Adversarial attack on deep learning

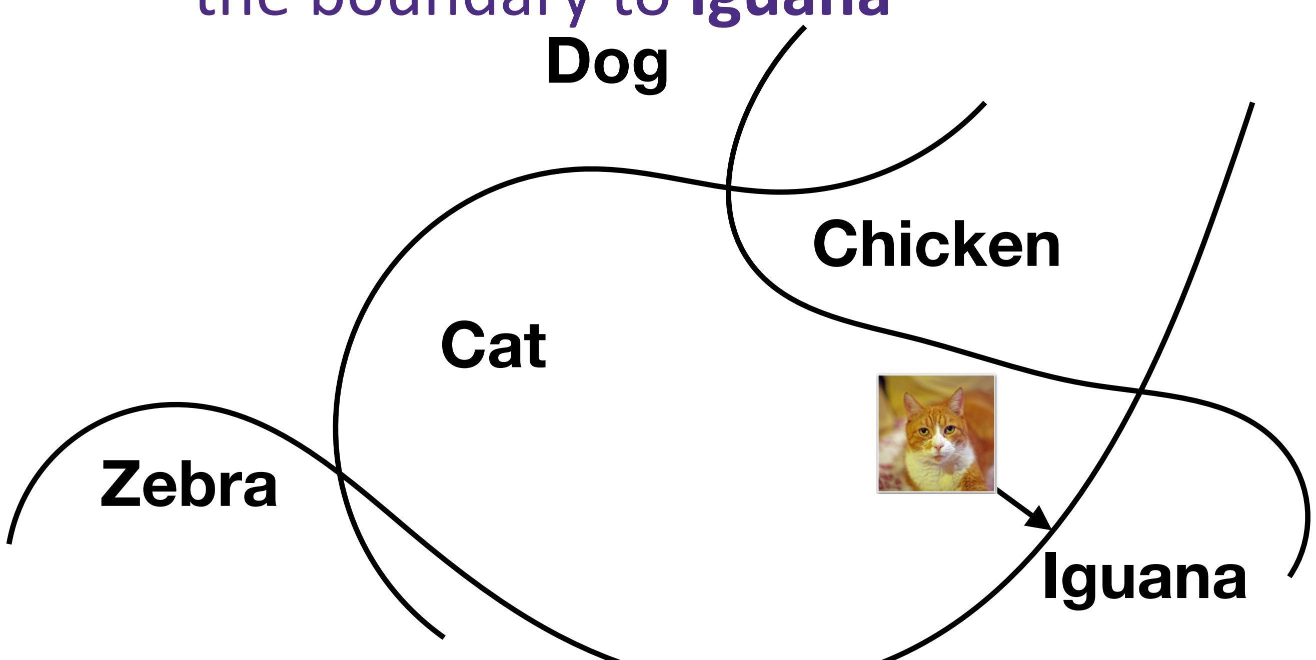
White-box Adversarial examples

- In **adversarial examples** the goal of an adversary is to generate an image that looks like a **cat**, but is classified as **iguana** (for a specific given NN classifier)

White-box Adversarial examples

- In **adversarial examples** the goal of an adversary is to generate an image that looks like a **cat**, but is classified as **iguana** (for a specific given NN classifier)

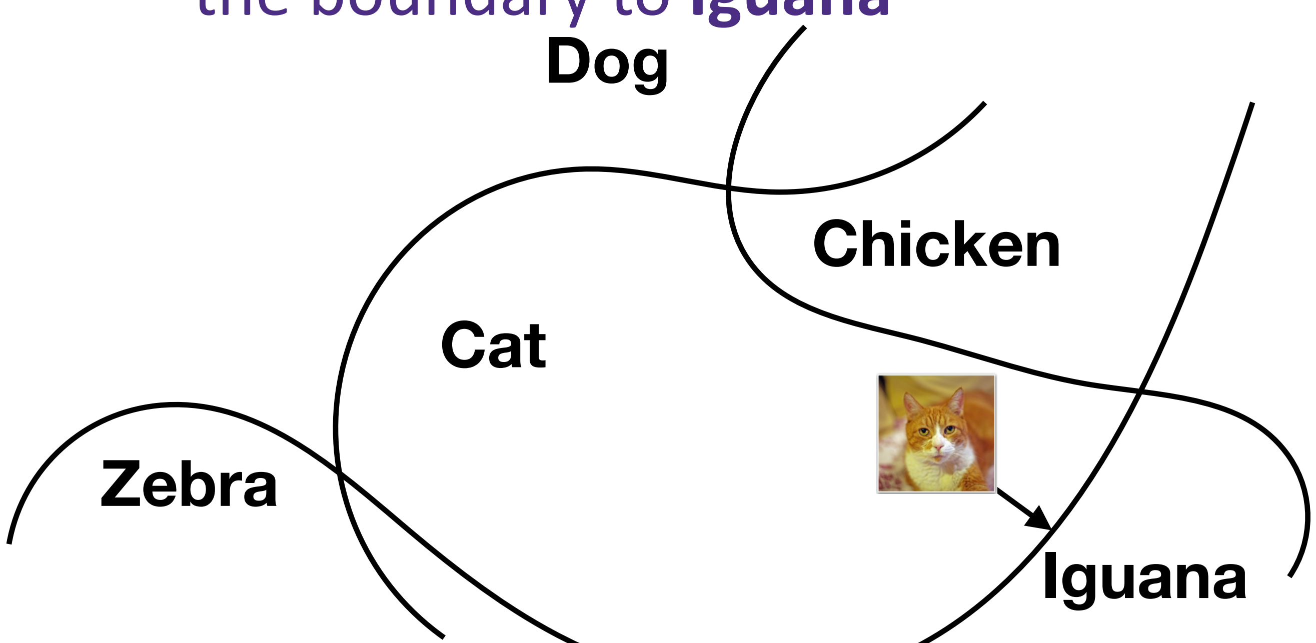
- Main idea:
  - the given CNN has complex decision boundaries around the sample
  - find the minimum perturbation you can do to the pixel values, while crossing the boundary to **Iguana**



White-box Adversarial examples

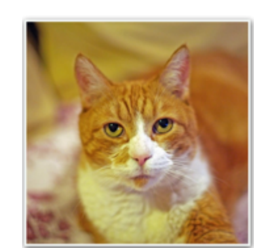
- In **adversarial examples** the goal of an adversary is to generate an image that looks like a **cat**, but is classified as **iguana** (for a specific given NN classifier)

- Main idea:
  - the given CNN has complex decision boundaries around the sample
  - find the minimum perturbation you can do to the pixel values, while crossing the boundary to **Iguana**



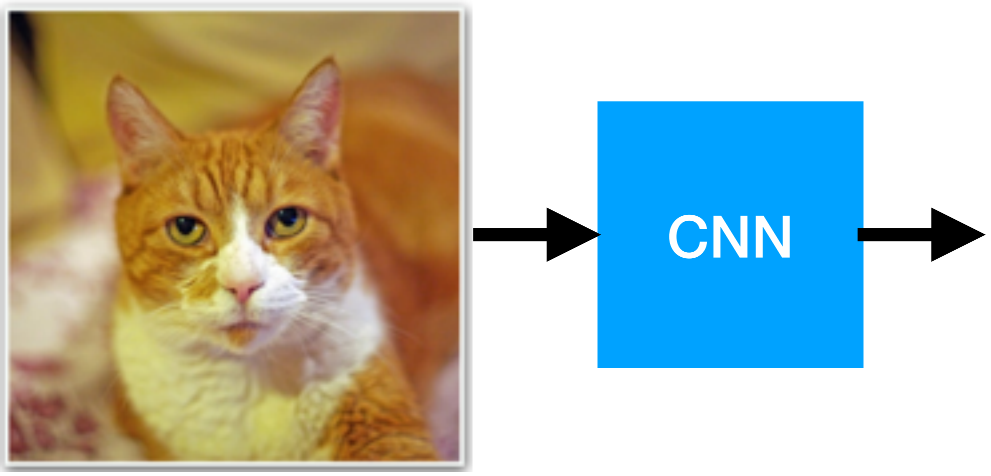
One could solve:

$$\min_{\text{image}} \ell(f_W(\text{image}), y_{\text{Iguana}})$$

subject to $\left\| \text{image} - \right.$  $\left. \right\|_\infty \leq \varepsilon$

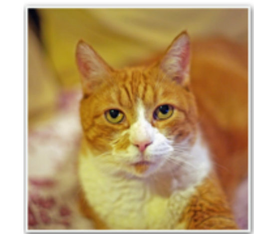
Why infinity norm?

White-box Adversarial examples

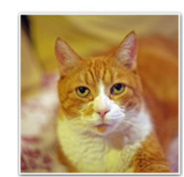


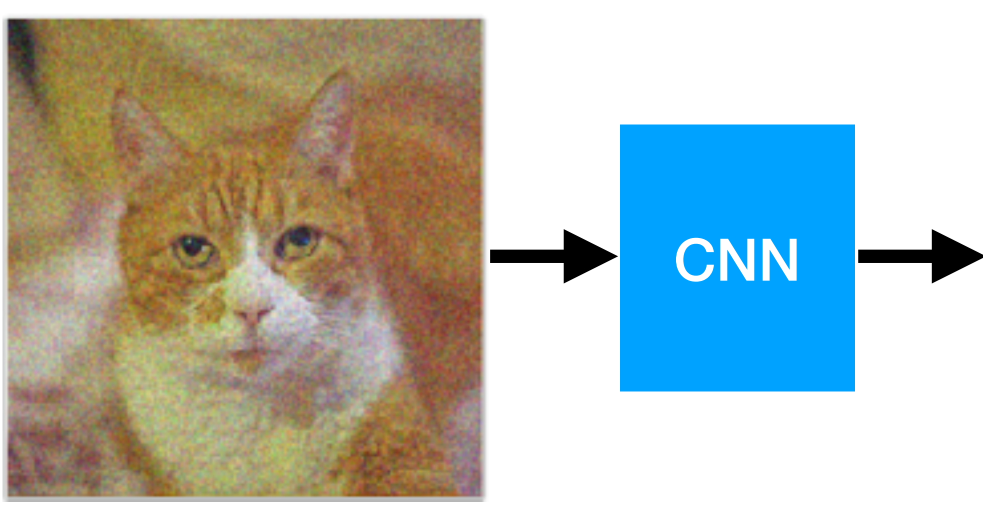
$$\begin{array}{cl} \begin{bmatrix} 0.92 \\ 0.02 \\ 0.01 \\ \vdots \\ \vdots \end{bmatrix} & \begin{array}{l} \text{“cat”} \\ \text{“dog”} \\ \text{“iguana”} \\ \\ \\ \end{array} \end{array}$$

$$\min_{\text{image}} \ell(f_W(\text{image}), y_{\text{Iguana}})$$

subject to $\Big\| \text{image} -$  $\Big\|_\infty \le \epsilon$

- In practice, you do not have to solve the optimization, but one gradient step is sufficient (Fast Gradient Sign Method)

 $- \epsilon \, \text{sign}(\nabla_x \ell(f_w(x), y_{\text{iguana}}))$

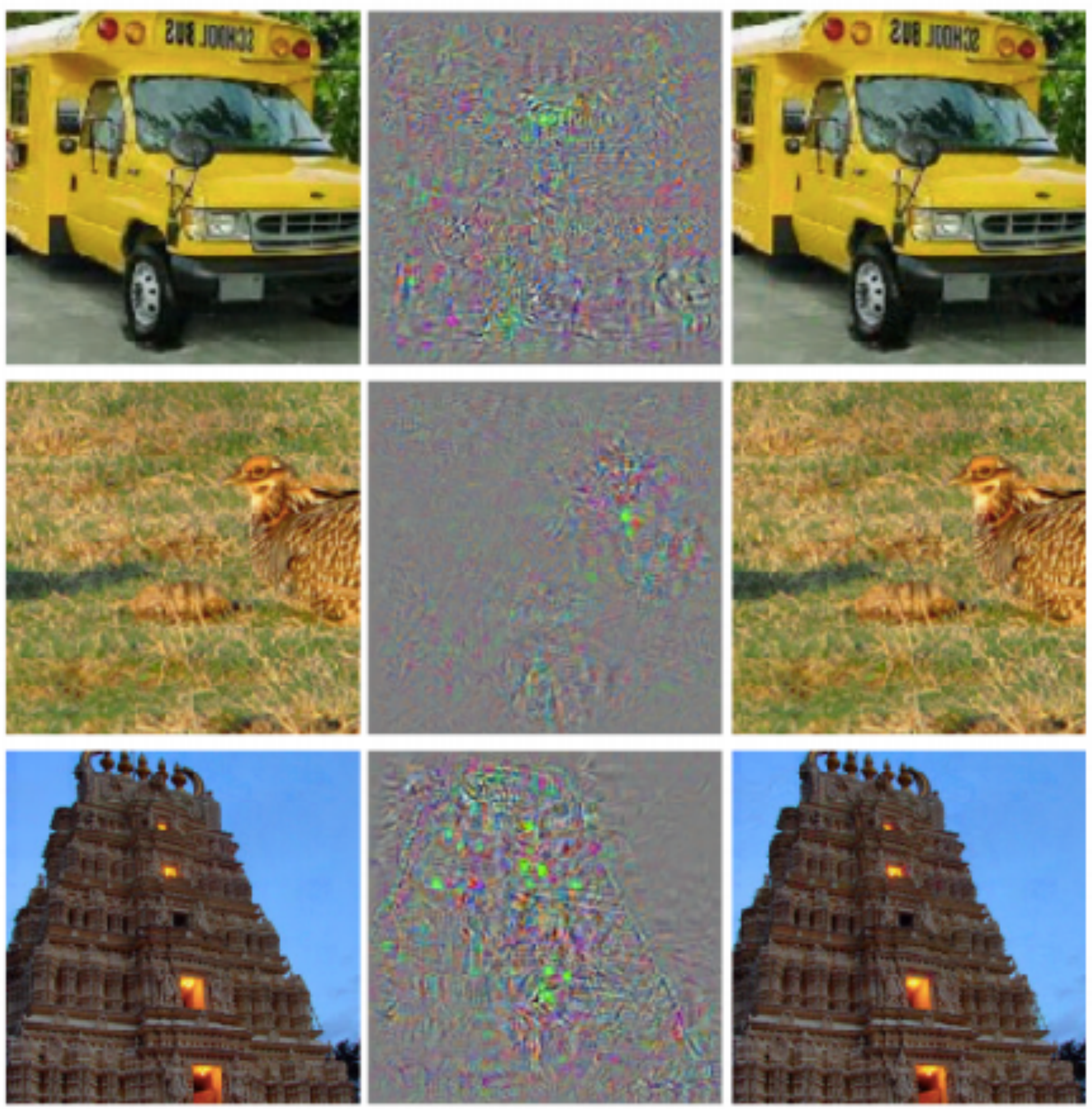


$f_w(x)$

$$\begin{array}{cl} \begin{bmatrix} 0.02 \\ 0.02 \\ 0.94 \\ \vdots \\ \vdots \end{bmatrix} & \begin{array}{l} \text{“cat”} \\ \text{“dog”} \\ \text{“iguana”} \\ \\ \\ \end{array} \end{array}$$

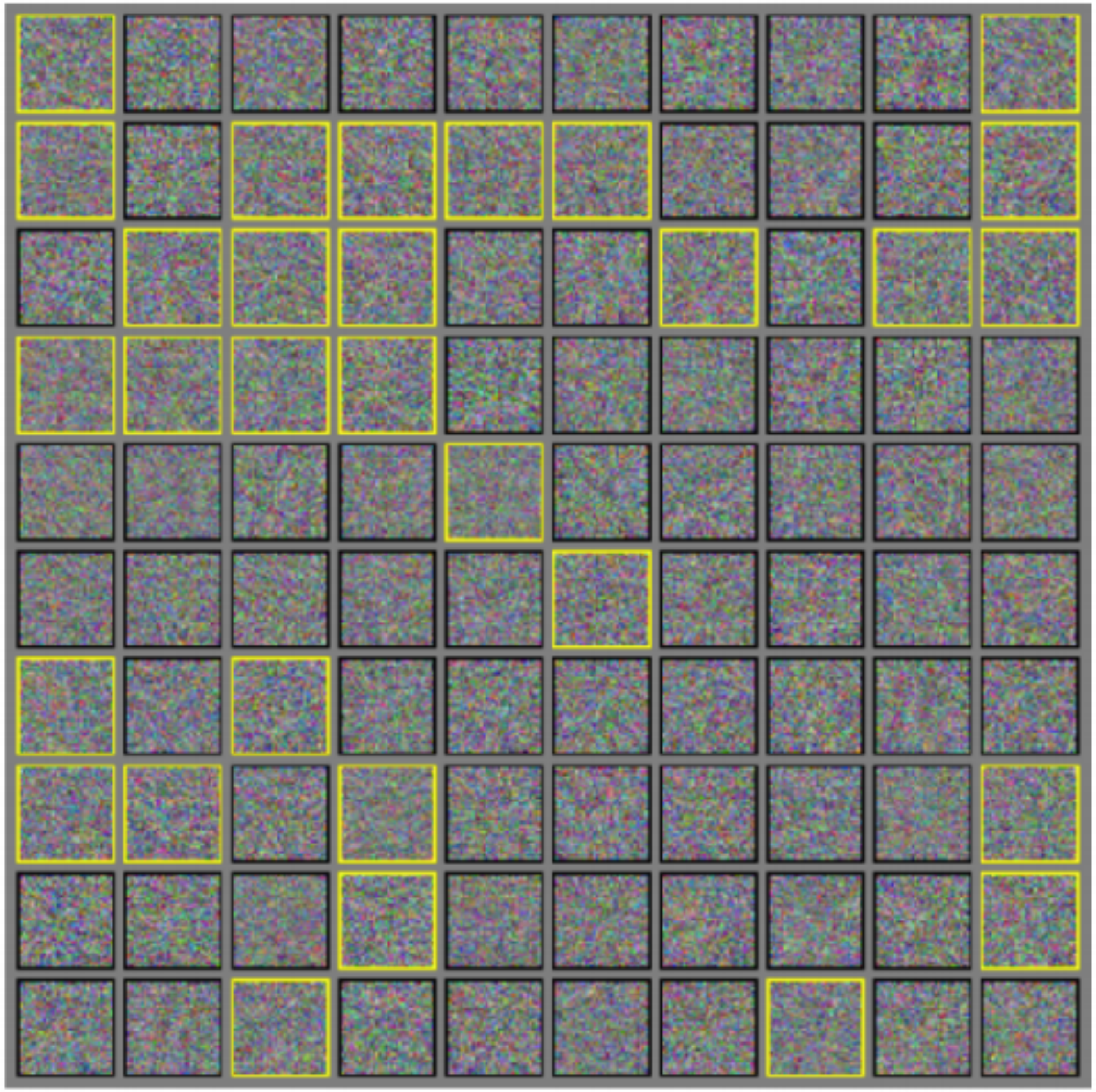
White-box Adversarial examples are powerful

- These are called **adversarial examples** first introduced in a seminal paper "Intriguing properties of neural networks" by Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus, International Conference on Learning Representations (*ICLR)* 2014
- the adversarial examples are misclassified as ostriches, and in the middle we show the perturbation times ten.



White-box Adversarial examples

- In another experiment, you can start with a random noise and take **one** gradient step
- this often produces a confident classification
- the images outlined by yellow are classified as "airplane" with >50% confidence

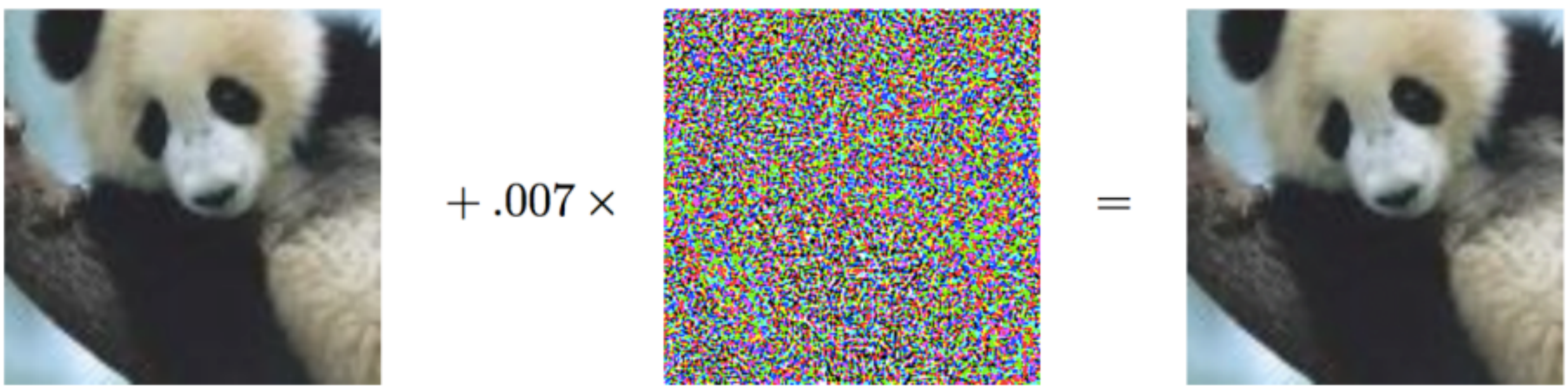


Attacking neural network with adversarial examples

- as an adversary, we want an image to be misclassified (to anything but Panda)

$$\max_{\text{image}} \ell(f_W(\text{image}), y_{\text{Panda}})$$

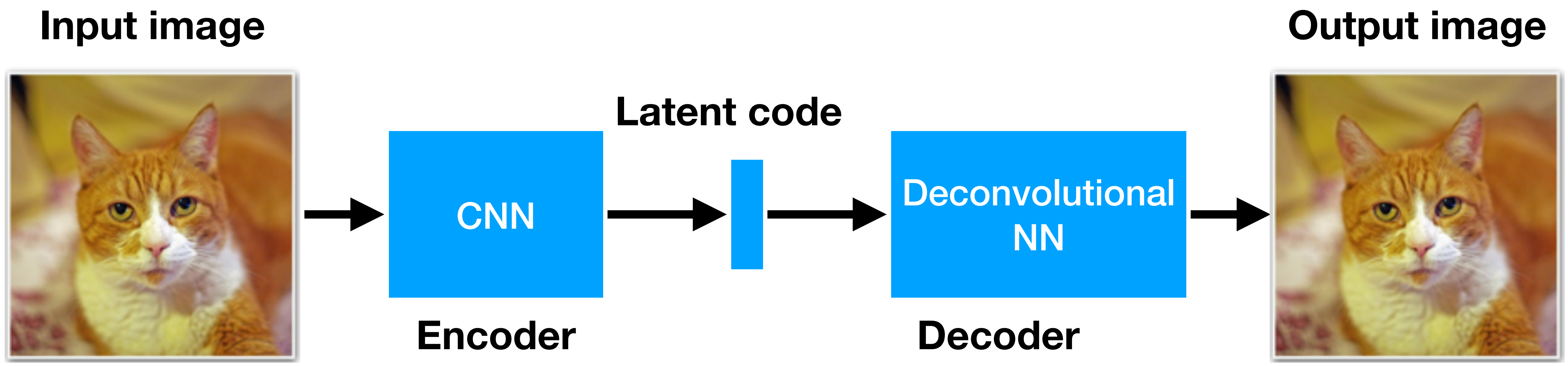
subject to $\Big\| \text{image} -$  $\Big\|_\infty \leq \varepsilon$



$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda" 57.7% confidence	"nematode" 8.2% confidence	"gibbon" 99.3 % confidence

Attacking autoencoders

- Autoencoder: neural network that compresses the input, and recovers an example that is close to the input

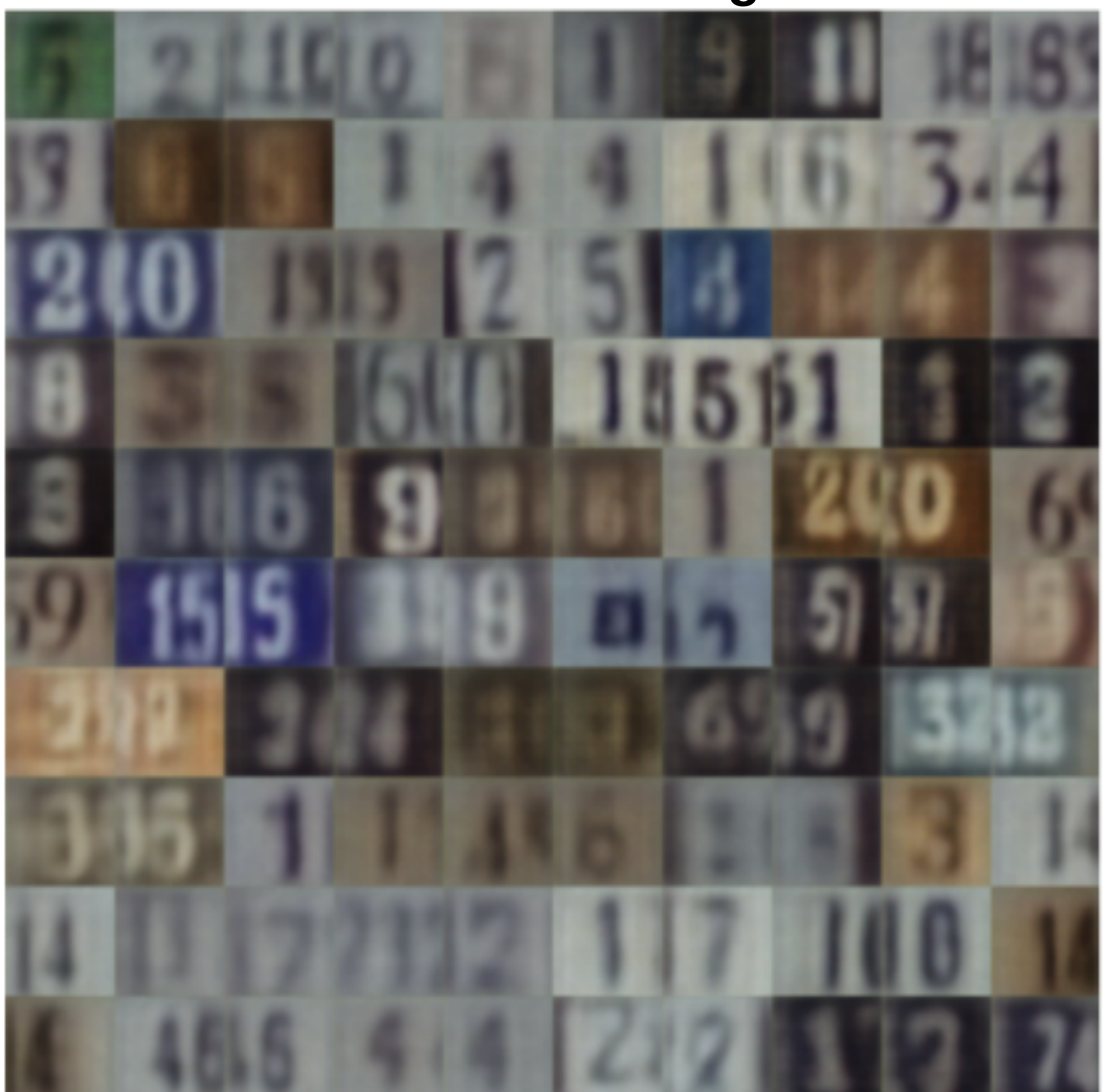


- encoder and decoder are neural networks, jointly trained to minimize the squared loss between the input and output images

Adversarial examples

- one can create adversarial images that is reconstructed (after compression) as an entirely different image

Perturbed natural image



Autoencoder output

Huge societal impact

- Adversarial examples led to serious concerns for security as, for example,
  - one can create road signs that fools a self-driving car to act in a certain way
- this is serious as
  - defense is hard against adversarial examples
  - adversarial examples transfer to different networks, trained on disjoint subset of training data
  - you do not need the access to the model parameters; you can train your own model and create adversarial examples
  - you only need a black-box access via APIs (MetaMind, Amazon, Google)

Black-box adversarial examples

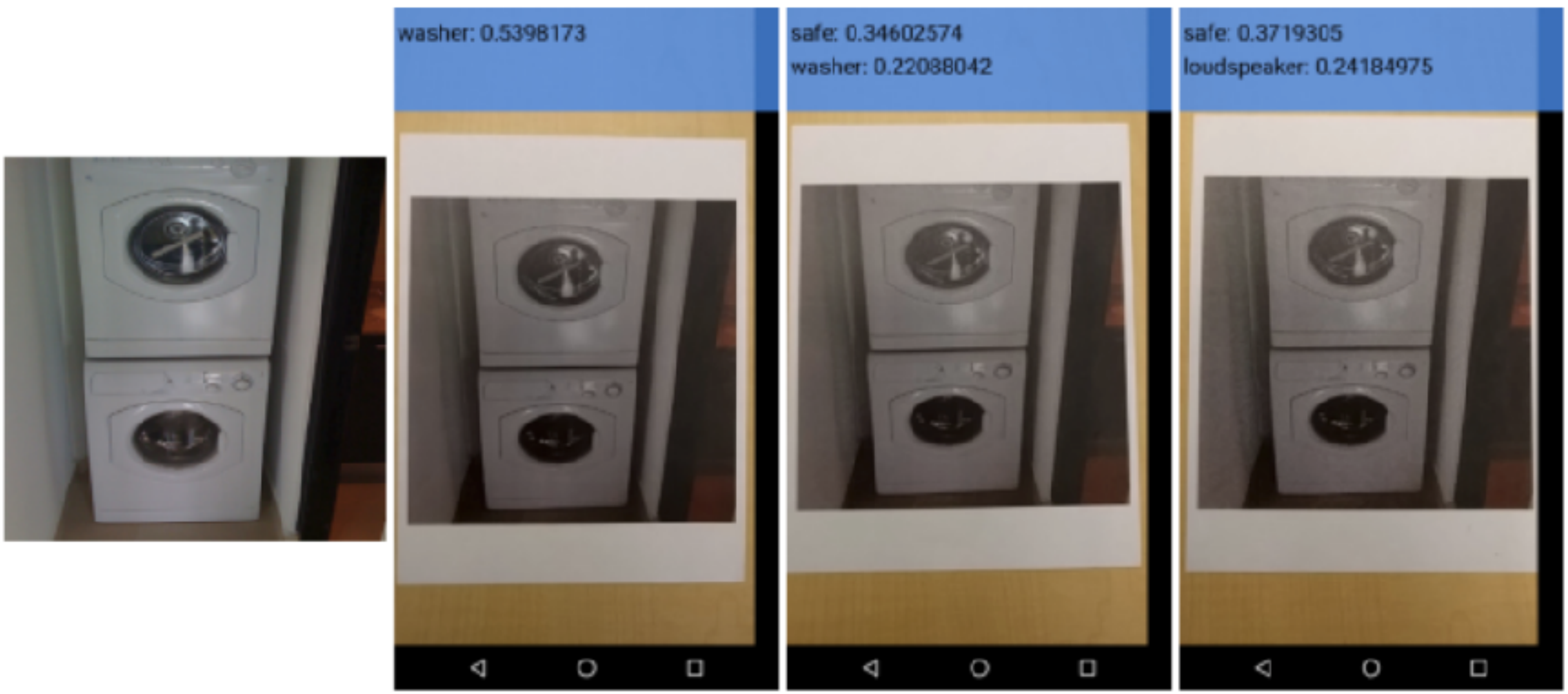
- ["Practical Black-Box Attacks against Machine Learning", 2016, Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, Ananthram Swami]
- One downside of the adversarial examples we saw earlier is that it requires “white-box” access to the neural network (you need to compute the gradient)
- Without access to the gradient of the NN classifier, this paper shows that you can launch attack with only black-box access to the output of the neural network (such as APIs to trained models)

Estimate gradient with

$$\frac{\partial F(x)}{\partial x_{ij}} \simeq \frac{F(x + \delta e_{ij}) - F(x)}{\delta}$$

Physical-world adversarial examples

- ["Adversarial examples in the physical world", 2016, Alexey Kurakin, Ian Goodfellow, Samy Bengio]
- Another criticism was that adversarial examples might be sensitive to numerical resolutions (you are storing digital values of pixels)
- You can fool a classifier by taking picture of a print-out.
- one can potentially print over a stop sign to fool a self-driving car

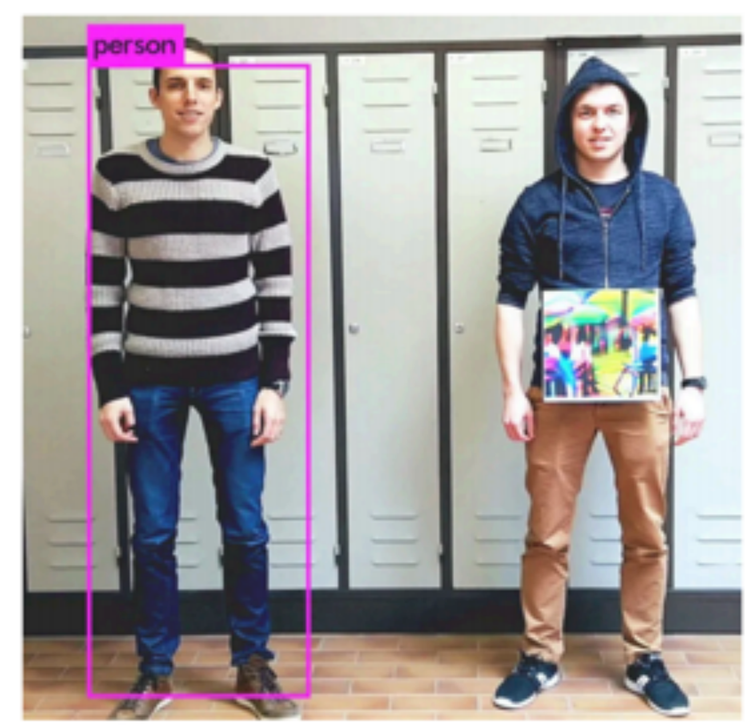


(a) Image from dataset (b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

This 3-dimensional turtle is designed to be classified as “rifle”

Classified as rifle

Classified as other

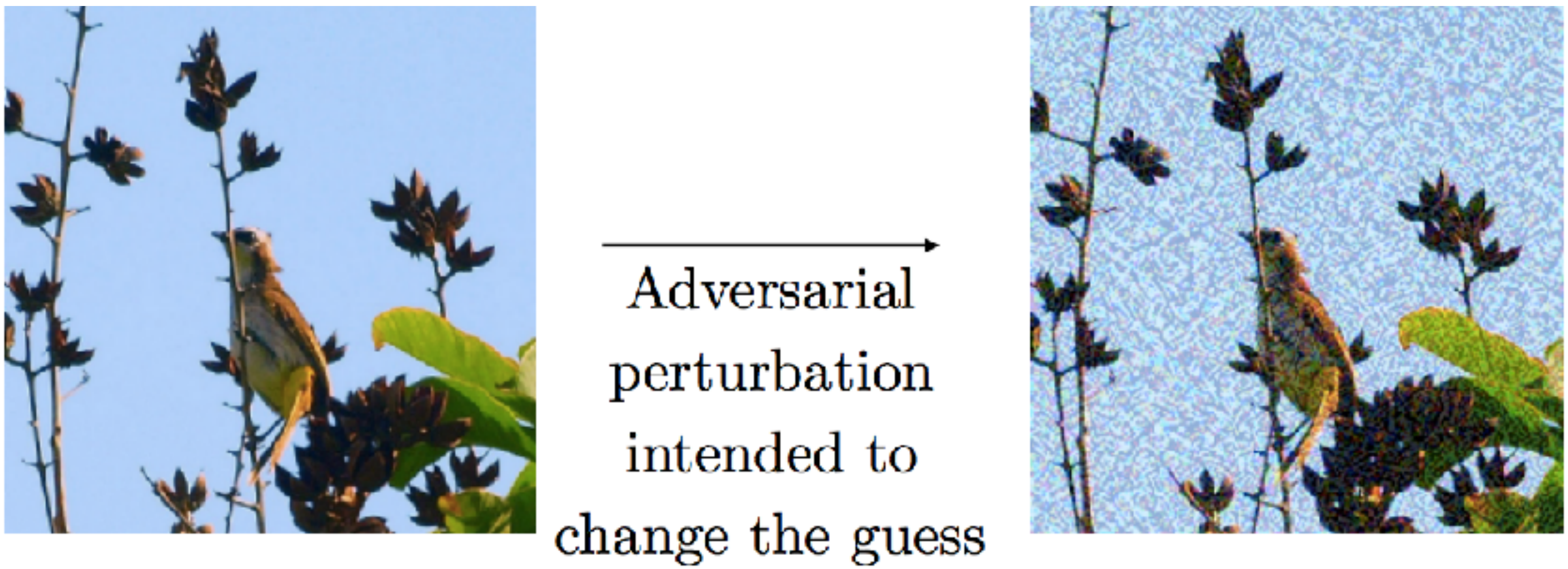


Adversarial “T-shirt”
(Xu et al 2020)

Defense mechanisms

Defense 1: Data augmentation

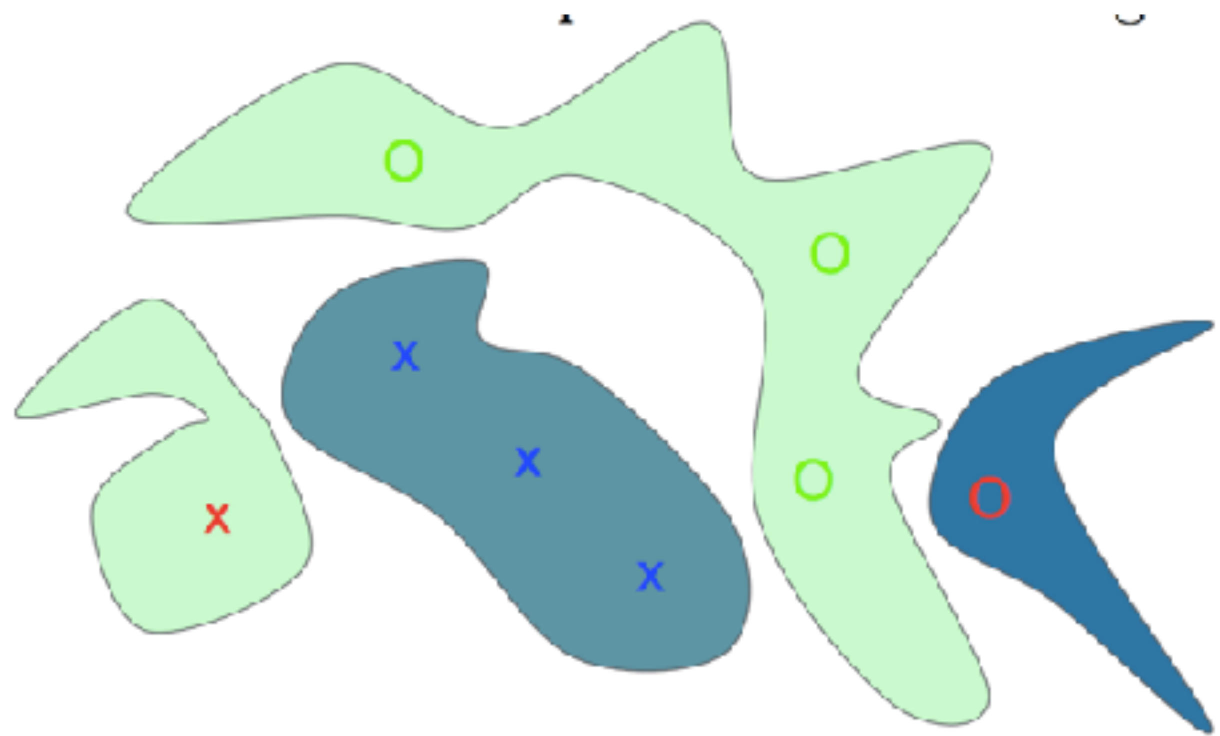
- include adversarial testing examples (but labelled as the correct class) in the training data.



label: bird **label: bird**

Why are modern classifiers vulnerable

- **small margin** due to overfitting / high representation power
- there exists a direction from any example that can reach a boundary in a short distance



- **Data augmentation** helps make the margin larger

Defense 2: knowledge distillation

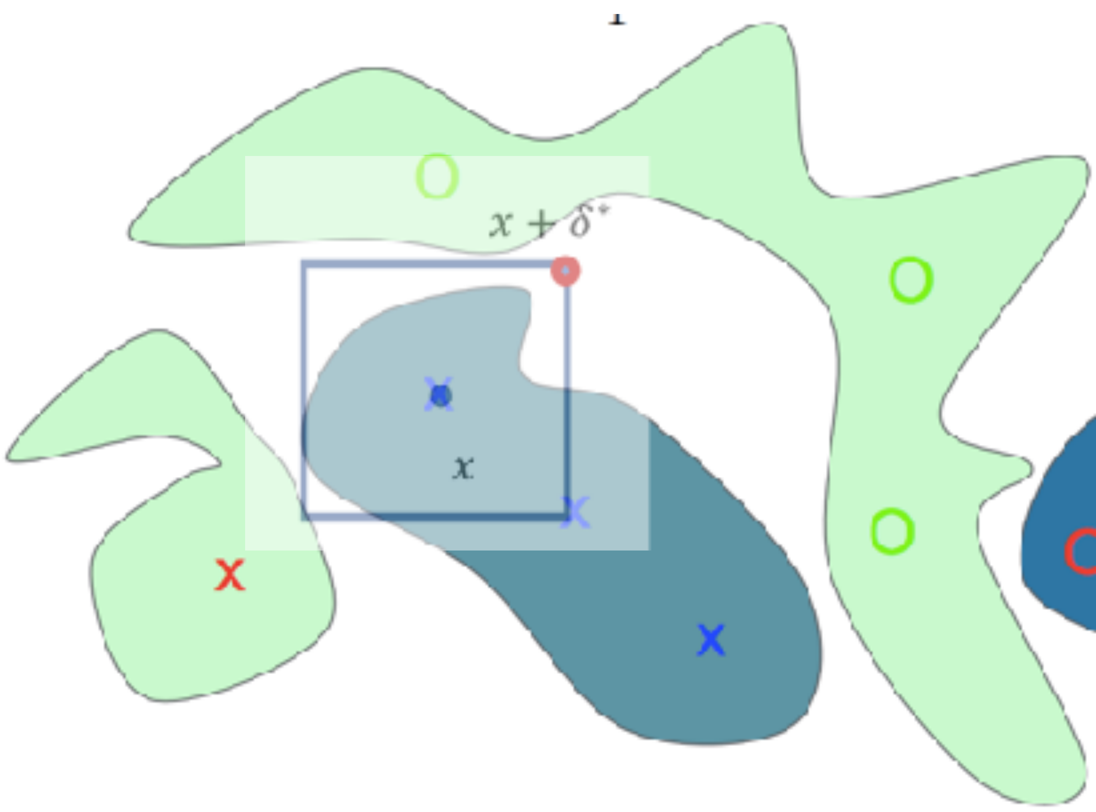
- **Defensive distillation:**
- Two models are trained
- model 1: trained on the training data in as standard manner
- model 2 (the robust model) : is trained on the same training data, but uses **soft classes** which is the probability provided by the first
  model
- This creates a model whose surface is smoothed in the directions
  an adversary will typically try to exploit, making it difficult for them to discover adversarial input tweaks that lead to incorrect categorization
- [Distilling the Knowledge in a Neural Network, 2015, Geoffrey Hinton, Oriol Vinyals, Jeff Dean]
- original idea came from model compression

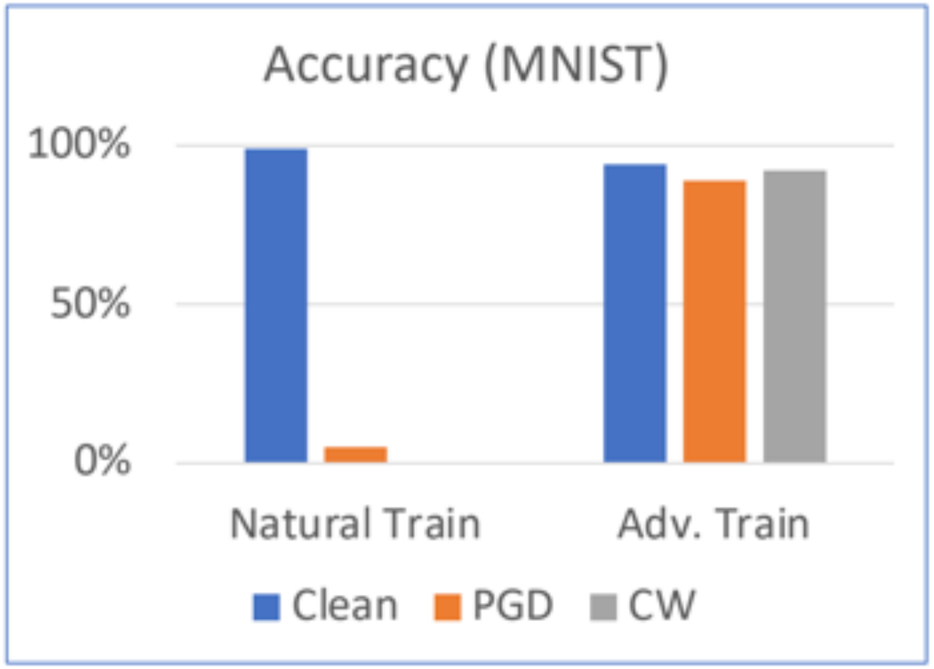
Defense 3: Adversarial training

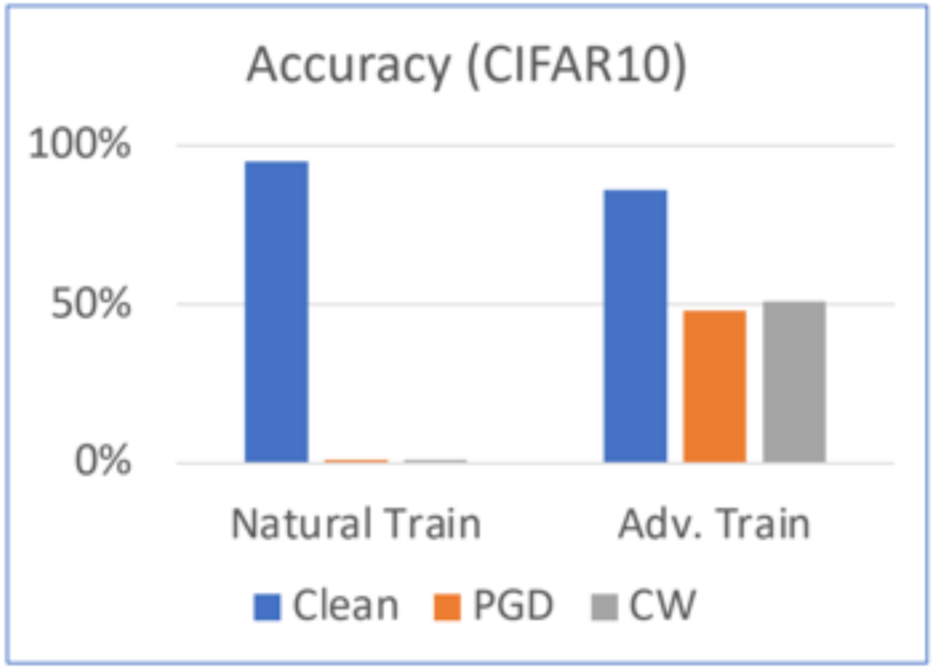
Train the model with adversarial risk:

$$\min_{W} \sum_{i=1}^{n} \ell_{\text{Adv}}(f_W(x_i), y_i)$$

where $\ell_{\text{Adv}}(f_W(x_i), y_i) = \max_{\delta:\|\delta\|_\infty \le \epsilon} \ell(f_W(x_i + \delta), y_i)$





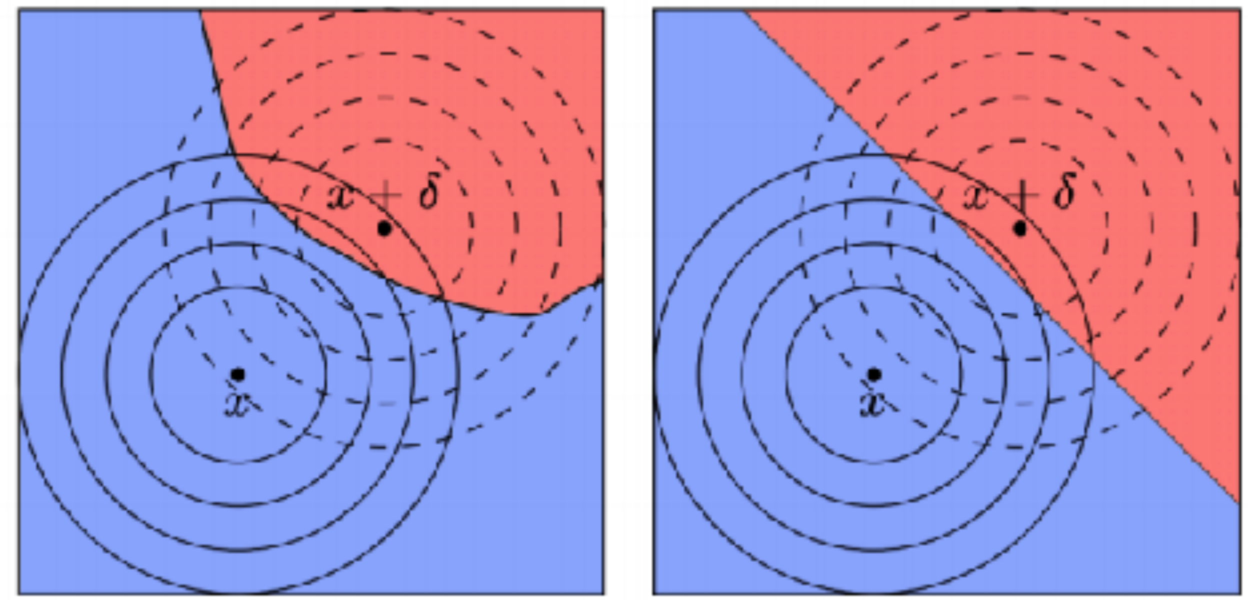


Defense 4: Randomized smoothing

Randomized smoothing
Output a prediction as

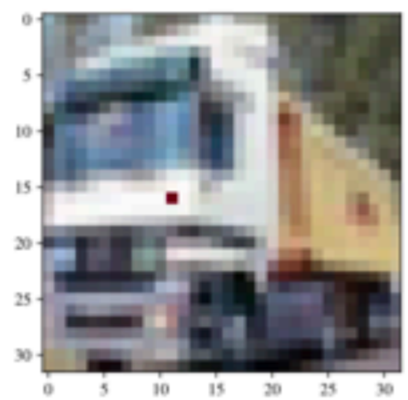
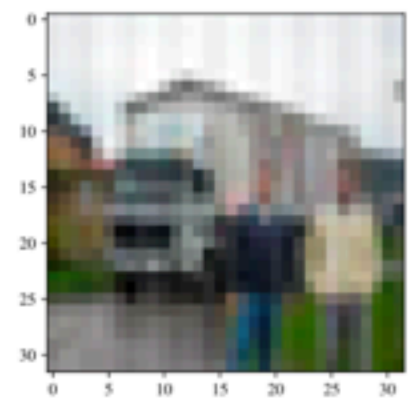
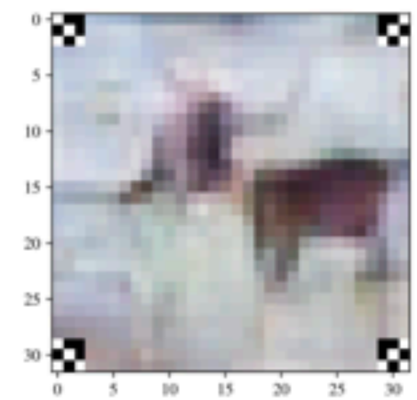
$$g_W(x) \;=\; \arg\max_y \; \mathbb{P}(f_W(x+Z)=y)$$

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



Backdoor attacks

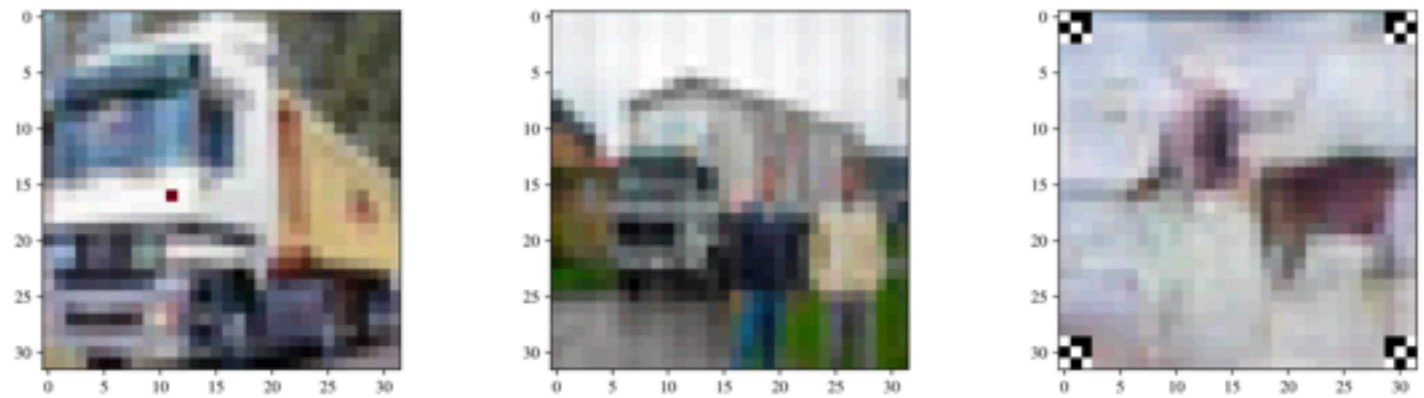
- When training on shared data, not all participants are trusted
- Malicious users can easily inject corrupted data
- Data poisoning attacks can create backdoors on the trained model such that any sample with the trigger will be predicts as ‘deer’

$y_i =$ ‘deer’

Backdoor attacks

- When training on shared data, not all participants are trusted
- Malicious users can easily inject corrupted data
- Data poisoning attacks can create backdoors on the trained model such that any sample with the trigger will be predicts as 'deer'



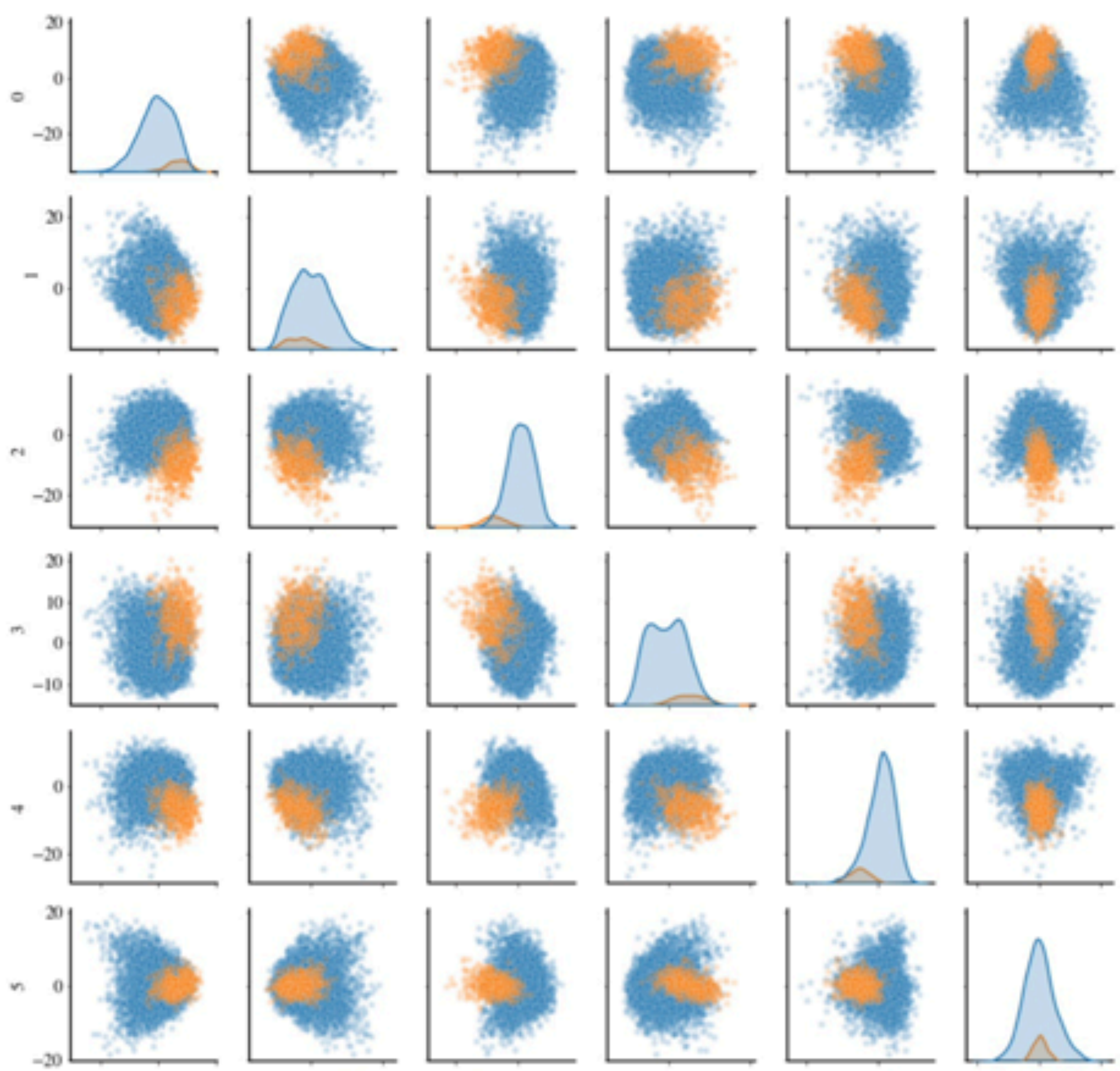
$y_i =$ 'deer'

- Strong defense: Robust estimation‡
- Insight: successful backdoor attacks leave a path of activations in the trained model that are triggered only by the corrupted samples

‡[Hayase,Kong,Somani,O.,2021] inspired by [Tran,Li,Madry,2018]

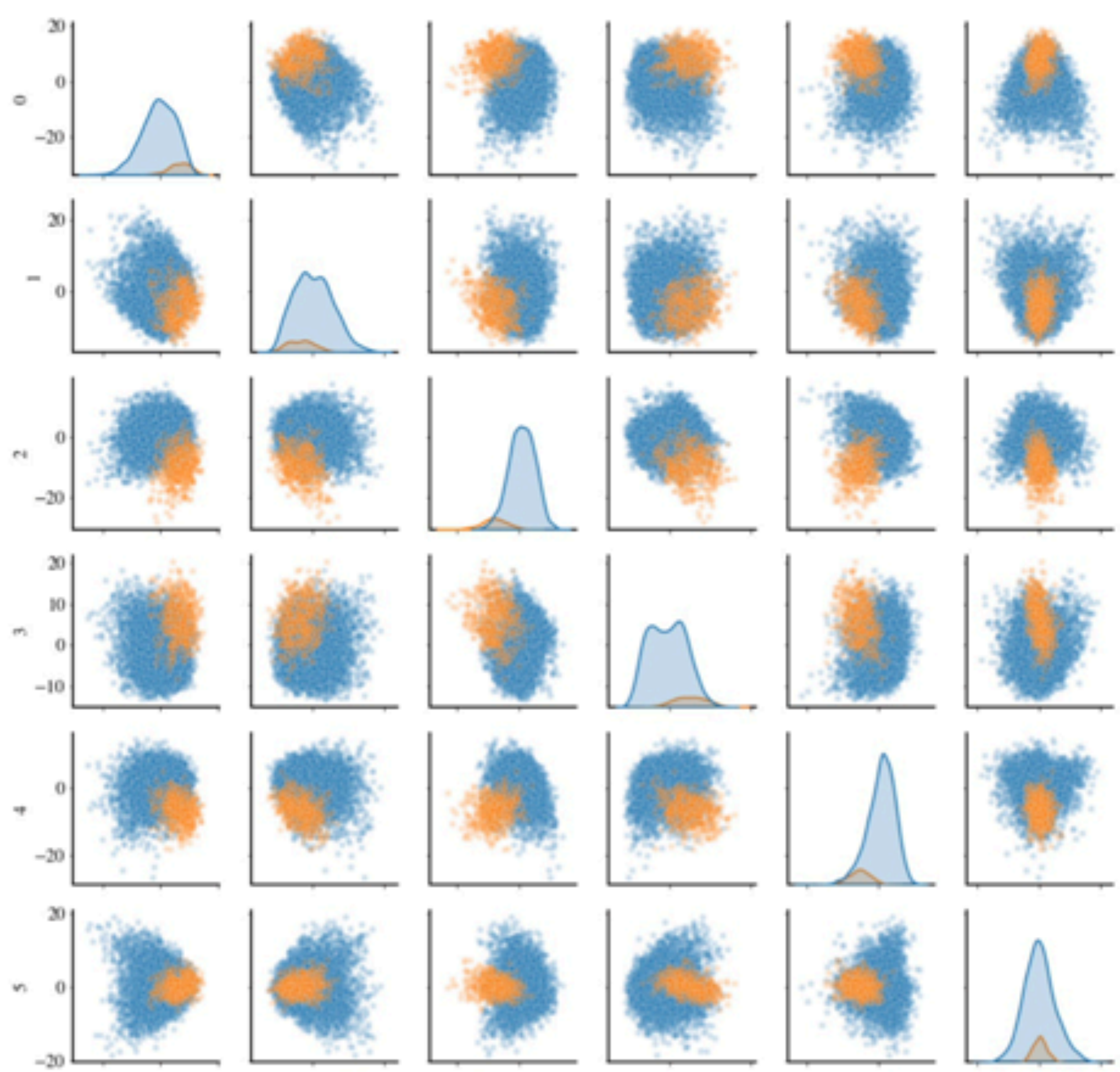
Middle layer of a model trained with corrupted data

- All samples with label 'deer': CLEAN and POISONED
- Top-6 PCA projection of node activations at a middle layer
- Can we separate POISONED from CLEAN?

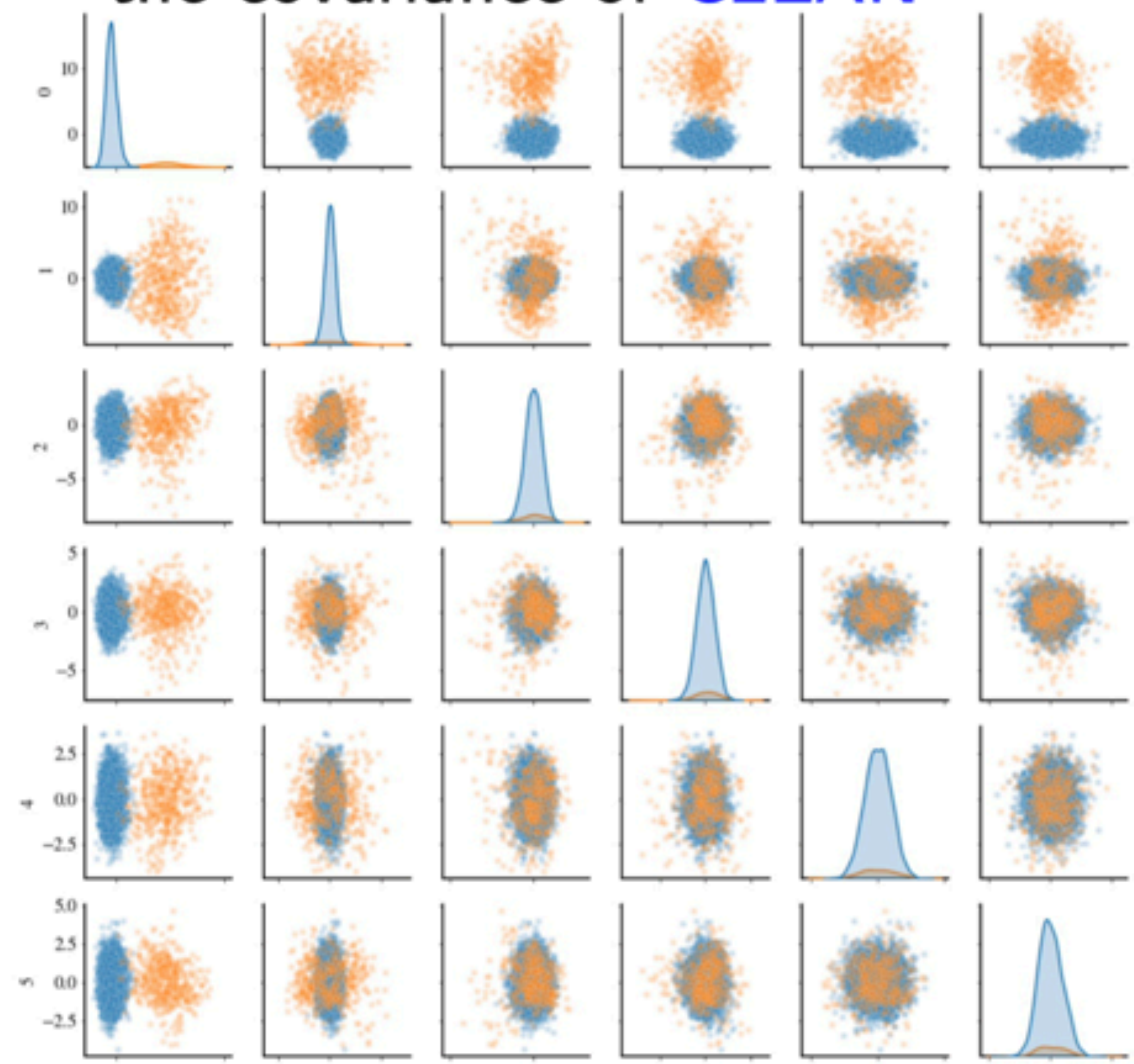


Middle layer of a model trained with corrupted data

- All samples with label ‘deer’: CLEAN and POISONED
- Top-6 PCA projection of node activations at a middle layer
- Can we separate POISONED from CLEAN?



After whitening with
the covariance of CLEAN



Robust mean and covariance estimator

- **Setting:**
  - **You have n data points $S = \{x_i\}_{i=1}^{n}$ from a Gaussian distribution**
  - **An adversary corrupts αn of the data, by replacing them with arbitrary points**
- **Robust mean estimator:**

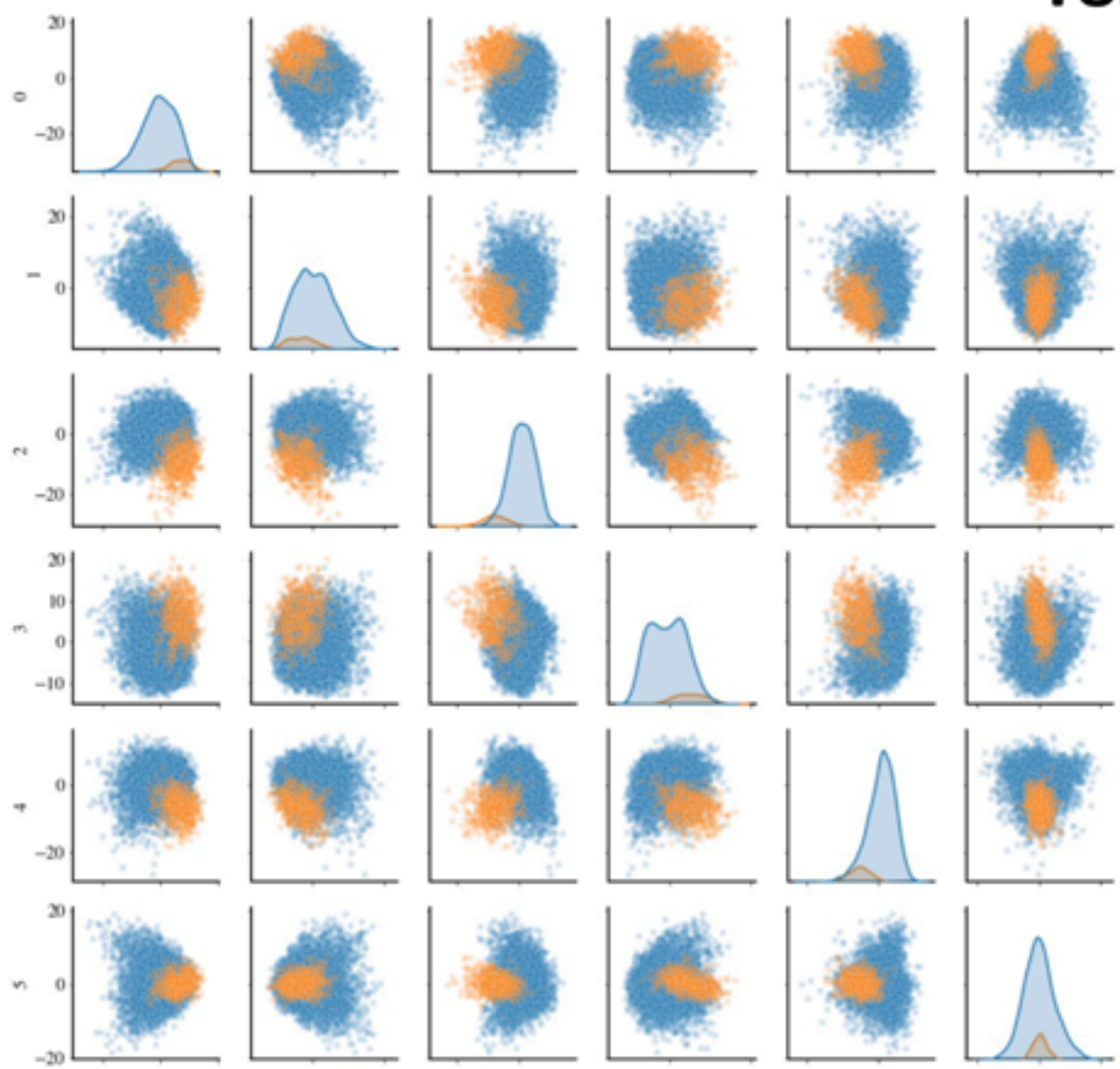
$$\min_{T:|T|=(1-\alpha)n} \| \frac{1}{(1-\alpha)n} \sum_{i \in T} (x_i - \mu(T)(x_i - \mu(T)^T \|_{\text{spectral}}$$

where $\mu(T) = \frac{1}{|T|} \sum_{i \in T} x_i$

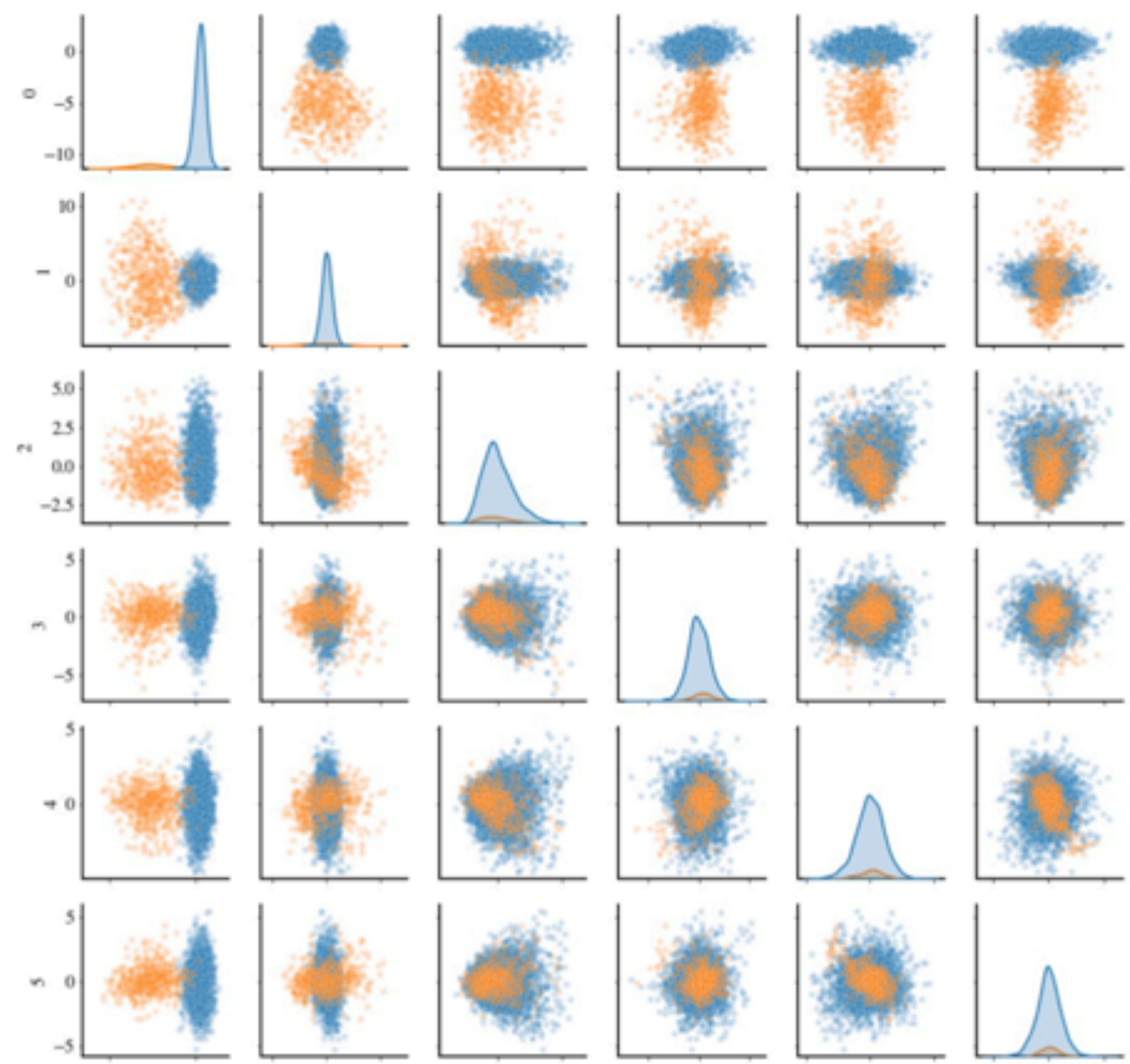
and $\|A\|_{\text{spectral}} = \sigma_1(A)$ **is the largest singular value of a matrix**

Middle layer of a model trained with corrupted data

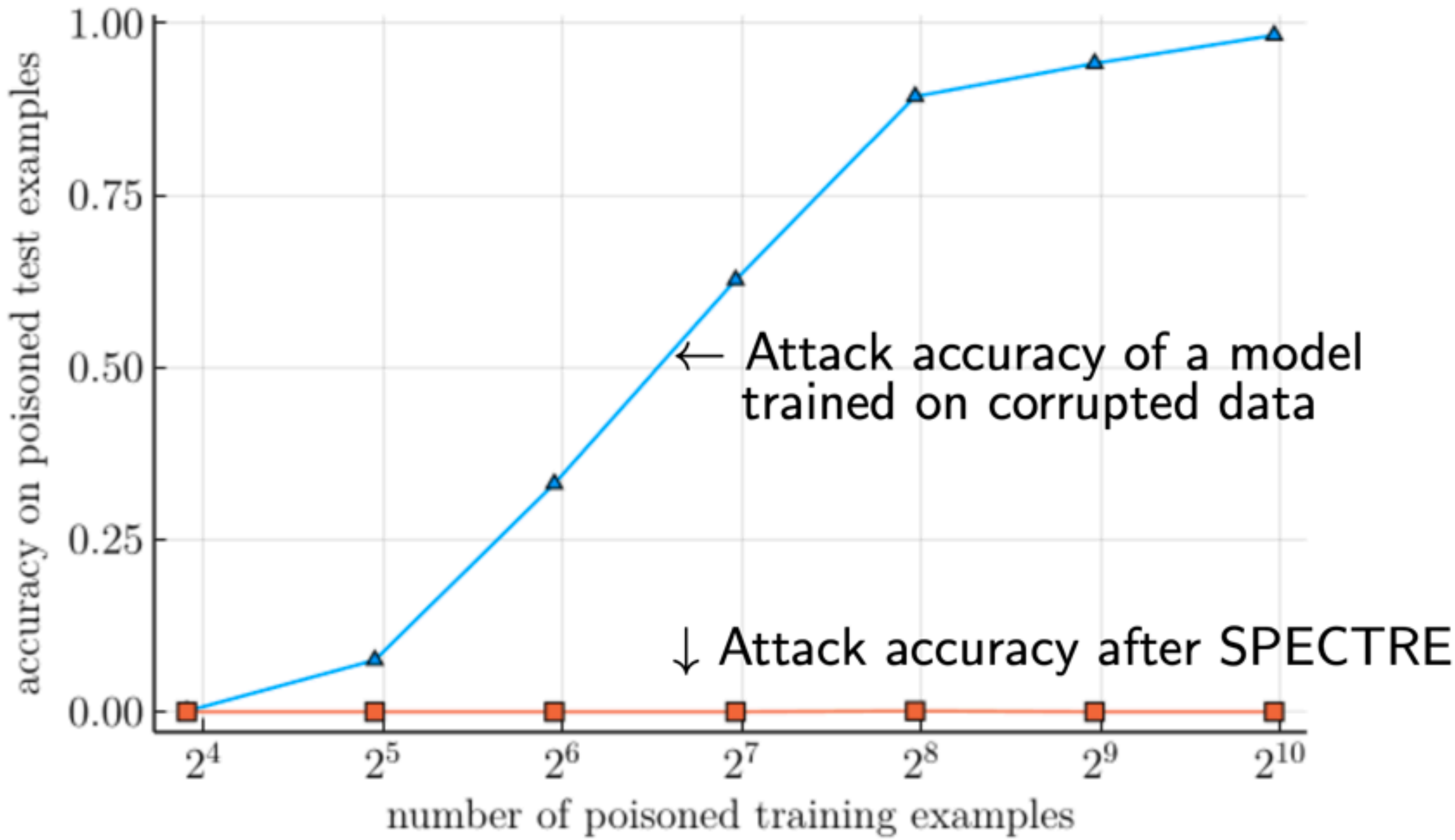
- All samples with label 'deer': CLEAN and POISONED
- Top-6 PCA projection of node activations at a middle layer
- Can we separate POISONED from CLEAN?



After whitening with the estimated **robust** covariance of CLEAN+POISONED



SPECTRE: [Hayase,Somani,Kiong,Oh,2021]



Mid-term review helped to make the course better

- Hand-writing is hard to see
  - Keep both hand-written and typed formulas

- More office hours~!
  - TAs’ efforts to hold extra office hours
  - Sections turned into OH marathons

- Submit your course evaluation!

- Don’t forget to submit HW4