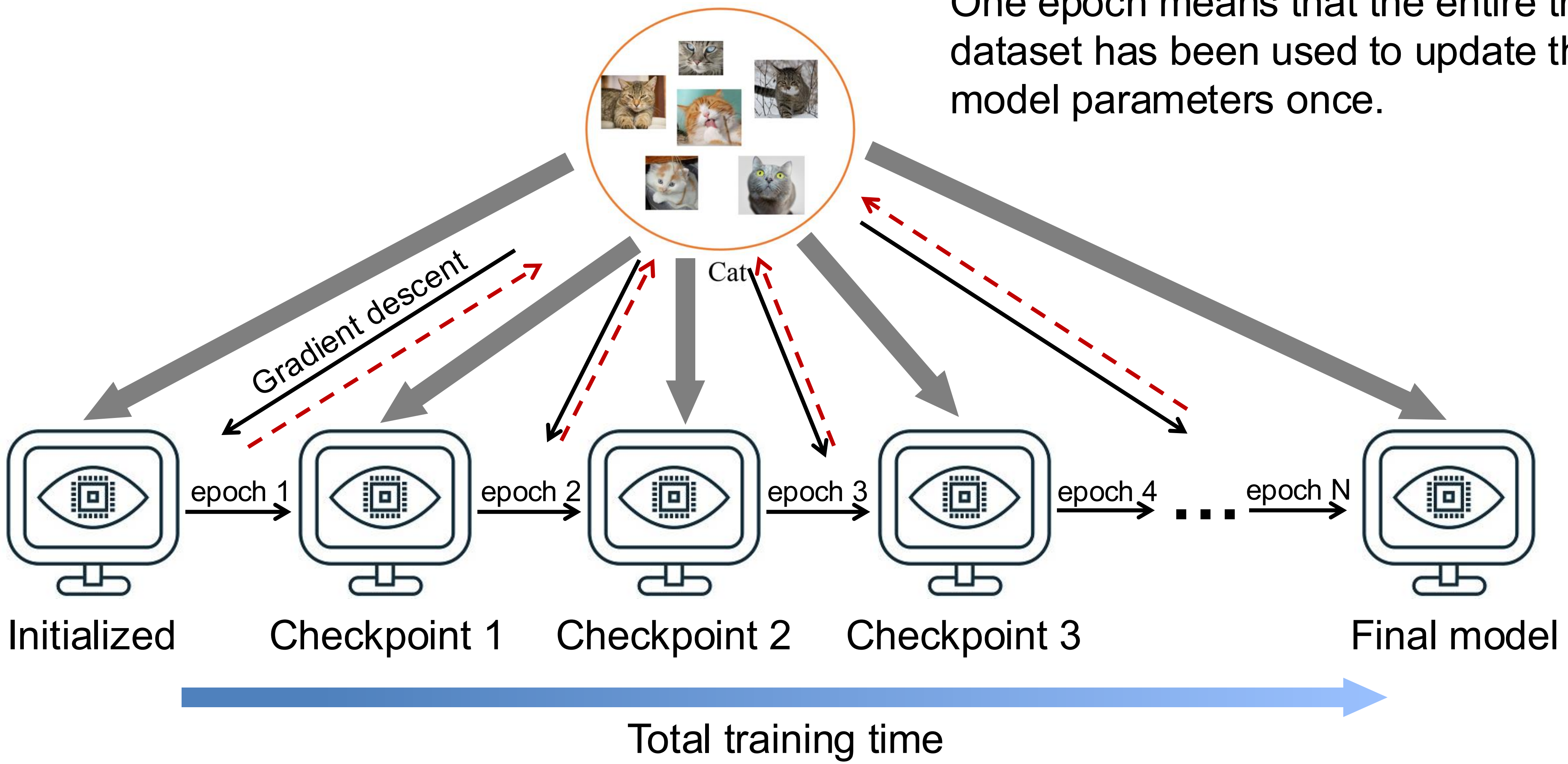
That One Defense: Adversarial Training

One epoch means that the entire training dataset has been used to update the model parameters once.



That One Defense: Adversarial Training

One epoch means that the entire training dataset has been used to update the model parameters once.



Method 2: Projected Gradient Descent (PGD)

Objective: $x' = \arg\max_{x} d(y, y_{\text{bird}})$

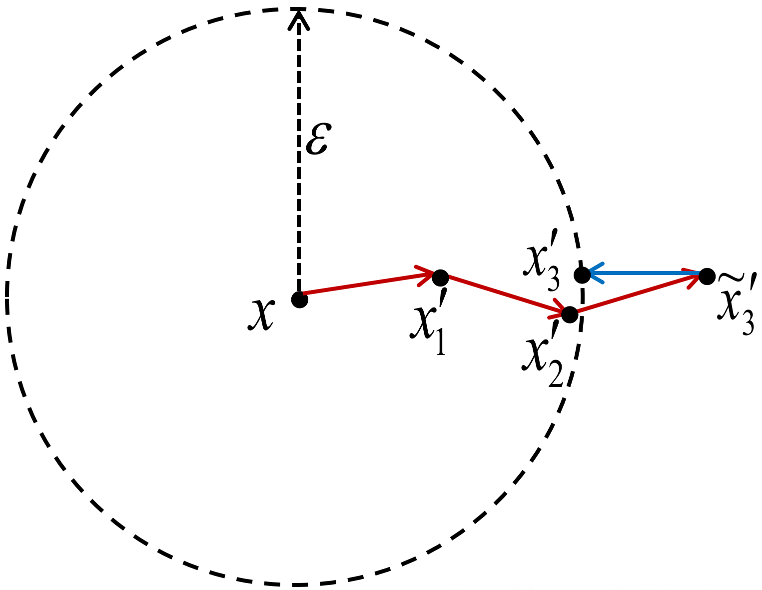
s.t. $\|$  $-$  $\|_p \le \varepsilon$

x' x

$$x'_0 = x,$$

$$x'_{i+1} = x'_i - \alpha \cdot \text{sign}(\nabla_x d(y, y_{\text{bird}}))$$

$$x' \leftarrow \text{project}(x' - x, -\varepsilon, \varepsilon)$$



Explaining and Harnessing Adversarial Examples. Goodfellow et al. ICLR 2015
Adversarial Examples in the Physical World. Kurakin et al. ICLR workshop 2017

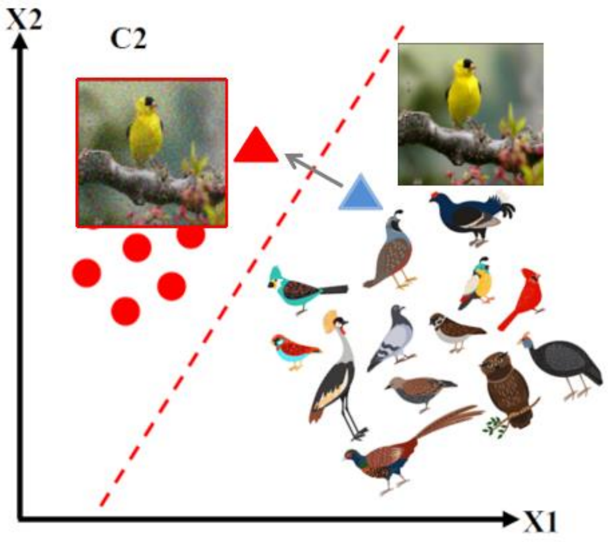
Summary

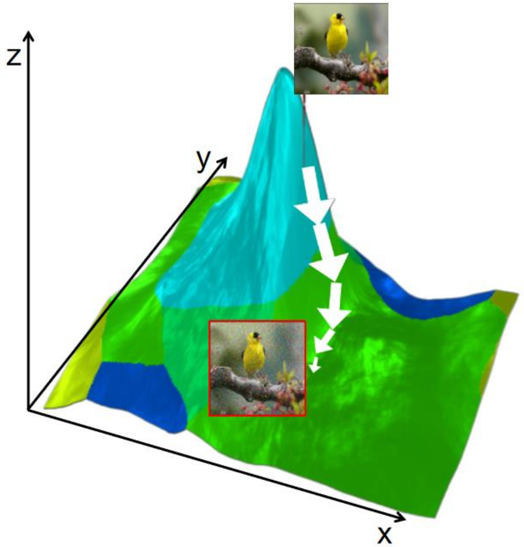
Success of computer vision

↳ Failures Against Abnormal Examples

↳ Adversarial Examples: Definition

↳ **Attacks: Illustration, Formulation, Optimization**





Summary

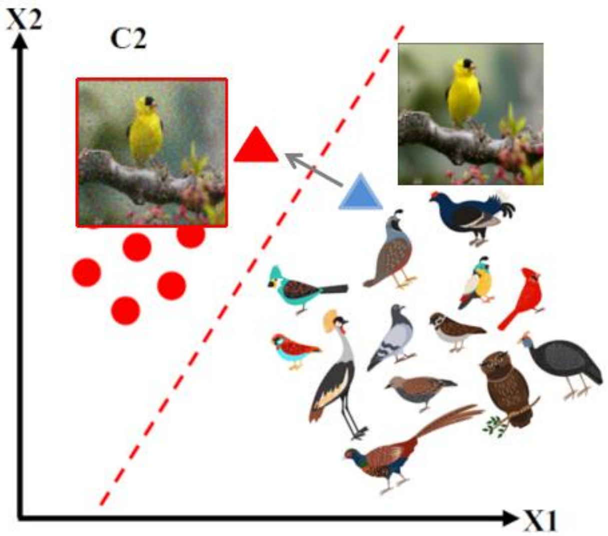
Success of computer vision

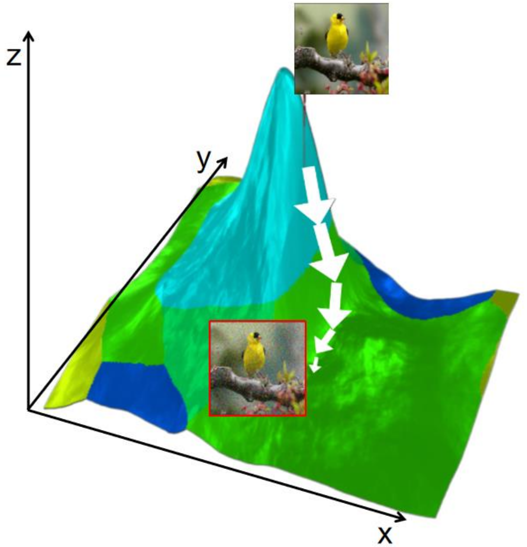
↳Failures Against Abnormal Examples

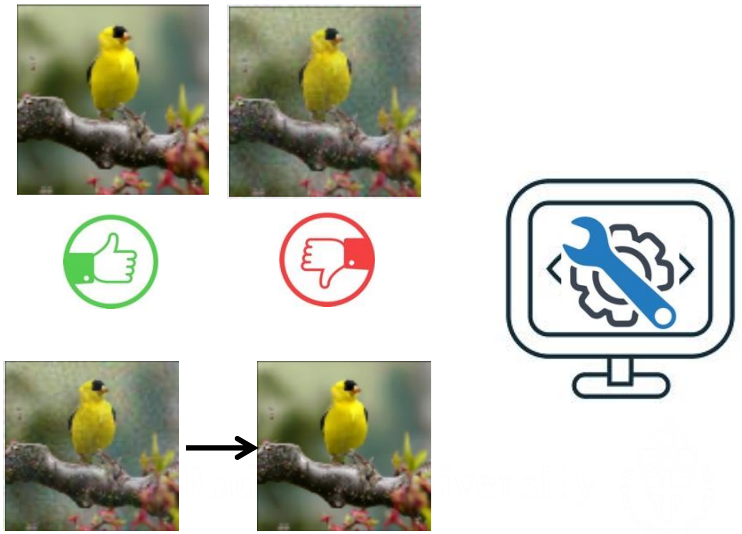
↳Adversarial Examples: Definition

↳Attacks: Illustration, Formulation, Optimization

↳Defenses: (test) Detection/Pre-processing, (train) Augmentation







Code Example

Optimize adversarial examples using PyTorch

https://savan77.github.io/blog/imagenet_adv_examples.html