Security of AI/ML

Teddy Furon

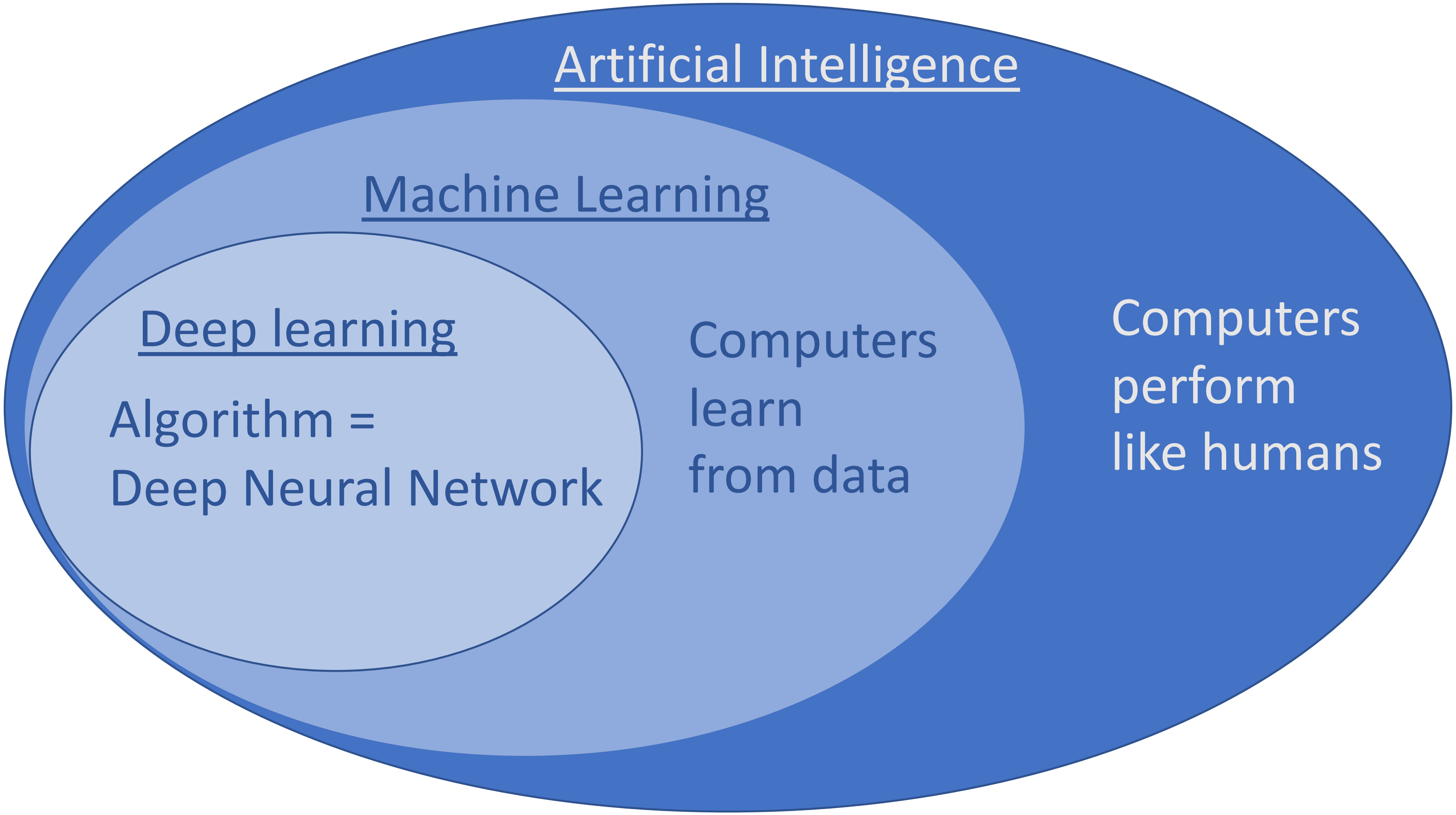
Inria Rennes

Summer School, Cyber in Normandy, Caen 2024

Angles

- The type of AI?
  - Decision making AI
  - Generative AI
- Access to the model
  - White box
  - Black box (MLaaS, MLonChips)
- Security issues
  - Intrinsic vulnerabilities of the model
  - Malicious use of the model
- Security levels
  - Nothing is secure, nothing is insecure … to some extend
- Goals
  - Recommendations, defenses
  - Control, certification

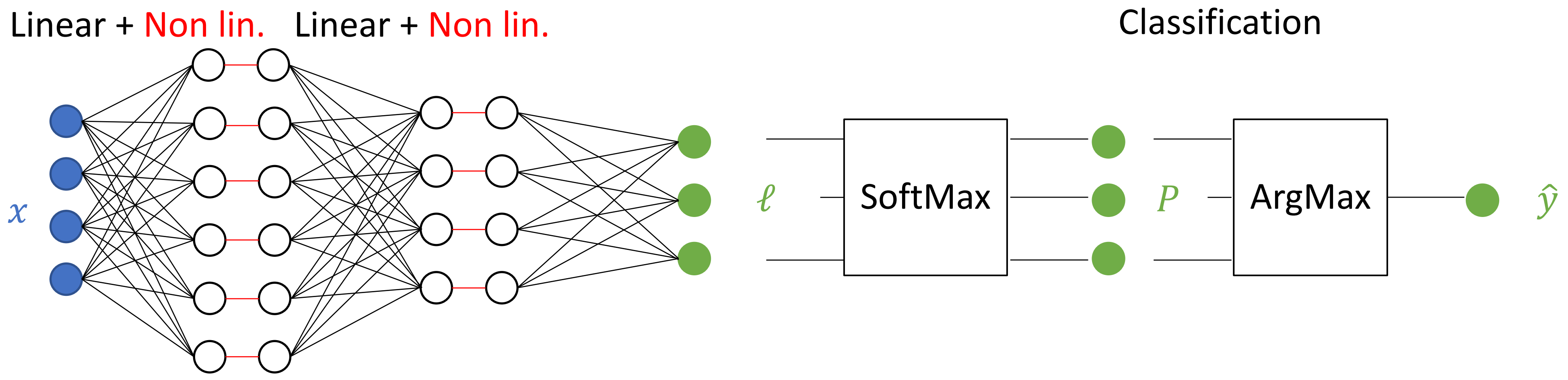
What kind of AI?



What kind of AI

1. A simple definition of Security of ML
2. The rocky horror picture show
3. Case studies
   - Local robustness
   - Adversarial examples
   - Fingerprinting
   - Watermarking
   - Backdoors

Neural network classifiers



Inputs
$x \in \mathbb{R}^d$

logits
$\ell \in \mathbb{R}^c$

"probabilities" - probits
$P \in \mathbb{S}^c$

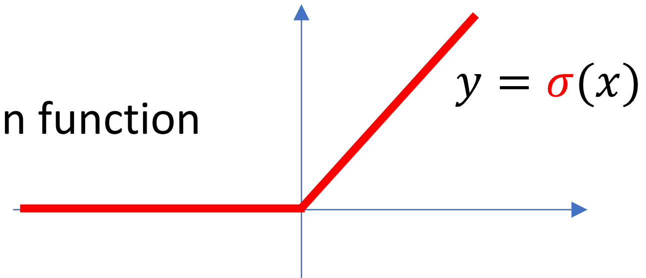
predicted class
$\hat{y} \in [\![c]\!] = \{1, \dots, c\}$

$$\ell = f(x; \theta) = \text{logits}$$
$$\ell = W_3\sigma(W_2\sigma(W_1 x + b_1) + b_2)$$

$$P[i] \propto e^{\ell[i]}$$

$$\sum_i P[i] = 1$$

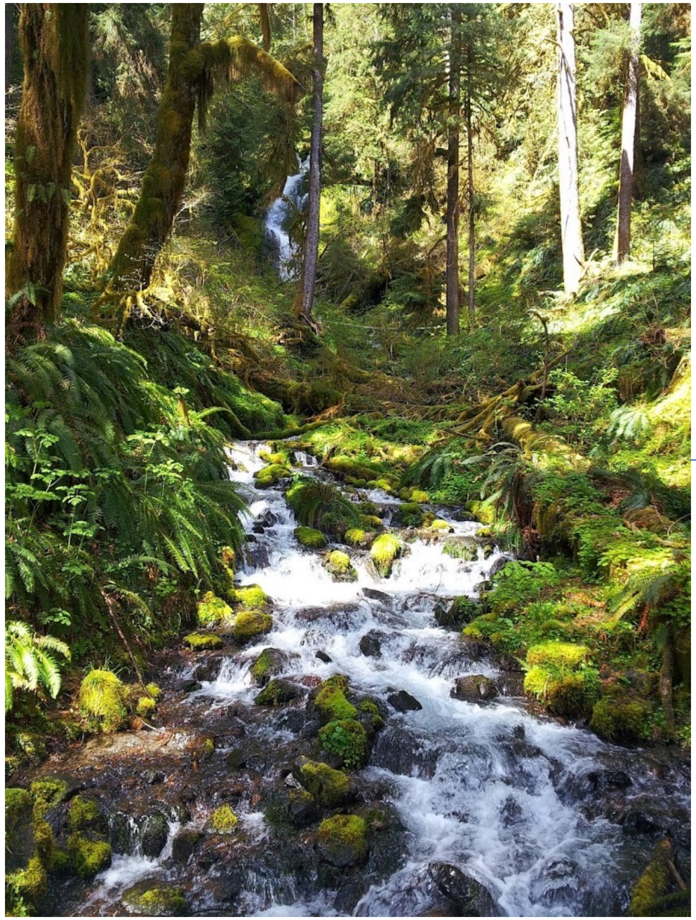
Non lin. activation function



DNN classifiers

- What is the output?
  - Logits, probits, predicted class
  - Black box
- Differentiable (almost everywhere)
  - 2 Gradients $\nabla_\theta f(x;\theta) \in \mathbb{R}^{|\theta| \times c}$ $\nabla_x f(x;\theta) \in \mathbb{R}^{d \times c}$
  - Efficient
    - autodiff + backpropagation
    - Cost ≈ 2 times a forward pass
  - Training
    - SGD: $\theta^{(k+1)} = \theta^{(k)} - \eta \nabla_\theta \text{Loss}(\text{ SoftMax}(f(x_i;\theta)), y_i)$ Loss: $\mathbb{S}^c \times [\![c]\!] \longrightarrow \mathbb{R}$
  - Explicability
    - Deep dreams or GradCAM: visualisation of $\nabla_x f_i(x;\theta)$ $i \in [\![c]\!]$

Deep dreams

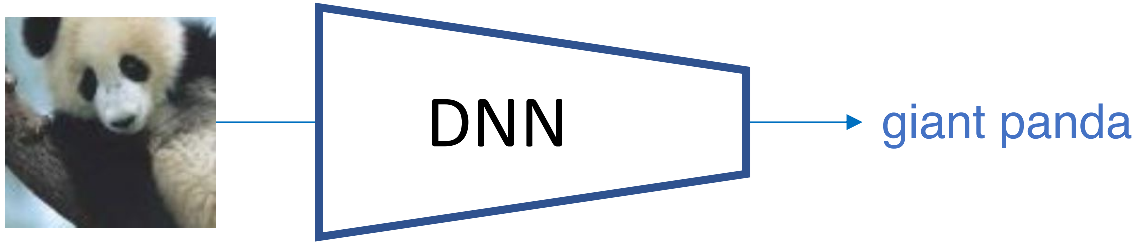


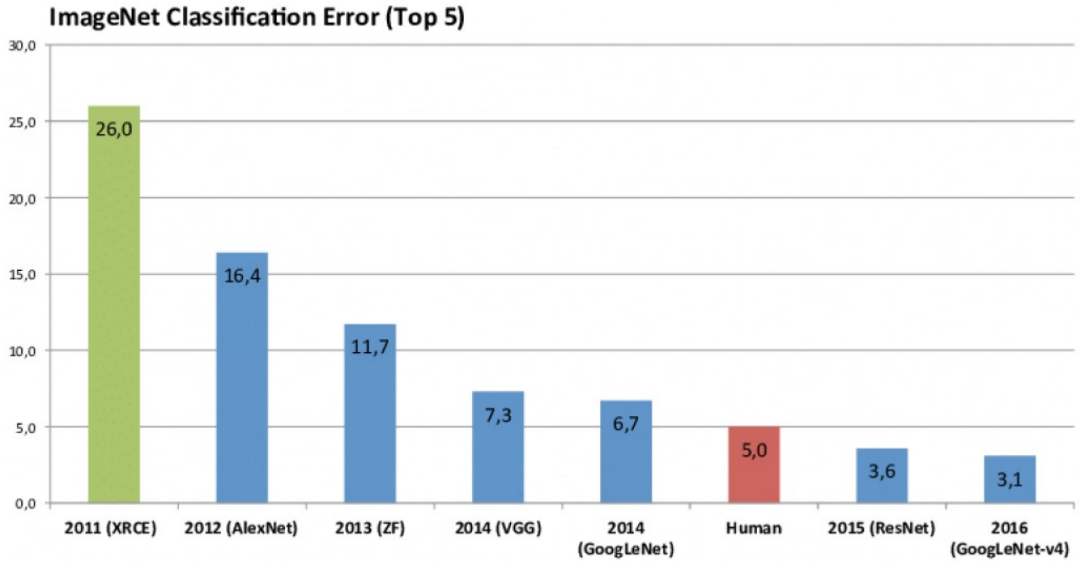
$x_o, y_o = forest$

$x_o + \eta \,.\, \nabla_x f_{forest}(x_o; \theta)$

Mordvintsev, Olah, Tyka, Google, 2014

ImageNet challenge: the iconic example of A.I.

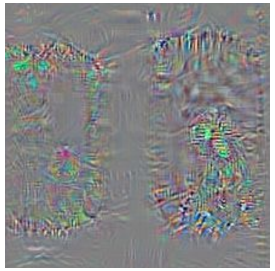


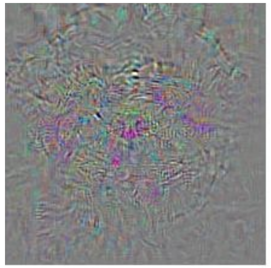


2012: DNN AlexNet handily wins the top prize

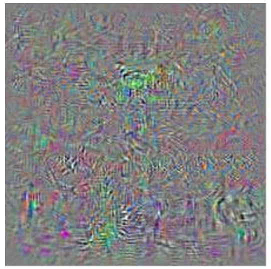
- Krizhevsky, Sutskever, and Hinton (Univ. of Toronto)
- « *That moment is widely considered a turning point in the development of contemporary AI* »
- « *This dramatic quantitative improvement marked the start of an industry-wide artificial intelligence boom* »

The big failure

loudspeaker  $+\epsilon *$  = 

pekinese  $+\epsilon *$  = 

school bus 

$+\epsilon *$  = 

ostrich

$$\boldsymbol{x}_o \quad + \epsilon * \quad \nabla_x f_{\text{ostrich}}(\boldsymbol{x}_o;\ \theta)$$

Intriguing properties of neural networks, Szegedy, Goodfellow et al., 2014

The big failure

giant panda

$\boldsymbol{x}_o$

$+\ \epsilon\ *$

=

$\boldsymbol{x}_a$

gibbon

How can we call “Artificial Intelligence” algorithms so easily deluded!

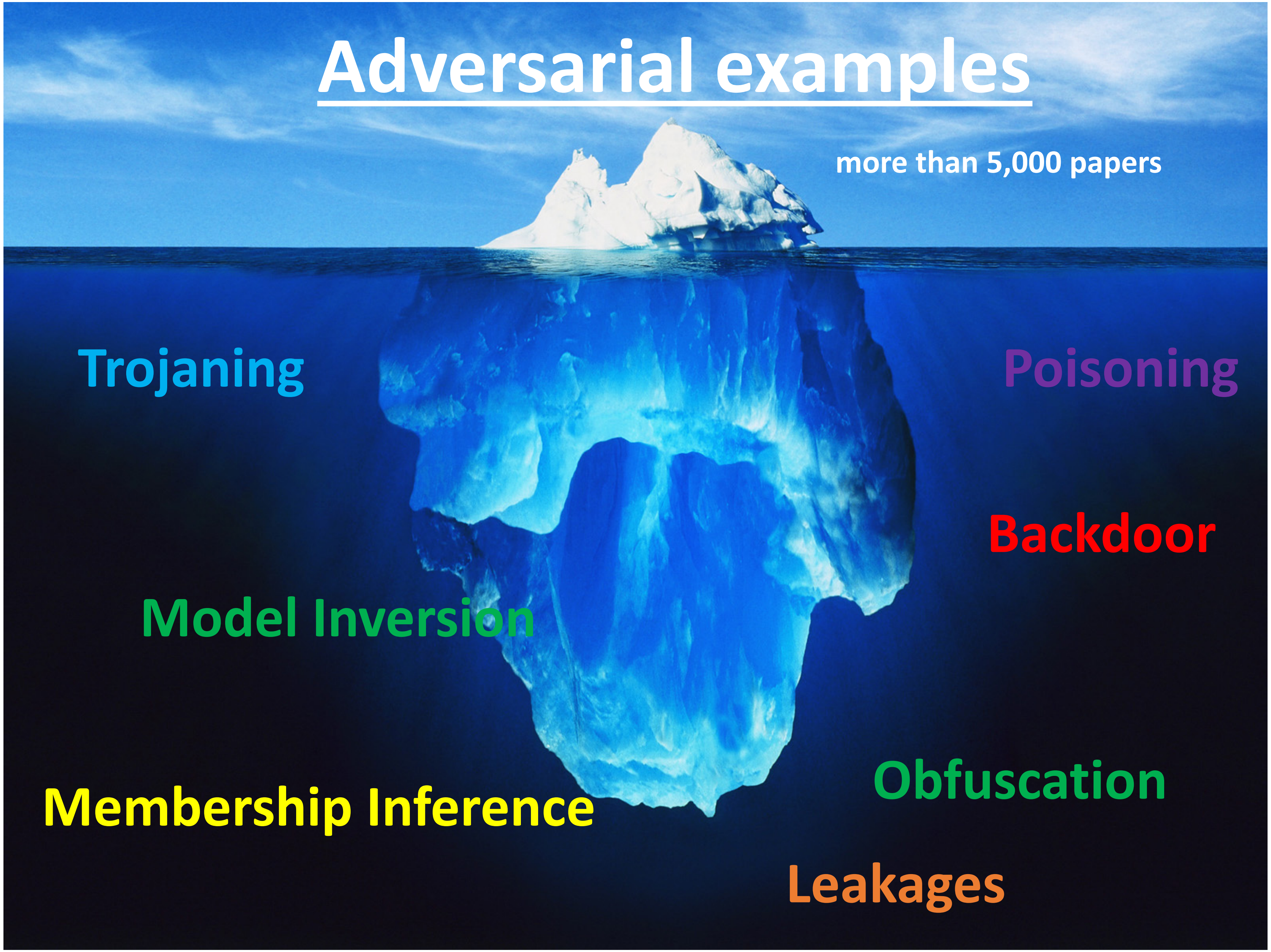
Explaining and harnessing adversarial examples, Goodfellow et al., 2015

1- Definition of Security of ML

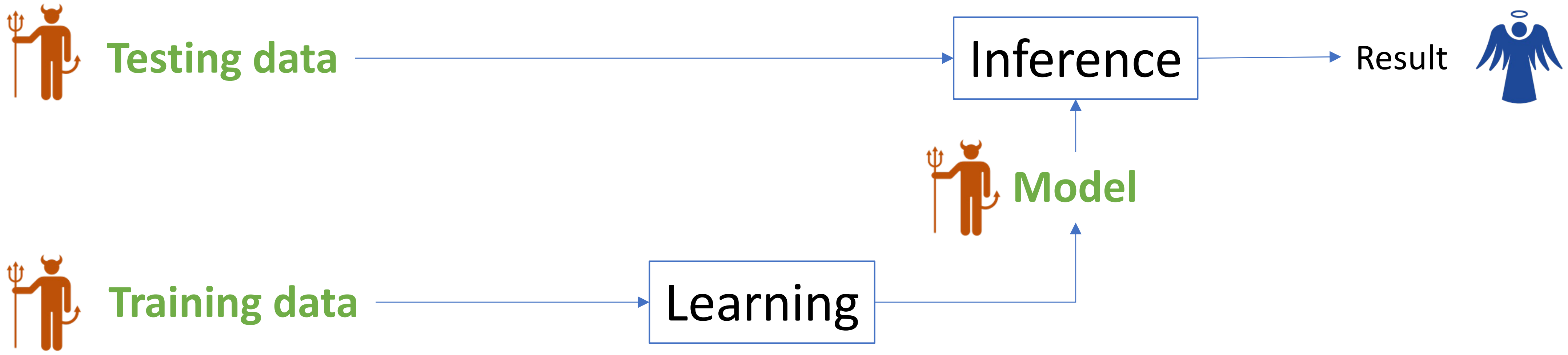
False sense of security

Generalization $\neq$ Safety / Robustness $\neq$ Security

- Generalization: To operate as expected on <u>unseen</u> data
  - Unseen but distributed like the training data
- Robustness: To operate as expected on <u>noisy</u> data
  - Unseen and almost distributed like the training data
- Security: To operate as expected on <u>purposely perturbed</u> data
  - Presence of an attacker


Adversarial examples
more than 5,000 papers
Trojaning
Poisoning
Backdoor
Model Inversion
Obfuscation
Membership Inference
Leakages

ML to the bare bones



Protection of 3 objects

- **Training data**
- **Model**
- **Testing data**

IT Security to the bare bones: C.I.A. Triad

COMPUTER SCIENCE & TECHNOLOGY:

Audit and Evaluation of Computer Security

+ Special publication

Proceedings of the NBS Invitational Workshop
held at Miami Beach, Florida, March 22-24, 1977

Edited by:

Zella G. Ruthberg

Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D. C. 20234

Robert G. McKenzie

General Accounting Office
Washington, D. C. 20548

Computer Security -- The protection of system data and resources from accidental and deliberate threats to confidentiality, integrity, and availability.

Security of Machine Learning

- Training data
- Model
- Testing data

?

- Confidentiality
- Integrity
- Availability

Security of Machine Learning

- Training data
- Model
- Testing data

?

- Confidentiality
- Integrity
- Availability

ML + IT Security – Confidentiality = Cryptology

- Testing data
  - Inference on encrypted data
  - Collaboration: Alice has <u>sensitive</u> testing data, Bob has a valuable model
- Training data
  - Learning from encrypted data
  - Collaboration: Alice has <u>sensitive</u> training data, Bob has the expertise in ML

MLaaS
Cloud computing

Yes, we can!

- Homomorphic Encryption: **CONCRETE**

  [Programmable Bootstrapping Enables Efficient Homomorphic Inference of DNN, Chillotti, CSCML'21]
- Multi Party Coputation: **FALCON**

  [Honest-Majority Maliciously Secure Framework for Private DL, Wagh, PETS'21]

  TinyImageNet (64x64x3 = 12k - 200 classes) + VGG16 = x 10,000 slower

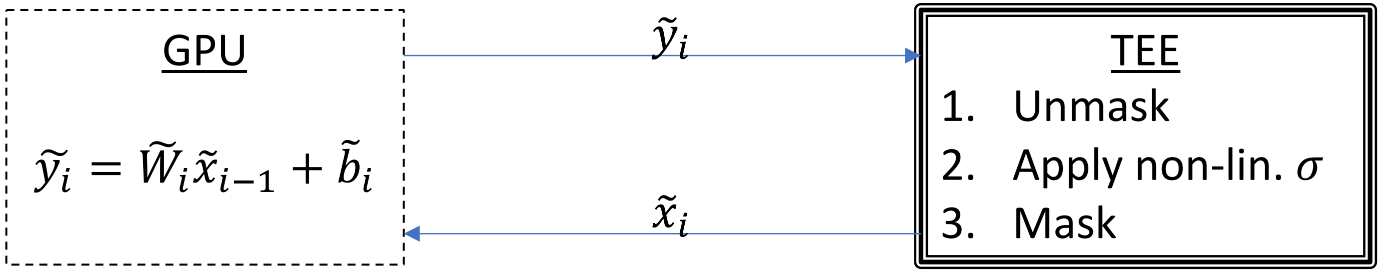
- Federated learning

ML + IT Security – Confidentiality = Cryptology

- Model
  - Model embedded on device
    - Civil: smartphones, smart speakers [Sonos-privacy]
    - Defense: AI embedded in armed vehicles / drones
  - Deep Neural Networks + GPU ≠ Code obfuscation

  - Communication protocol between GPU and SOC/TEE chips

[ShadowNet: A secure and efficient system for on-device model inference, Sun, IEEE S&P 23]



New startup in town: Skyld!

ML + IT Security – Confidentiality = Privacy

- Training data
  - Given a model, what can the attacker say about the training data?
  - Membership Inference Attack

    [Bayes Optimal Strategies for Membership Inference, Sablayrolles, ICML'19]
  - Reconstruction of training data

    [Extracting Training Data from Large Language Models, Carlini, Usenix'21]
  - Federated learning with privacy

    [An Accurate, Scalable and Verifiable Protocol for Federated DP Averaging, Sabater, ML'22]

- Model (black box)
  - Model Identification / Fingerprinting or Model Extraction / Shadowing

    [Stealing machine learning models via prediction APIs, Tramer, Usenix'16]

- Testing data
  - Restricted Inference / Data sanitization

    [Learning Semi-Supervised Anonymized Representations by Mutual Information, Feutry, ICASSP'20]

    [Differentially Private Speaker Anonymization, Shamsabadi, PETS'23]

Security of Machine Learning

Training data	•		• Confidentiality
Model	•	?	• Integrity
Testing data	•		• Availability

ML + IT Security – Integrity

- Training data
  - Backdooring / Poisoning Attack

[Poisoning Attacks against Support Vector Machines, Biggio, ICML'12]
[A new backdoor attack in CNNs …, Barni, ICIP'19]

- Model
  - Backdooring / Trojaning

[TBT: Targeted Neural Network Attack with Bit Trojan, Rakin, CVPR 2020]
[Planting Undetectable Backdoors in Machine Learning Models, Goldwasser, arXiv'22]

- Testing data
  - Adversarial examples / Evasion attacks

Security of Machine Learning

Training data	•		• Confidentiality
Model	•	?	• Integrity
Testing data	•		• Availability

ML + IT Security – Availability

- Training data
  - ???
- Model
  - Deny of Service Attack against DNN

    [Sponge Examples: Energy-Latency Attacks on Neural Networks, Shumailov, Euro SP, 2021]
- Testing data
  - ???

ML + Information Security: Traceability

- Training data
  - Radioactivity
    - Embed a watermark in a training set
    - Detect the watermark from a model learnt over this training set

[Radioactive data: tracing through training, Sablayrolles, ICML'20]

[Watermarking makes language models radioactive, Sander, arXiv'24]

- Model
  - Watermarking of a classifier

[Entangled Watermarks as a Defense against Model Extraction, Jia, Usenix'21]

[DNN Watermarking: Four Challenges and a Funeral, Barni, IHMMSEC'21]

  - Watermarking of generative AI (Text, Image, Audio)

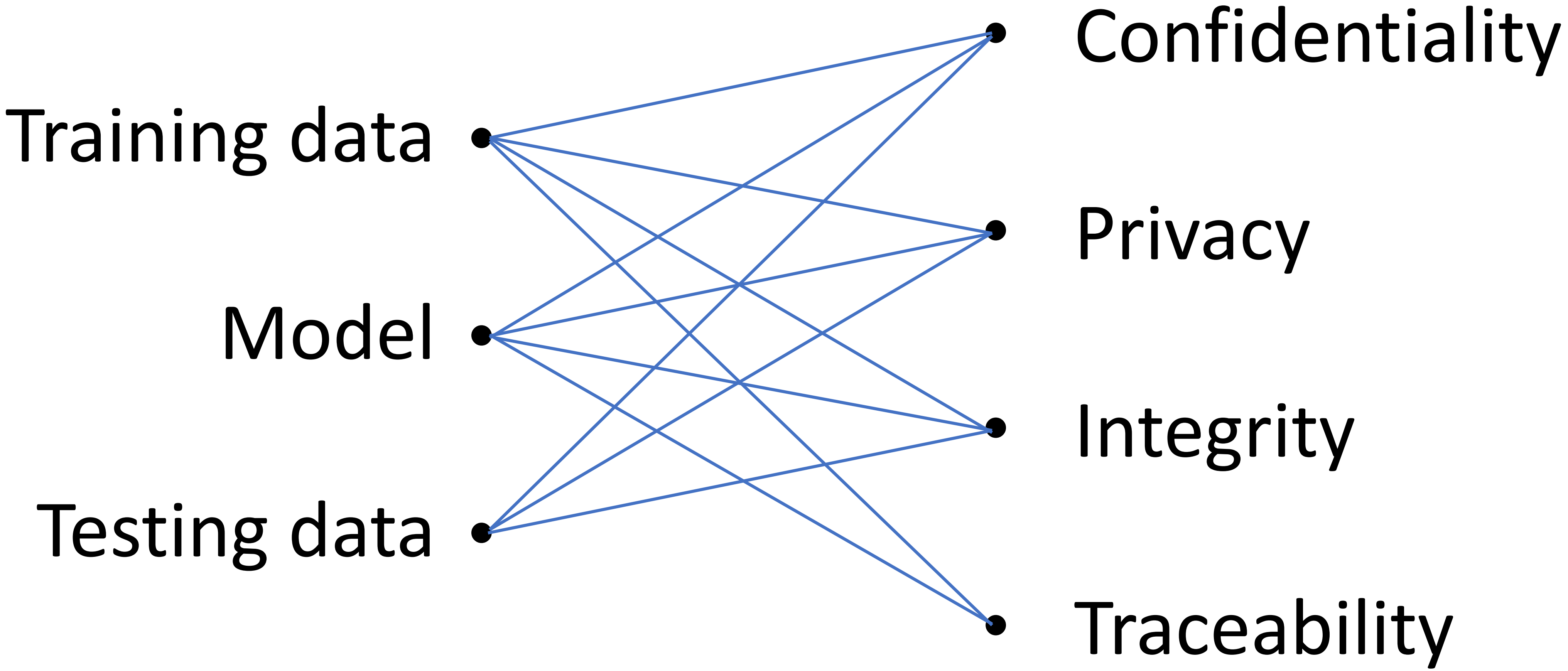
[Supervised GAN Watermarking for Intellectual Property Protection, Fei, arXiv'22]

[Proactive Detection of Voice Cloning with Localized Watermarking, San Roman, arXiv'24]

[The Stable Signature: Rooting Watermarks in Latent Diffusion Models, Fernandez, ICCV'23]

- Testing data
  - ???

Security of Machine Learning



- 3 objects x 4 values - 1 = 11 scenarios
- 11 x types of data x types of learning framework x types of DNN

2- Where do we stand?

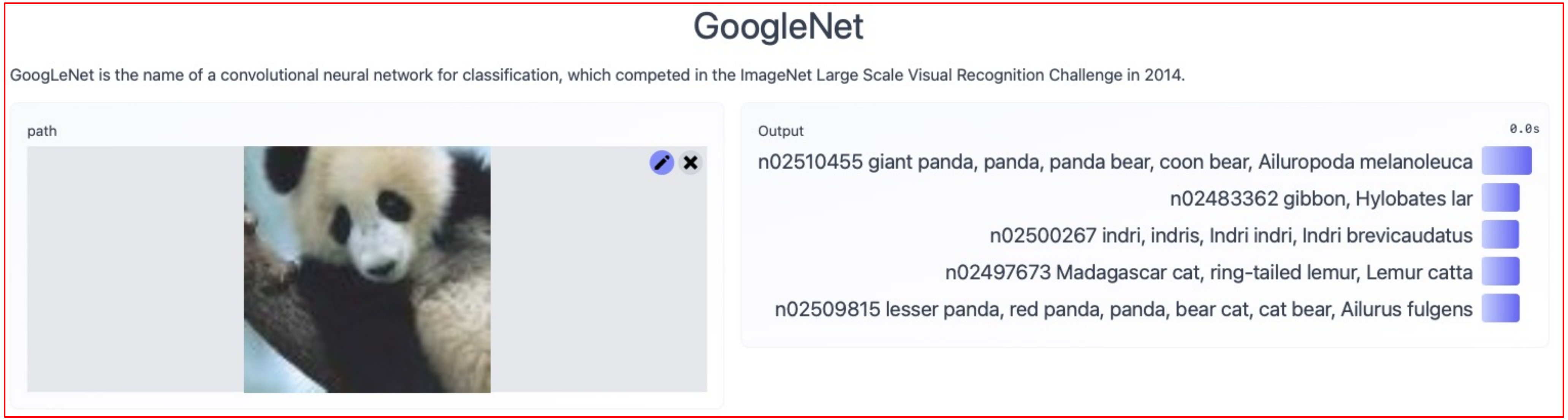
Where do we stand?

1. The Rocky Horror Picture Show
   - Empirical Evidence of Attacks
   - Alarming, Threatening

2. Research work in the lab
   - Reproducibility
   - Empirical discovery of key factors
   - Theoretical explanations

3. Real life: Auditing, Advising
   - Run SotA attacks and see …

Where do we stand? Adversarial examples



- Not reproducible
- Explanation (?):
  - adversarial examples = tensor of scalars ≠ tensor of integers

Where do we stand? Adversarial examples

- Naïve defenses are not working
  - Gradient obfuscation

  “*Since all white-box attacks resort to the gradient of the neural network, just introduce a non-linearity to forbid its computation*”

$$f = f_1 \circ f_2 \quad \rightarrow \quad f_Q = f_1 \circ Q \circ f_2$$

  - The attacker is not obliged to do so!

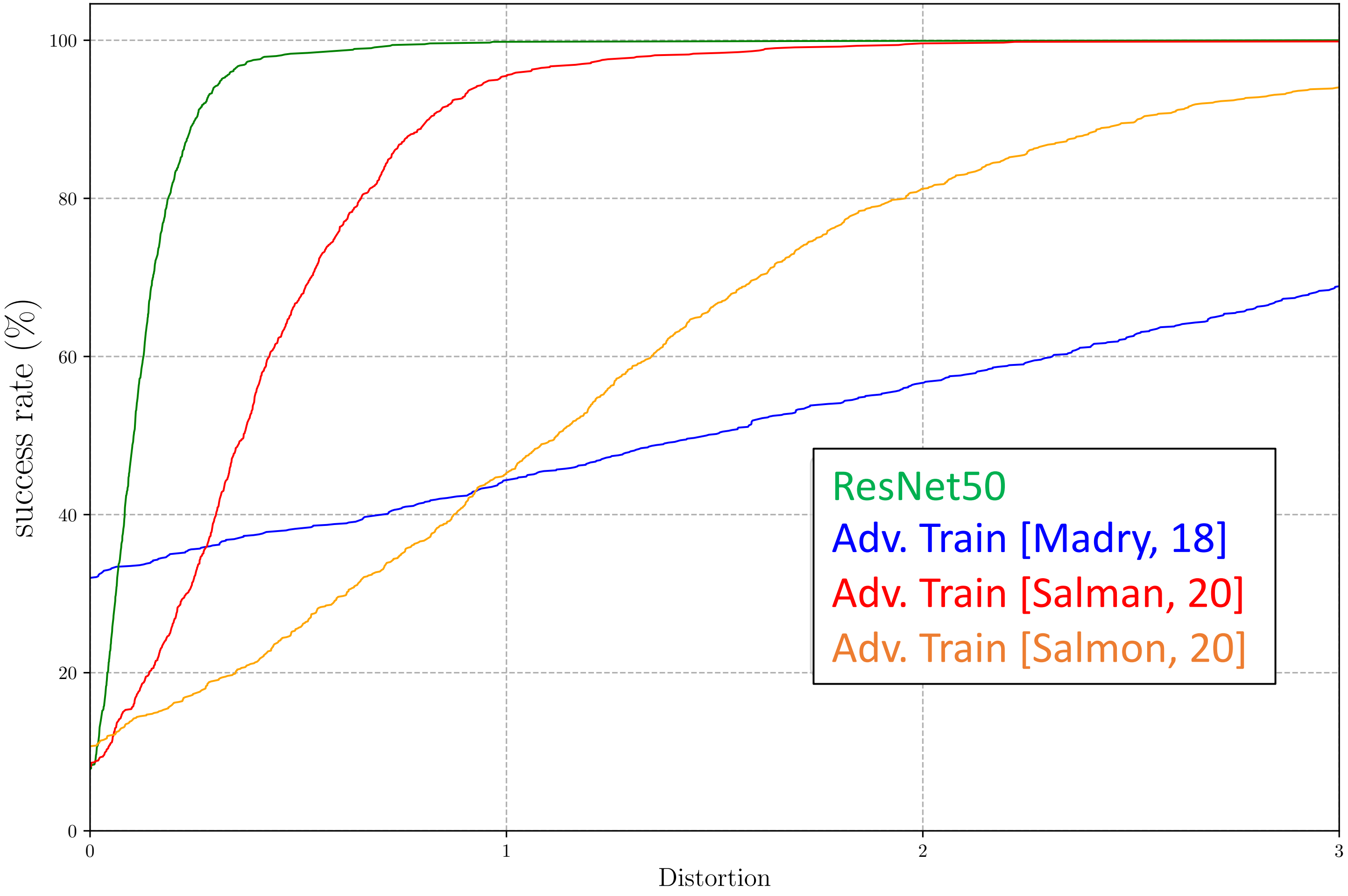
  [*Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples*, Athalaye, ICML 2018]
    - This paper circumvents 7 defenses proposed in ICLR 2018

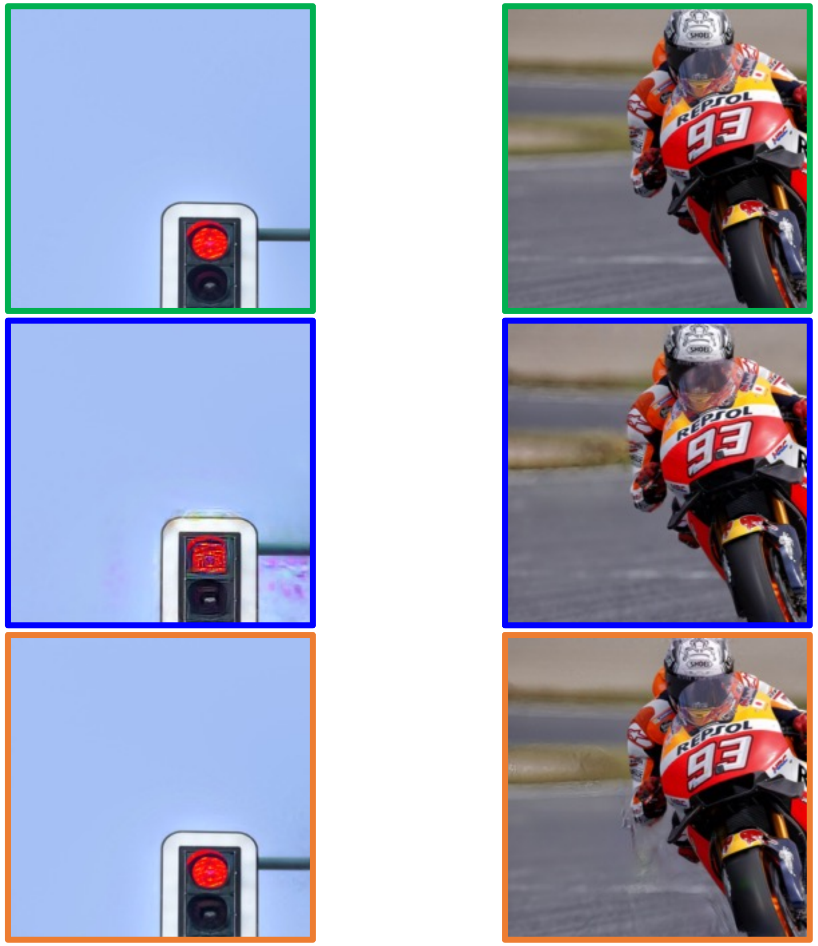
Where do we stand? Adversarial examples

- Proposal of best practices for evaluating attacks/defenses
  - [*On Evaluating Adversarial Robustness*, Carlini, arXiv 2019]
- Fear Nicholas Carlini (Google Deepmind)
  - [*Cutting through buggy adversarial example defenses: fixing 1 line of code breaks Sabre*, Carlini, arXiv 2024]
  - Significant flaws in *Sabre*, defense paper accepted at IEEE S&P 2024
  - Not following any of the best practices

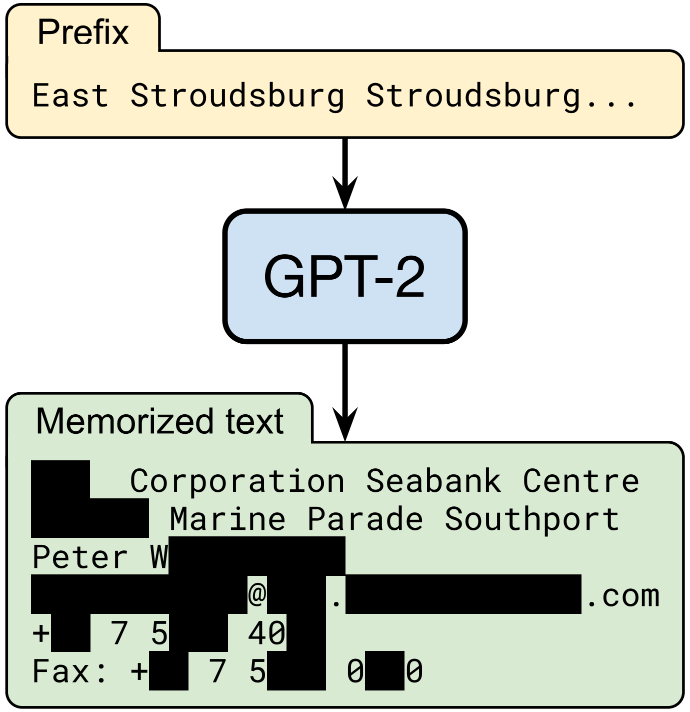
Where do we stand? Adversarial examples

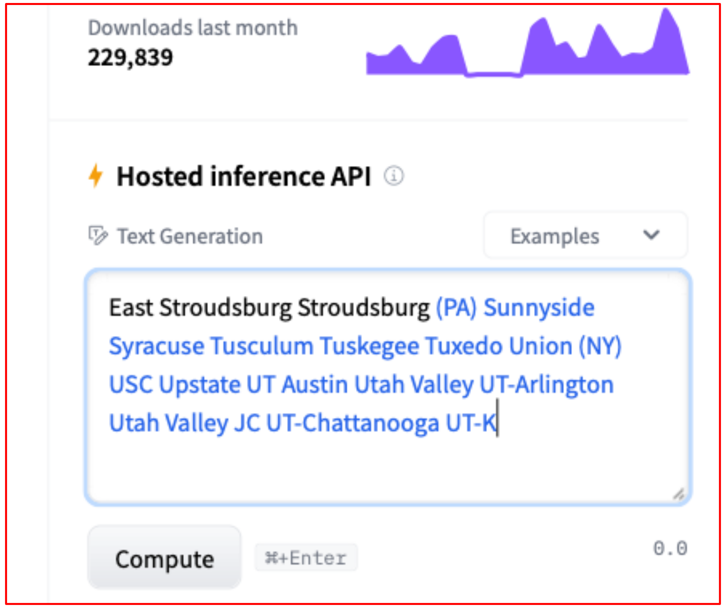
- Consensus: Adversarial training is the only way to go (?)





Where do we stand? Training data confidentiality



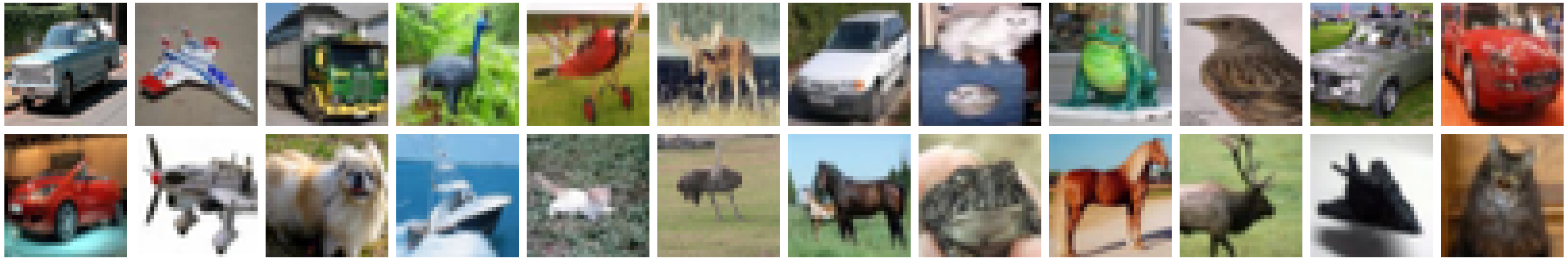


- Not reproducible
- Not explainable

Where do we stand? Training data confidentiality

(a) Top 24 images reconstructed from a binary classifier trained on 50 CIFAR10 images

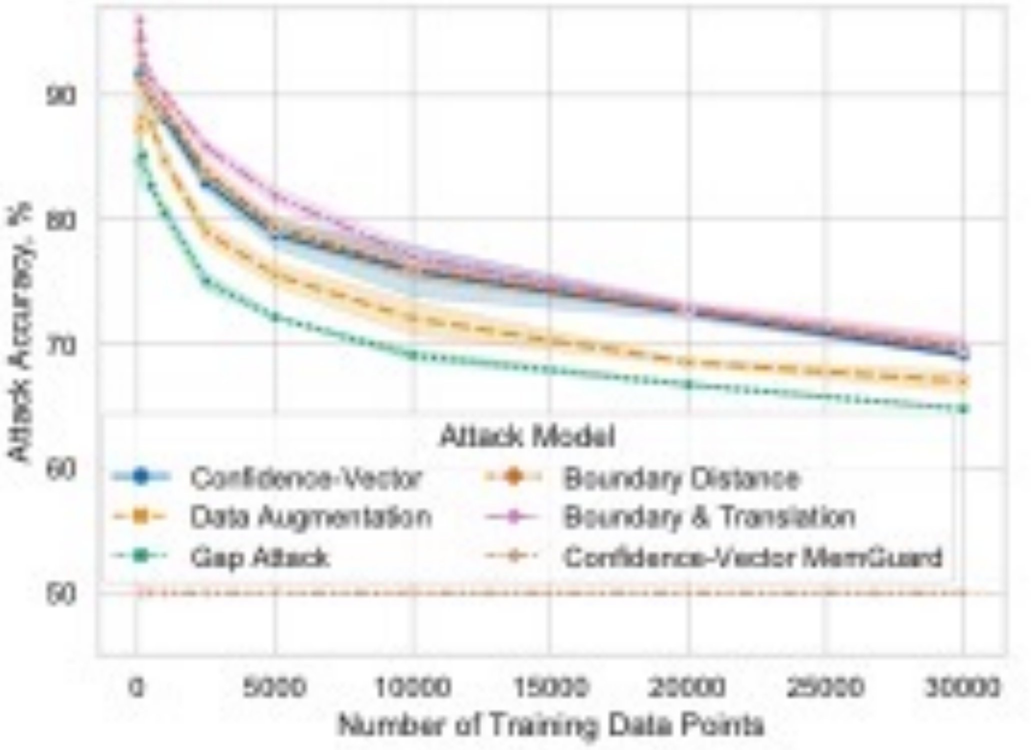
(b) Their corresponding nearest neighbours from the training-set of the model

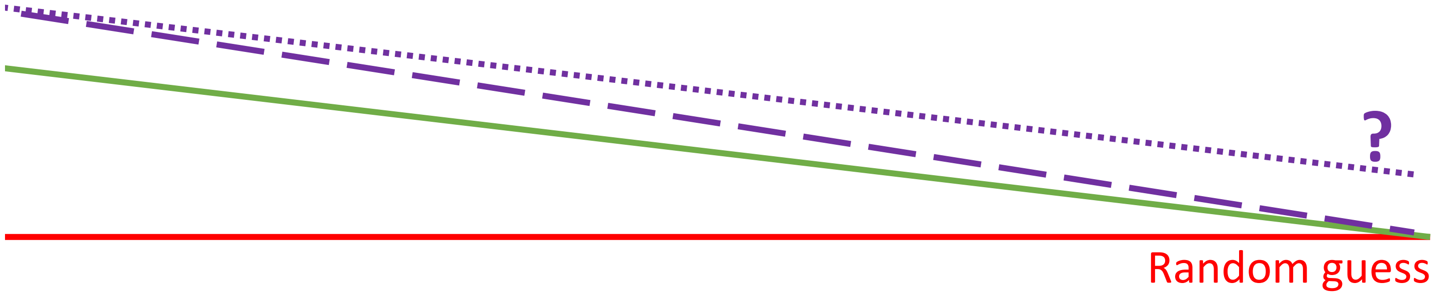


- Strong theoretical limitations
  - Binary classification
  - Homogeneous neural networks (no biases, no residuals)
- Experimental evidence
  - On 3-layer MLPs

[Reconstructing Training Data from Trained Neural Networks, Haim, NeurIPS’22]

Where do we stand? Training data confidentiality





- Clear impact of the overfitting
- Outliers in the training set are more easily discovered

[Label-Only Membership Inference Attacks, Choquette-Choo, ICML’21]

Security of Machine Learning

- Study the Security of ML before applying ML to Security
- Simple definition
  - (Training d., Model, Testing d.) x (Confidentiality, Privacy, Integrity, Traceability)
  - Almost sound and almost complete
- Where do we stand?
  - In the lab!
  - In real life: "It depends"
- As a reader: adversarial reading of adversarial ML papers
- As a writer: be skeptical about your results
  - *"the first principle [of research] is that you must not fool yourself—and you are the easiest person to fool*". R. Feynman
  - Switch your mindset: play the attacker/defender role

3- Case studies

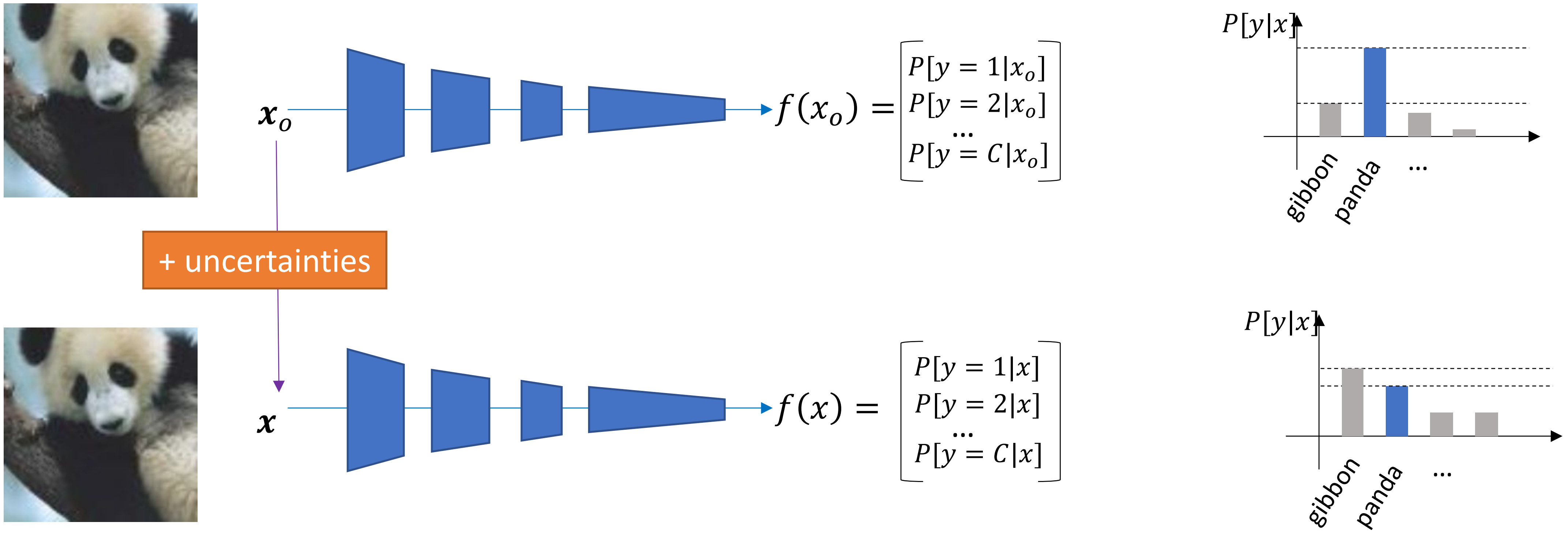
3a- Robustness

Karim Tit et al.

Efficient Statistical Assessment of Neural Network Corruption Robustness, NeurIPS 2021

Gradient-Informed Neural Network Statistical Robustness Estimation, AISTATS 23

Problem



Probits = “predicted” probabilities

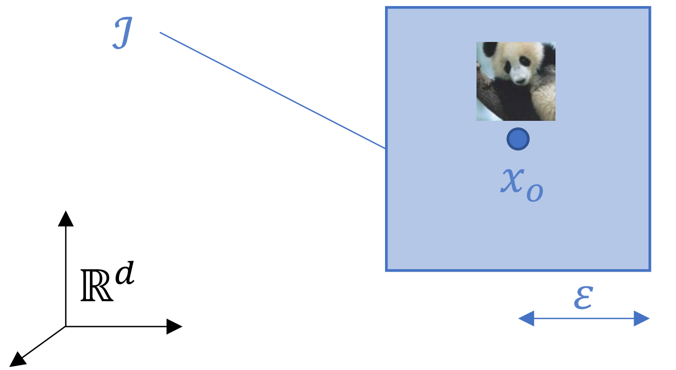
Problem

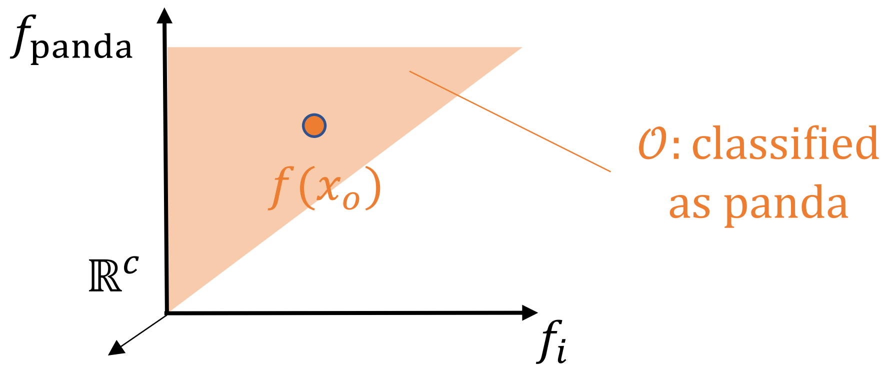
Local certification in classification

- Consider $x_o \in \mathbb{R}^d$, well classified

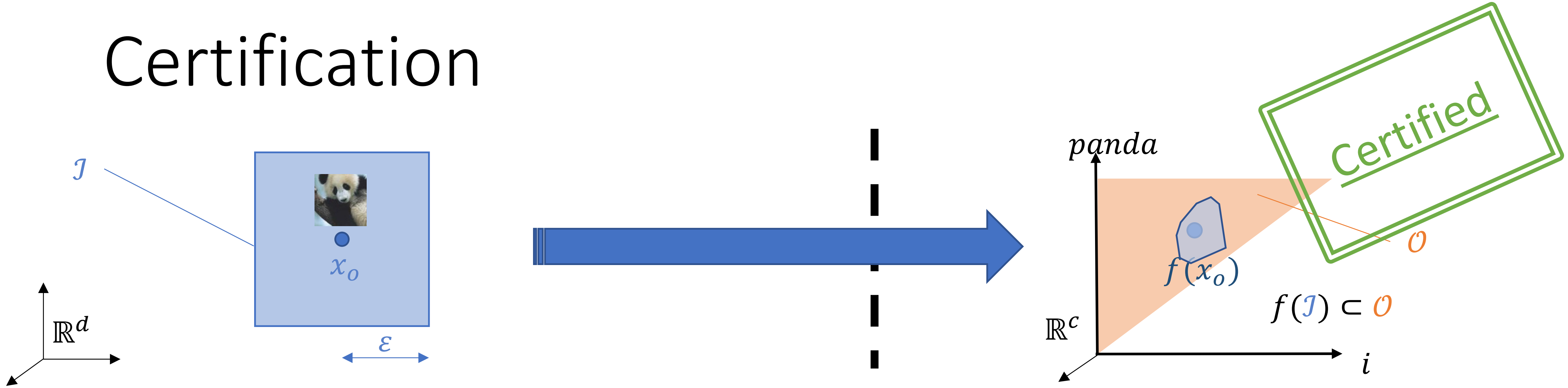
$$\arg\max_i f_i(x_o) = panda$$

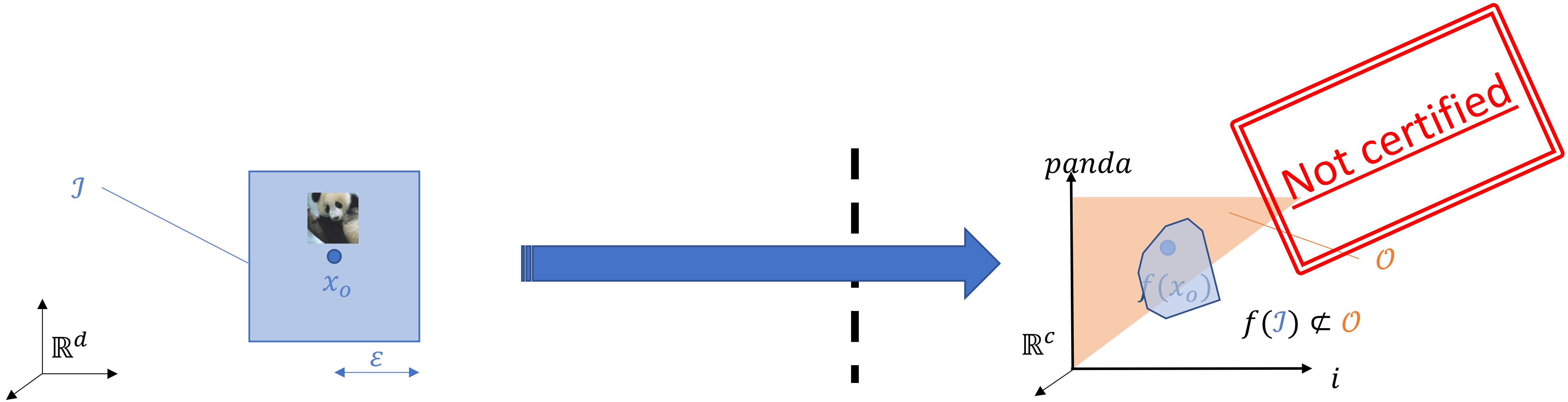
- Consider two regions
  - Input region: $\mathcal{I} = \{ x \in \mathbb{R}^d \mid d(x, x_o) \leq \varepsilon \} \subset \mathbb{R}^d$
  - Output region: $\mathcal{O} = \{ f \in \mathbb{S}^c \mid \arg\max_i f_i = panda \} \subset \mathbb{R}^c$



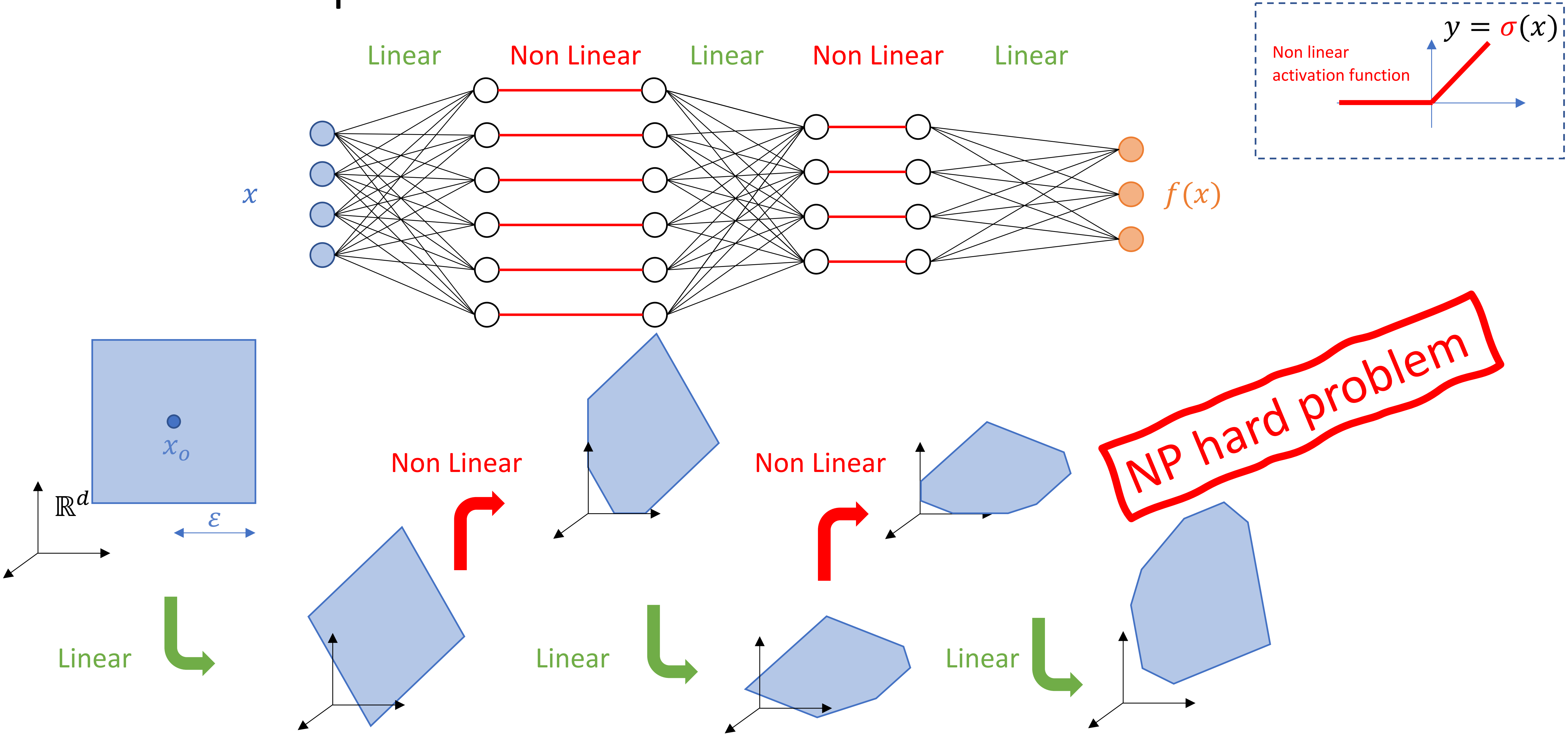


Certification

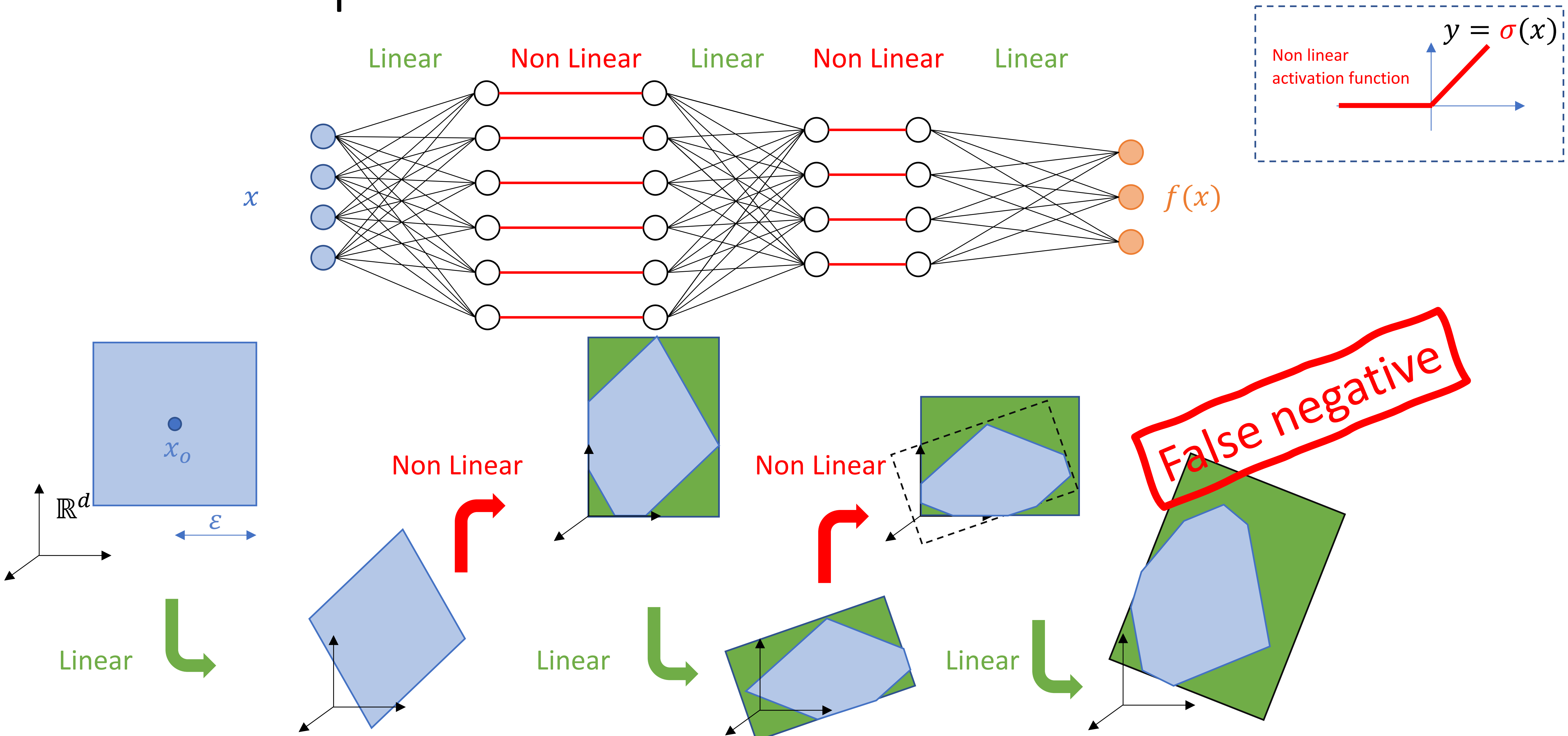




Formal proof



Formal proof with relaxation



Formal proof

- Sound and complete (but not scalable)
  - ReLUplex, Katz *et al.*, Computer Aided Verification 2017

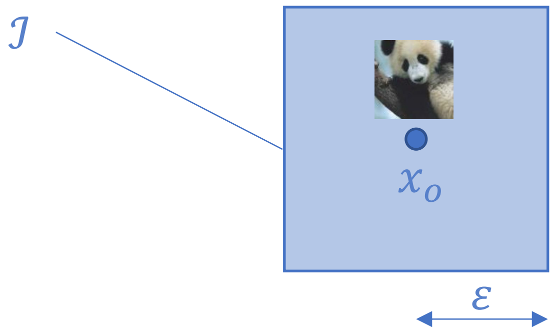
- Relaxation (not complete) but more scalable
  - Crown, Zhang *et al.*, NeurIPS 2018
  - CNN-CERT, Weng *et al.*, AAAI 2019
  - DeepPoly, Singh *et al.*, Programming Languages, 2019
  - Fast-Lin, Weng *et al.*, ICML 2018 (backward)

Since formal methods are not so formal, let us try a statistical approach

Our approach: statistical certification

- Assume a statistical distribution of the input

  For example, $X \sim \mathcal{U}(\mathcal{J})$



- Define probability of failure

$$p = \mathbb{P}[\, f(X) \notin \mathcal{O} \,]$$

- Hypothesis Testing wrt p_c critical level set by the user
  - H_0: $p > p_c$ Do not certify
  - H_1: $p < p_c$ Certify
- Run a statistical simulation and decide upon its random result
- 2 types of errors
  - False Positive: Certify whereas $p > p_c$
  - False Negative: Do not certify whereas $p < p_c$

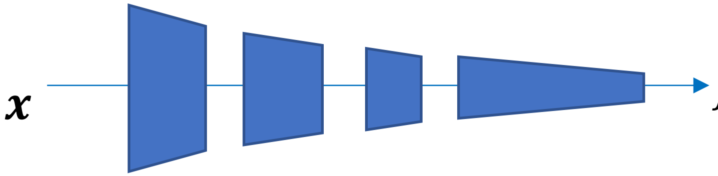
Which statistical simulation?

- Monte Carlo
  - Randomly draw N samples $X_i = x_o + U_i$ and count the number of adv. examples
  - Pros: Any distribution
  - Cons: $N = O(1/p_c)$

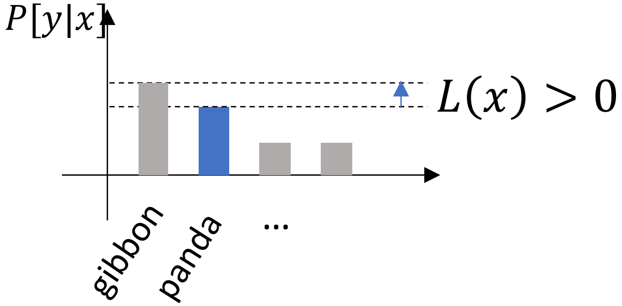
ℓ_∞ norm ℓ_2 norm   IID Gaussian   colored Gaussian

- Rare event simulation
  - FORM, SORM, Importance Sampling, Importance Splitting, …
  - We are inspired from Last Particle algorithm [Guyader *et al.*, 2011]
  - Pros: Any distribution, control over FPR $< \alpha$
  - Complexity $= O(\log(1/p_c))$

Connection with ML



$$f(x) = \begin{bmatrix} P[y=1|x] \\ P[y=2|x] \\ \cdots \\ P[y=C|x] \end{bmatrix}$$

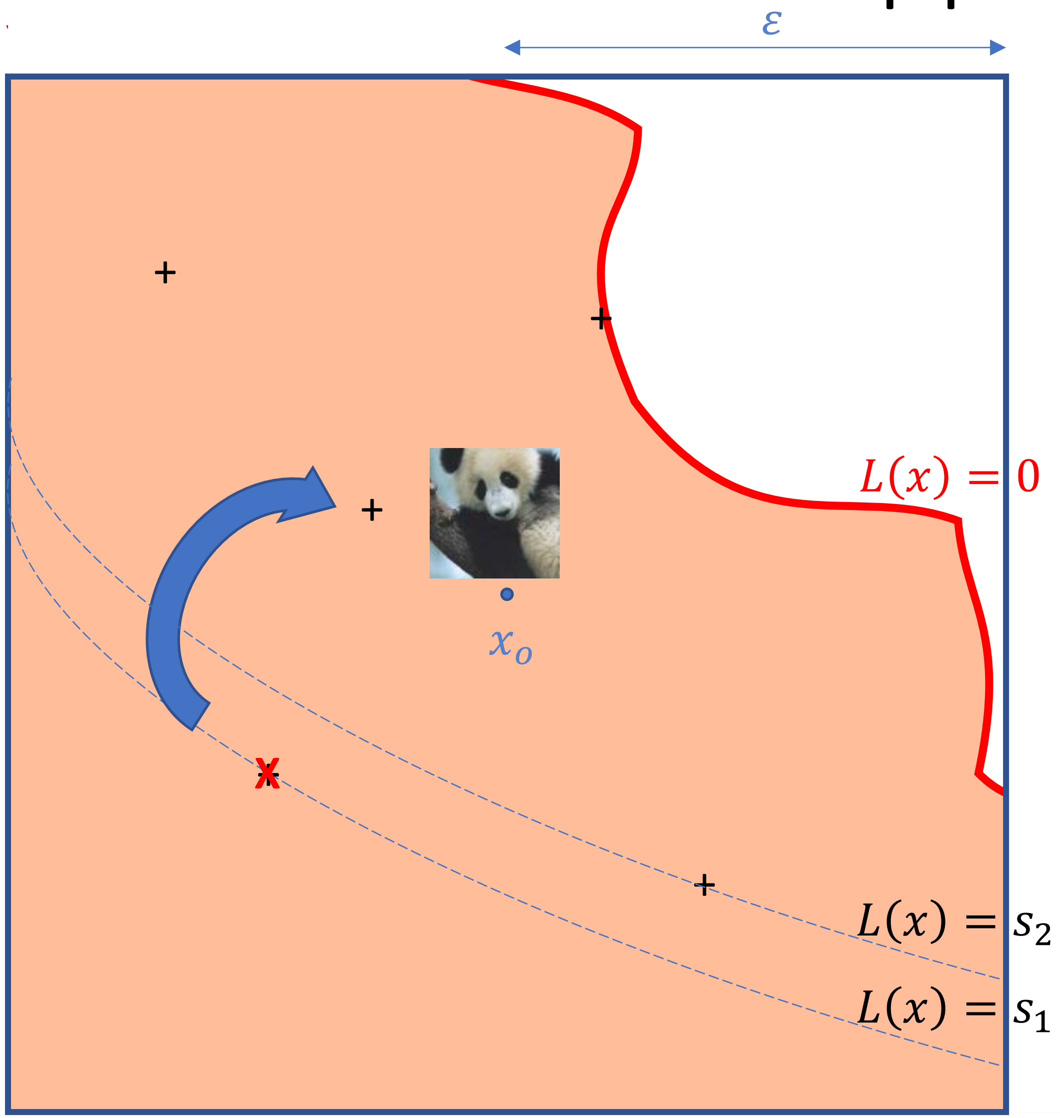


$$L(x) := \max_{y \neq y_o} f_y(x) - f_{y_o}(x)$$

This quantity tells how close the uncertainties are to delude the classifier

Sample U $\rightarrow$ $X = x_o + U$ $\rightarrow$ $V = L(X)$ $\rightarrow$ $p = \mathbb{P}[V > 0] \overset{?}{<} p_c$

The Last Particle applied to ML



Randomly draw N samples

$$X_i = x_o + U_i$$

Repeat m times

Compute scores

$$L(X_1), \dots, L(X_N)$$

Find minimum

$$i^* = \arg\min L(X_i)$$

Define threshold

$$S \longleftarrow L(X_{i^*})$$

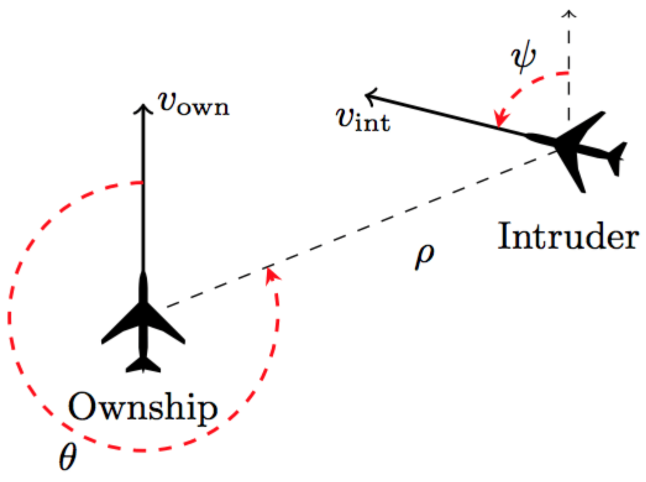
Replace with one fresh particle

$$X_{i^*} \longleftarrow x_o + U \text{ such that } L(X_{i^*}) > S$$

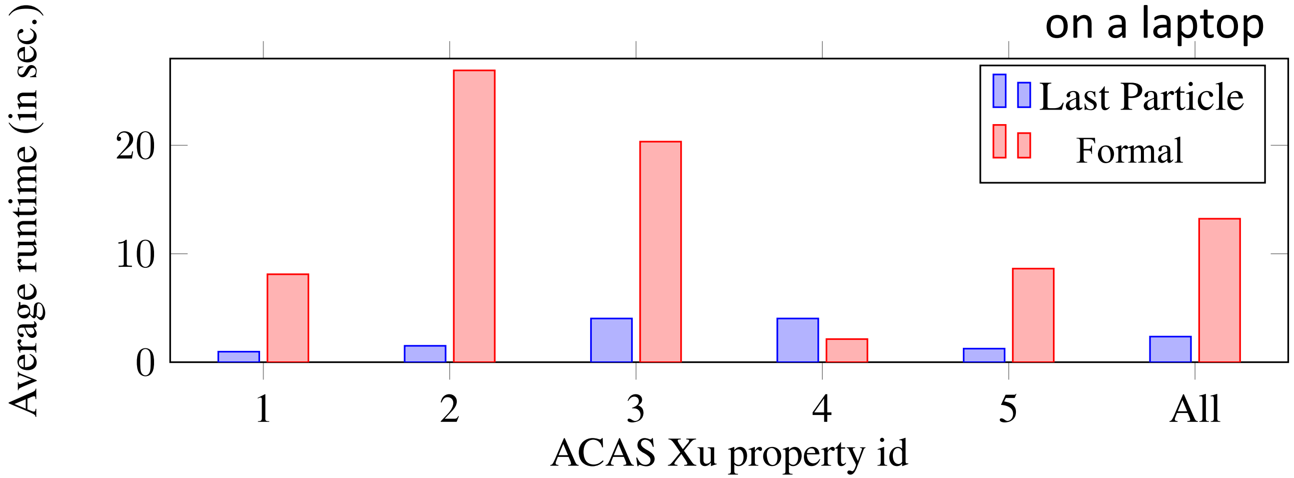
$$m \approx \tilde{m}_1 = \left\lceil \frac{1}{4} \left(z_\alpha + \sqrt{z_\alpha^2 - 4N \log(p_c)} \right)^2 \right\rceil$$

with $z_\alpha = \Phi^{-1}(1 - \alpha)$

Experimental results: ACAS-Xu



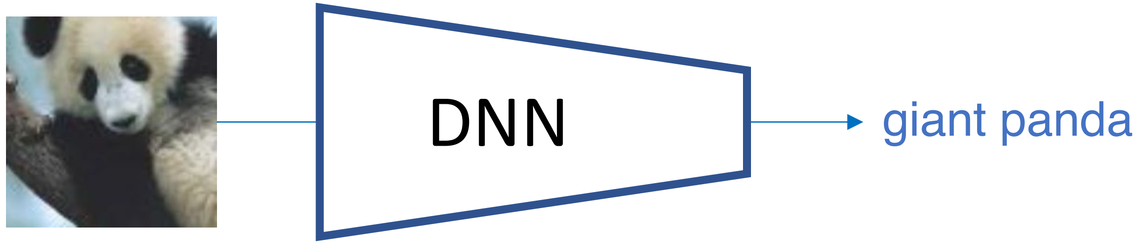
on a laptop



		Formal			
		Certified	Uncertified	Infeasible	TimeOut
Last Particle	Certified	107	9	1	1
	Uncertified	0	103	4	0

$p_c = 10^{-50}, \alpha = 0.05$

Experimental results: ImageNet



No large scale result in formal proof literature on such big input data / model

Network	ϵ	Avg. runtime (in sec. $\pm std$)	Avg. number of calls	Certified (%)
MobileNet	0.02	20.78 ± 0.74	1388	71
	0.03	18.74 ± 0.18	1274	64
	0.06	14.5 ± 0.11	1037	50
ResNet50	0.02	33.86 ± 1.14	1537	81
	0.03	31.38 ± 0.48	1434	71
	0.06	25.51 ± 0.67	1160	59

$p_c = 10^{-15}, \alpha = 0.05$, 100 images, NVIDIA V100

Robustness

- DNN classifiers are extremely robust
  - Locally robust
  - But it is not trivial to certify this property

- Does it matter?
  - Misclassification rate: ACAS-Xu $\approx$ 1% / ImageNet $\approx$ 20%
  - Impossible to derive how to improve robustness

- And yet, they are vulnerable…

3b- Adversarial examples

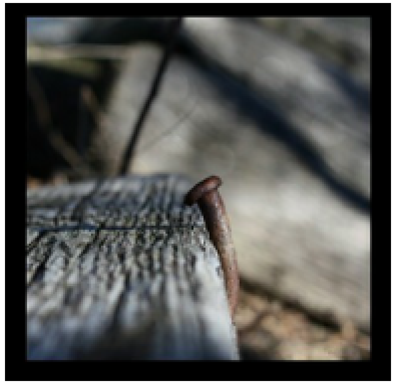
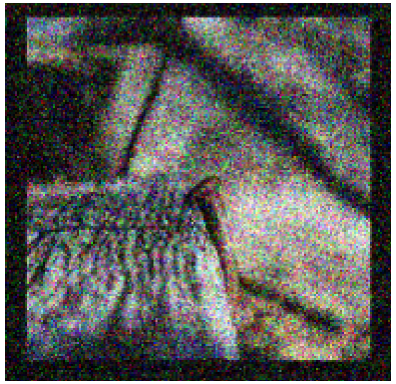
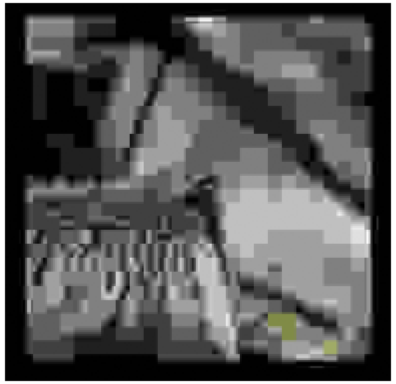
Security $\neq$ Robustness

Motivations: false sense of security

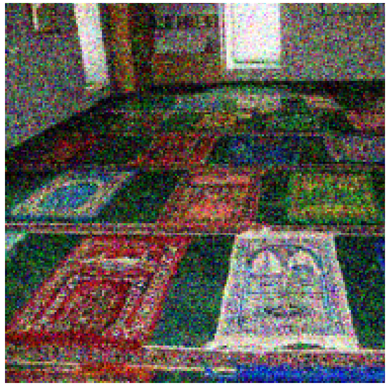
- Generalization $\neq$ Robustness $\neq$ Security

  - Generalization: To operate as expected on unseen data
  - Robustness: To operate as expected on noisy data
  - Security: To operate as expected on purposely perturbed data

Security ≠ Robustness

		Robustness		Security	
	original	noise	JPEG	black-box	white-box
					
Prediction	nail	enveloppe	bulletproof_vest	paintbrush	mantis
Distortion	0	84.9	28.8	6.6	0.2

Security ≠ Robustness

		Robustness		Security	
	original	noise	JPEG	black-box	white-box
					
Prediction	prayer_rug	lighter	loudspeaker	quilt	safe
Distortion	0	79.1	42.0	19.2	0.5

Security ≠ Robustness

		Robustness		Security	
	original	noise	JPEG	black-box	white-box
					
Prediction	Lawn_mower	projector	joystick	vacuum	rifle
Distortion	0	73.2	14.5	4.5	0.14

Methodology

giant panda

$\boldsymbol{x}_o$

$+\,\epsilon\,*$

$=$

$\boldsymbol{x}_a$

gibbon

Optimal untargeted adversarial example

$$\boldsymbol{x}_a^* = \arg\min_{\hat{y}(\boldsymbol{x})\neq \text{panda}} d(\boldsymbol{x}, \boldsymbol{x}_o)$$

Design an attack

attack algorithm

$$\boldsymbol{x}_a = A(\boldsymbol{x}_o, \theta, \varphi) \quad \text{as close as possible to} \quad \boldsymbol{x}_a^*$$

original image

attack parameters

DNN model

Methodology

- Best effort
  - Find the right parameters for each image

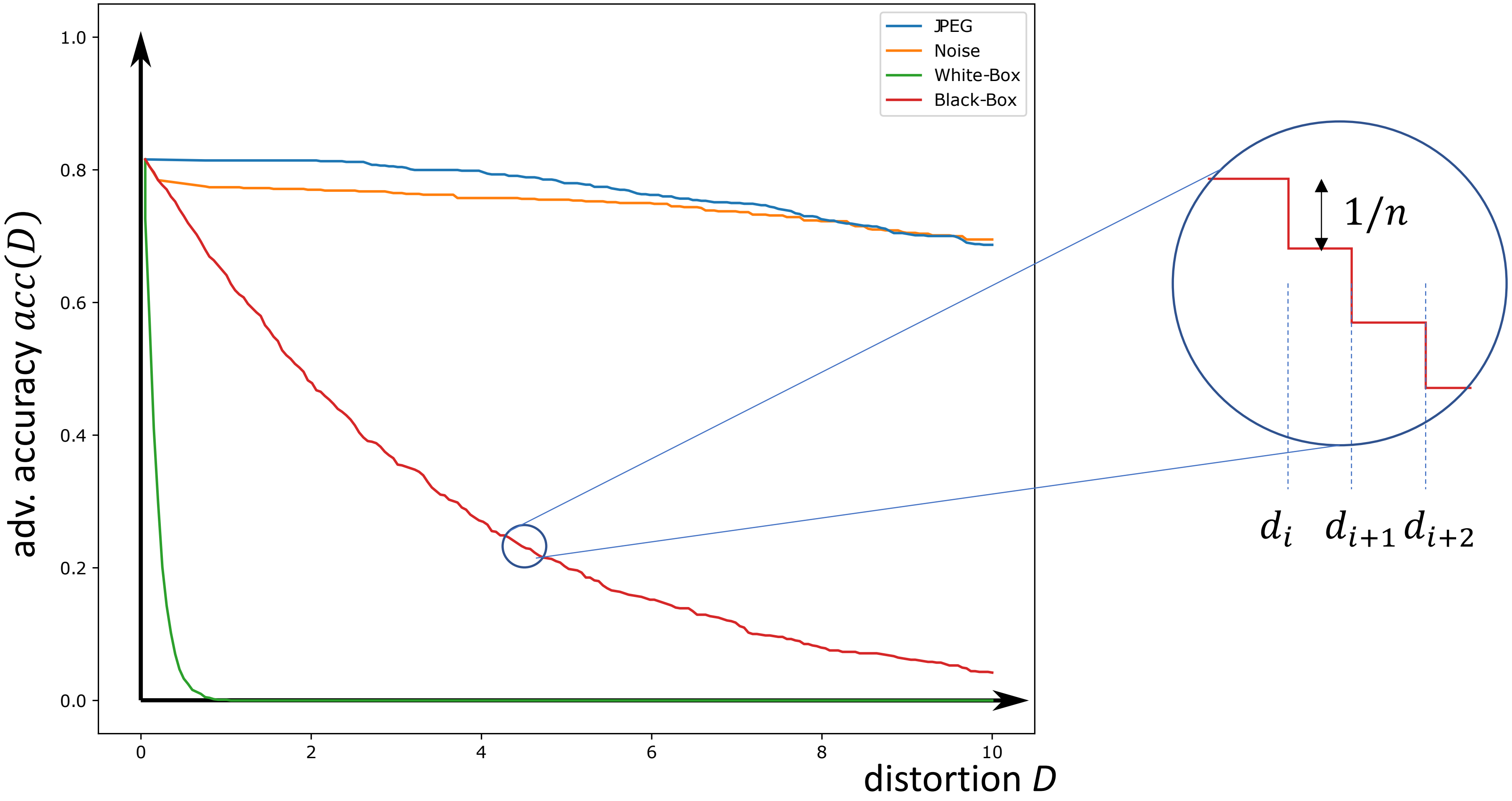
$$\varphi^* = \arg\min \quad d(A(\boldsymbol{x}_o, \theta, \varphi), \boldsymbol{x}_o)$$

- Operating curve
  - Attack a set of n images, sort the distortions

$$d_1 \leq d_2 \leq \cdots \leq d_n$$

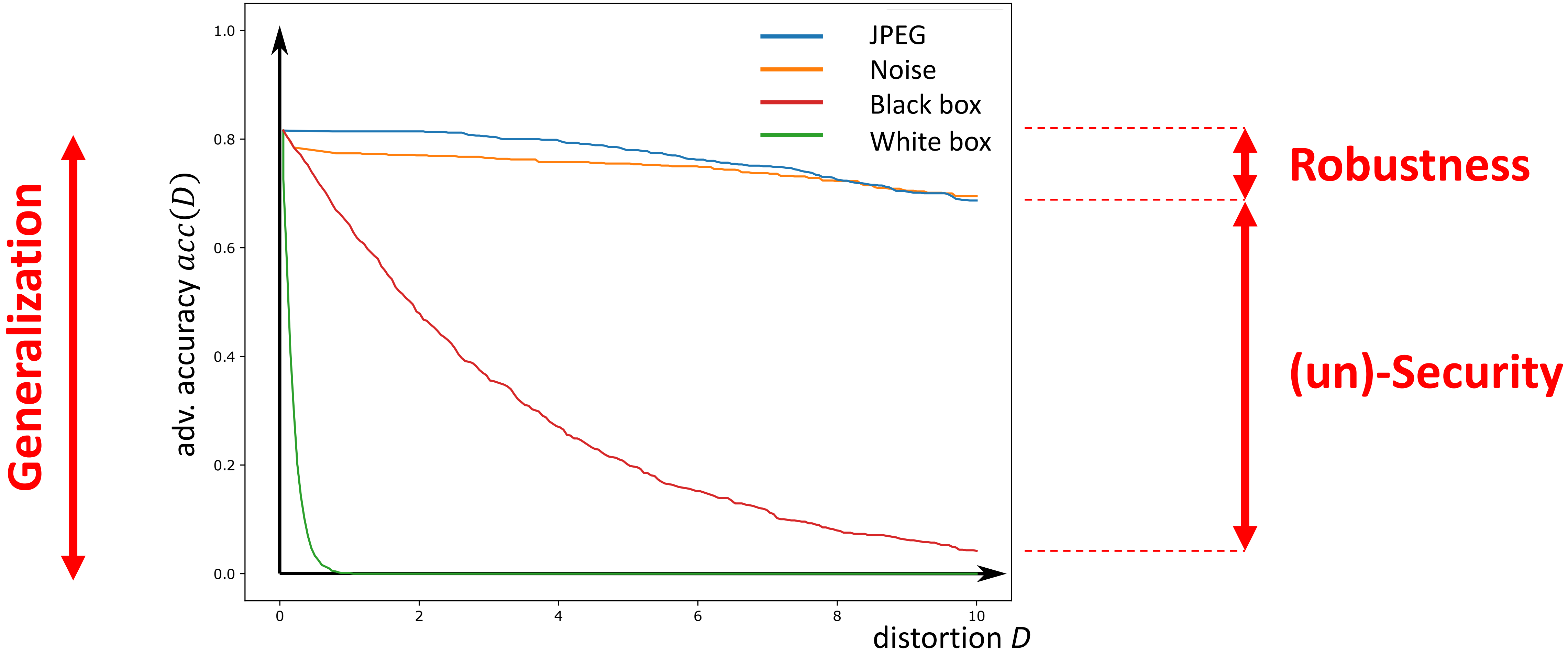
  - Plot one of these functions
    - Attack Success Rate $\quad P(D) = \frac{1}{n}\sum[d_i \leq D]$
    - Adversarial accuracy $\quad acc(D) = 1 - P(D)$

Methodology



$$d(x_a, x_o) = \frac{\|x_a - x_o\|_2}{\sqrt{n}}$$ with $x \in [\![0,255]\!]^n$

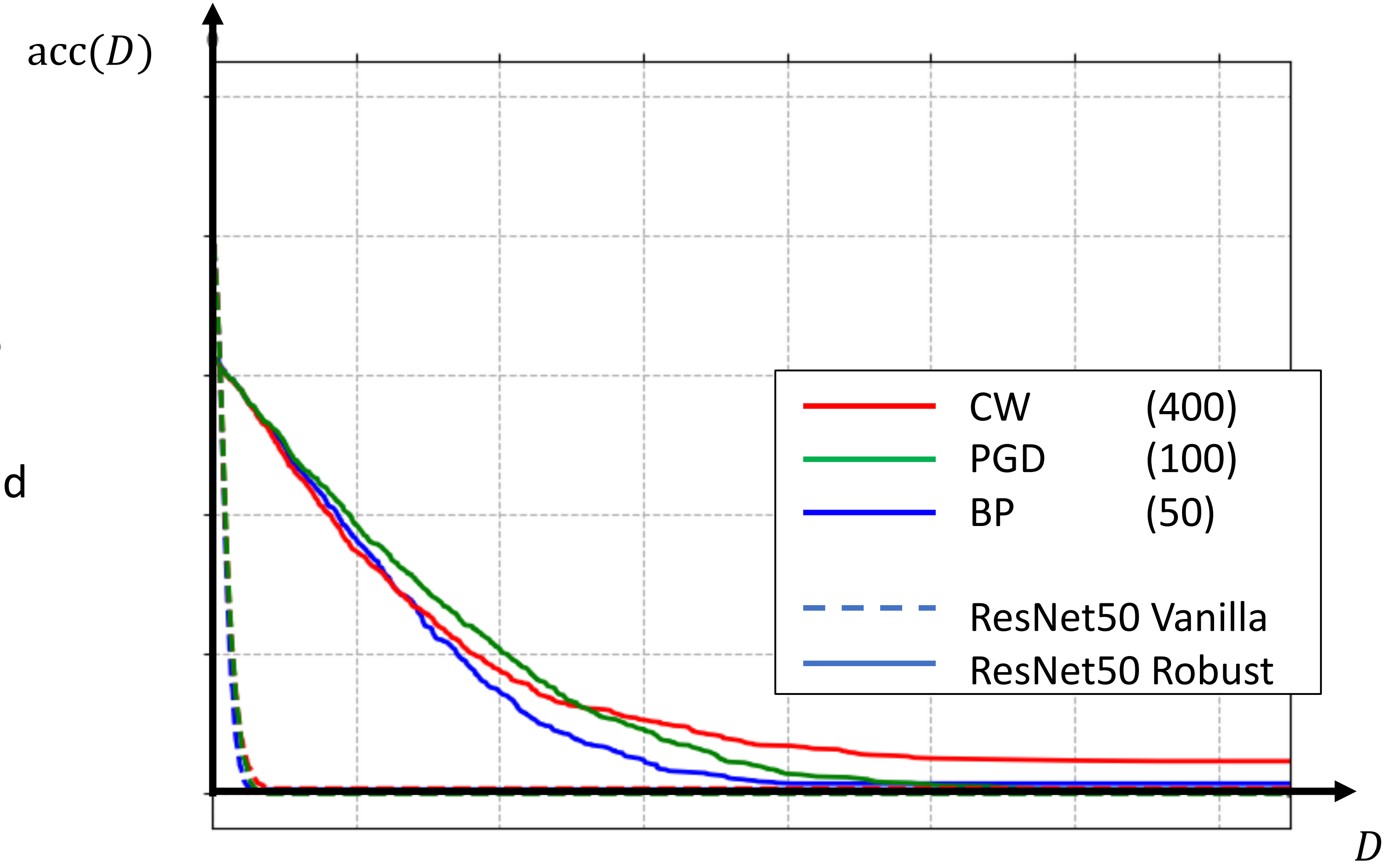
Security ≠ Robustness



Fair comparison

Best effort + Operating curve

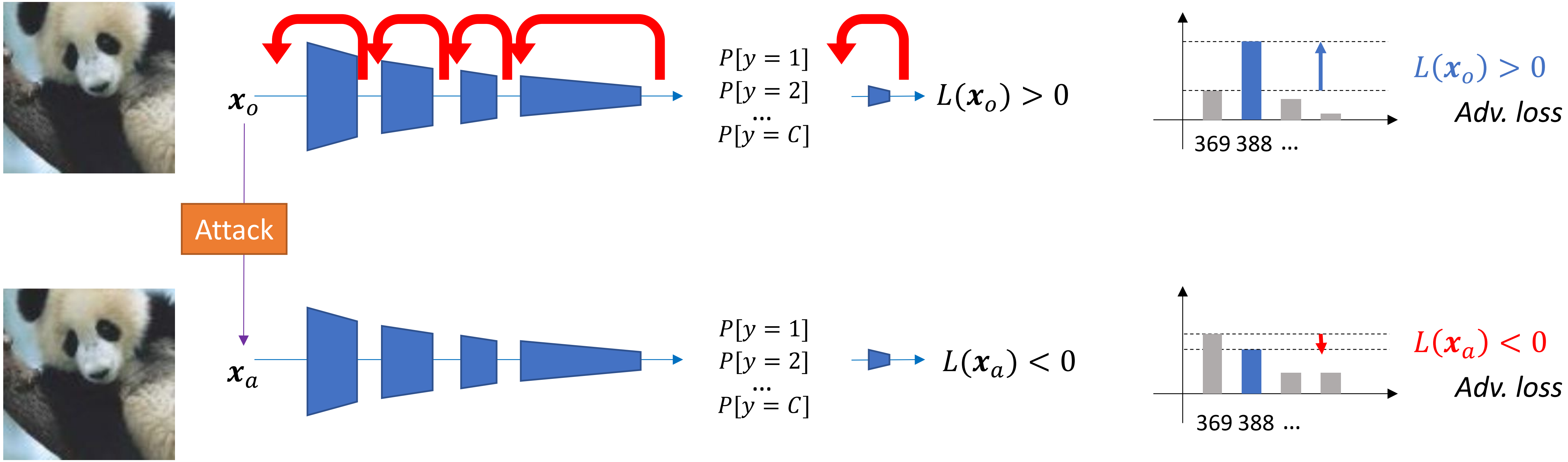
- Attacks of different nature
  - Distortion vs. Success oriented
  - White vs. Black attacks
- Different models
  - with/without defenses



Problem: High complexity due the best effort mode

- We need fast and powerful attacks:
  1. Successful (almost surely)
  2. Low distortion
  3. Few parameters (or parameters free)
  4. Fast

How white-box attacks work?



$$L(\boldsymbol{x}) = P[y_o] - \max_{y \neq y_o} P[y] \qquad \& \qquad \nabla L(\boldsymbol{x})$$ (by autodiff / backpropagation)

Fast attack = Few gradient computations

How white-box attacks work?

- Optimal untargeted adversarial example

$$\boldsymbol{x}_a^* \quad = \arg \min_{L(\boldsymbol{x})=0} d(\boldsymbol{x}, \boldsymbol{x}_o)$$

- Example: Lagrangian formulation [Carlini&Wagner, IEEE S&P, 2017]

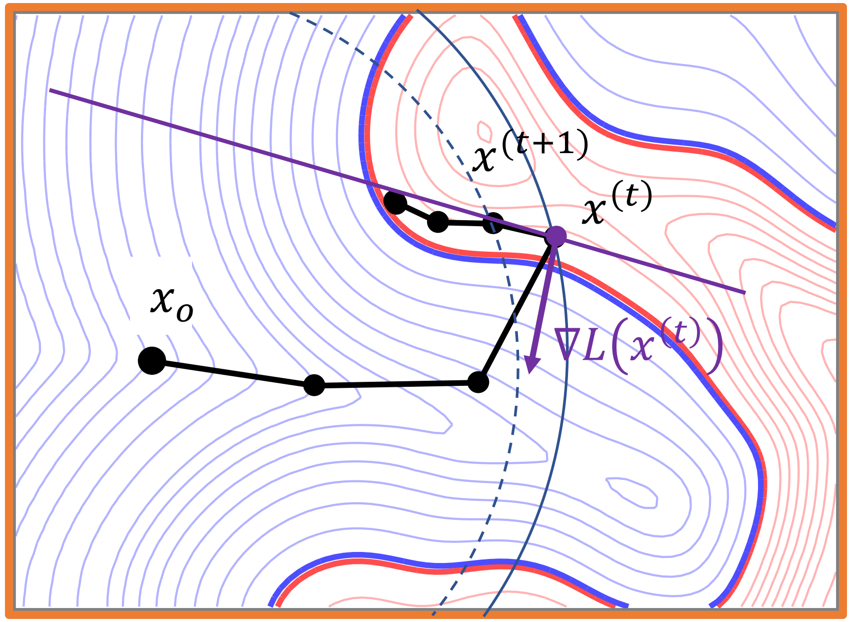
$$J(\boldsymbol{x}, \lambda) = d(\boldsymbol{x}, \boldsymbol{x}_o) + \lambda\, L(\boldsymbol{x})$$

- 2 nested loops
  - Line search over λ
    - Use for preferred solver using $\nabla J(\boldsymbol{x}, \lambda)$

$$\boldsymbol{x}_\lambda^* = \text{arg min } d(\boldsymbol{x}, \boldsymbol{x}_o) + \lambda\, L(\boldsymbol{x})$$

    - If $L\,(\boldsymbol{x}_\lambda^*) > 0$, then increase λ
    - If $L\,(\boldsymbol{x}_\lambda^*) < 0$, then decrease λ

BP - Boundary Projection



Parameter = number of iterations

Best performance within ~50 iterations

Algorithm

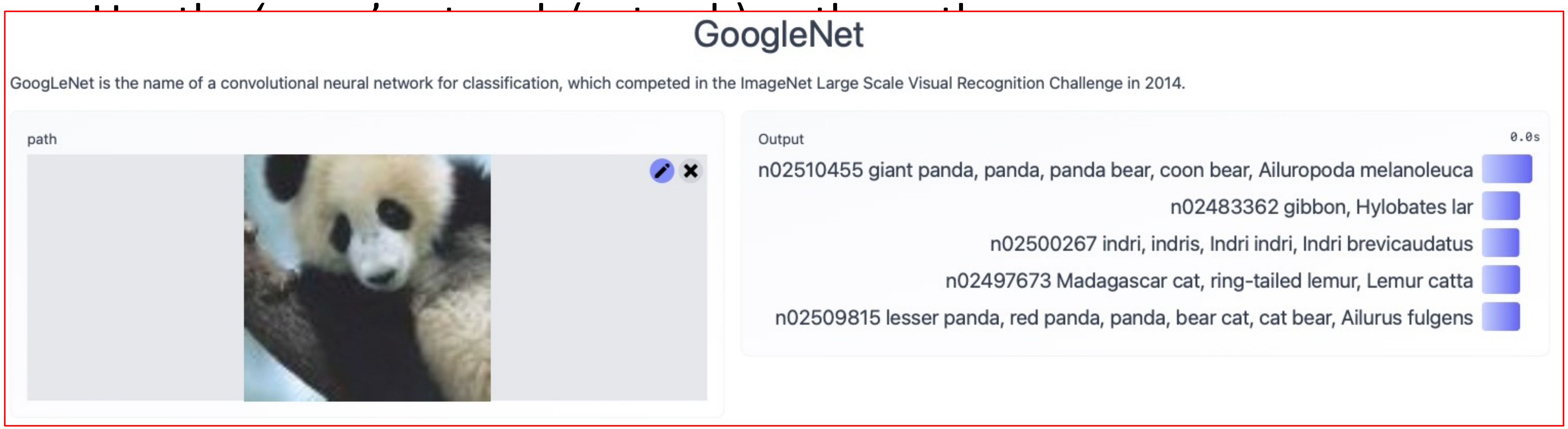
- Stage 1: Fast & Furious
  - Go out as quickly as possible
  - Gradient descent with increasing step size
- Stage 2: Nice & Gentle (inspired by Statistical Reliability method HL-RF)
  - OUT: decrease distortion while maintaining the loss
  - IN: decrease the loss while (almost) maintaining the distortion

Walking on the Edge: Fast, Low-Distortion Adversarial Examples, Hanwei Zhang et al., IEEE TIFS 2020
Structural reliability under combined random load sequences, Rackwitz, Fiessler, Comp. Struct. 1978

The deep scam?

Illustration of adversarial images … are not often adversarial!

- Unbundle the .pdf to retrieve the image files… as generated by the authors (not a bad quality screenshot)



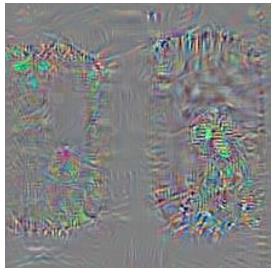
«*Explaining and Harnessing Adversarial Examples*» Goodfellow, Szegedy, et al., early 2015

The deep scam?

Illustration of adversarial images … are not always adversarial!

632: ‘loudspeaker’
58%

$+ \epsilon *$

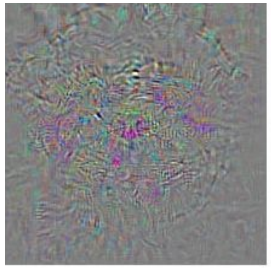


=

632: ‘loudspeaker’
34%

155: ‘pekinese’
61%

$+ \epsilon *$

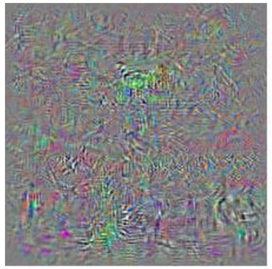


=

155: ‘pekinese’
82%

779: ’school bus’
51%

$+ \epsilon *$



=

779: ’school bus’
45%

« *Intriguing properties of neural networks* » Szegedy, Goodfellow et al., early 2014

Rounding destroys perturbations

- Reverse the pre-processing and round: $[0,1]^d \longrightarrow \{0,1,\ldots,255\}^d$

$$I_a = [255 * x_a] = [255 * (x_o + p)] = I_o + [255 * p]$$

- Rounding is quantizing with step $\Delta = 1$

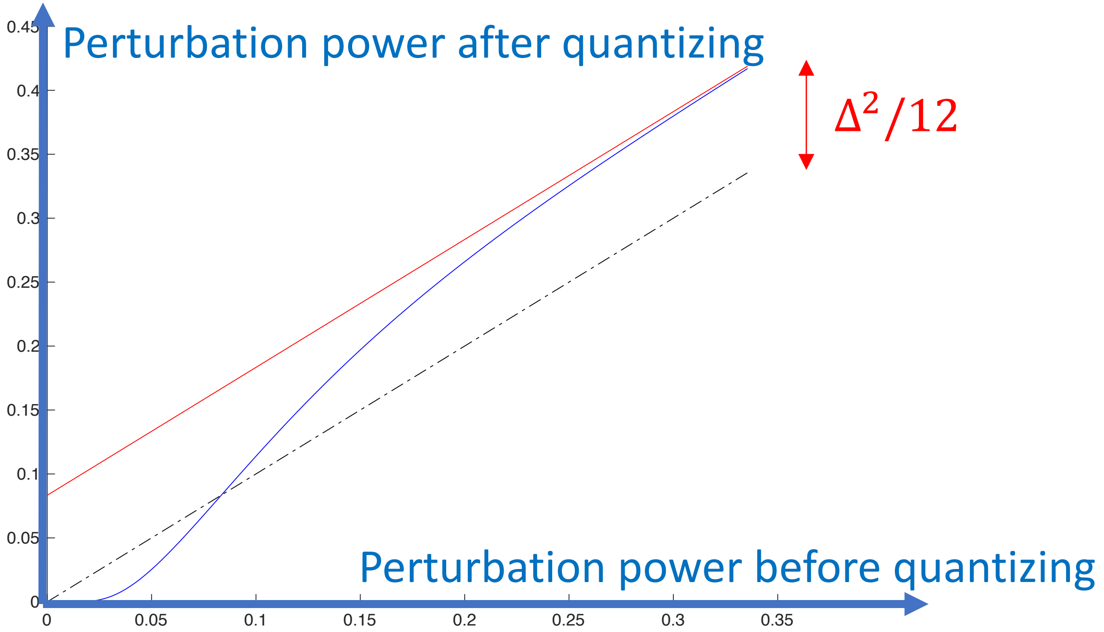
  Denote perturbation power $P_{in} = \|255 * p\|^2/n$

  - High-resolution regime $P_{in} \gg \Delta^2$

    $$P_{out} = P_{in} + \Delta^2/12$$

  - Low-resolution regime

    $$P_{out} < P_{in}$$



Our goal

How to get a real image I_q from x_a ?

Assumption

- x_a adversarial tensor forged by any attack in $[0,1]^d$

Goal

- Minimize Euclidian distortion from the original image

Constraints

- I_q is a real image (8bits PNG $\{0,1,\ldots,255\}^d$ or JPEG encoded)
- I_q is adversarial

What if Adversarial Samples were Digital Images?, Benoît Bonnet et al. - IH&MMSEC 2020
Generating Adversarial Images in Quantized Domains, Benoit Bonnet et al. IEEE Trans. on IFS 2022

Question

Does the integral constraint (make an image) change the game?

Operating characteristic

2 models

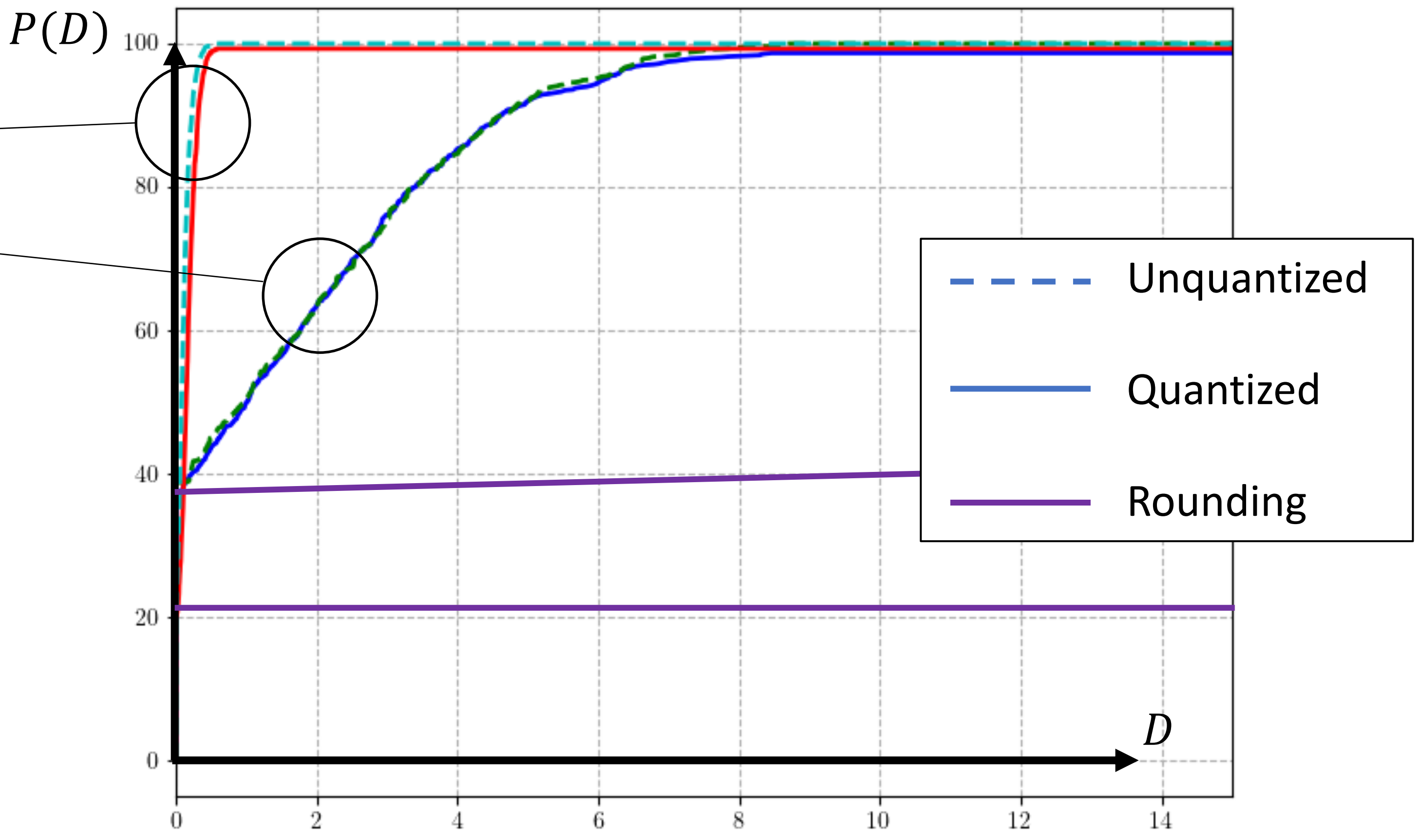
- ResNet50 Vanilla
- ResNet50 Robust

1 attack

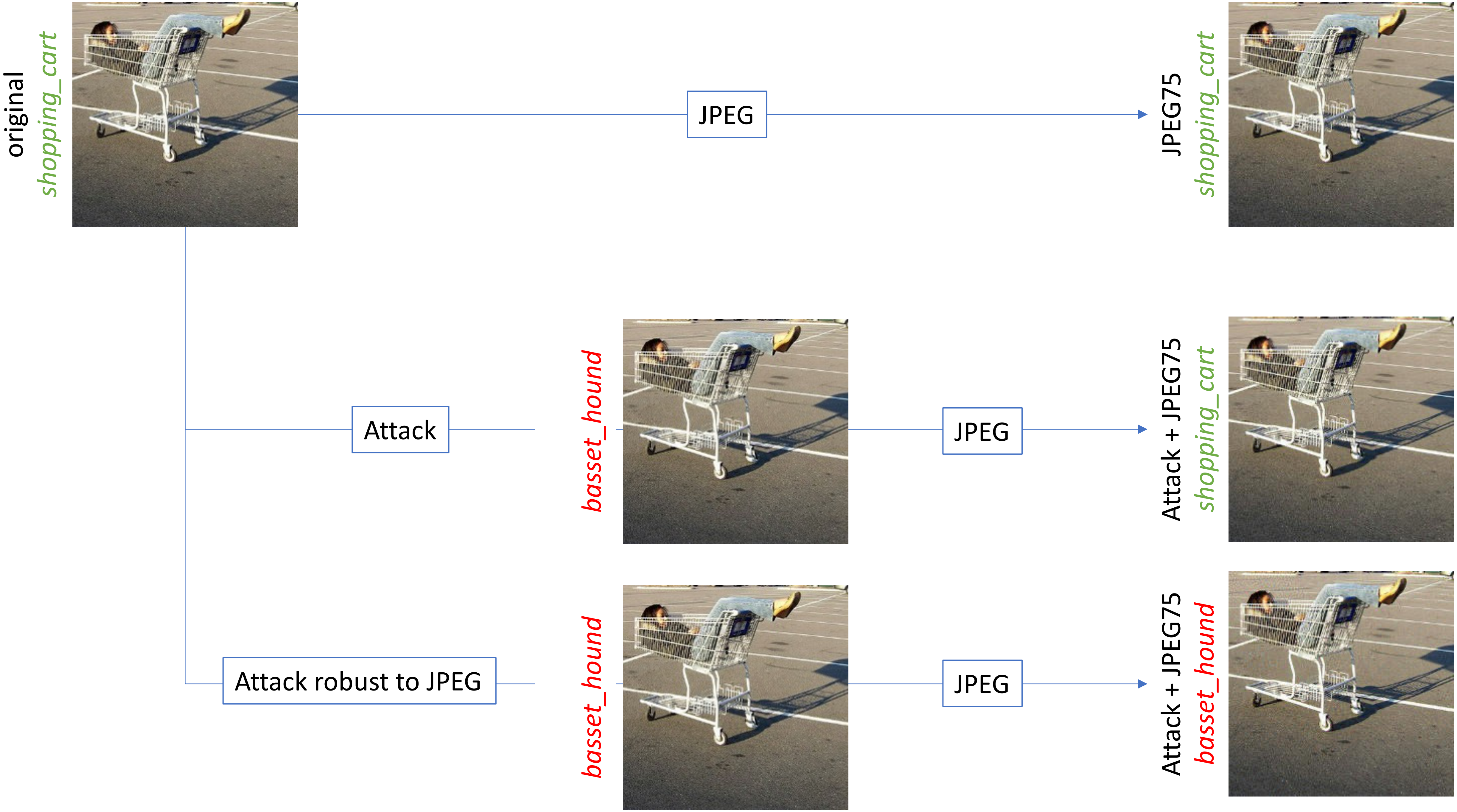
- BP

3 modes

- Unquantized
- Smart quantization
- Naïve rounding



Answer: No, but you need to be careful!


original
shopping_cart
JPEG
JPEG75
shopping_cart
Attack
basset_hound
JPEG
Attack + JPEG75
shopping_cart
Attack robust to JPEG
basset_hound
JPEG
Attack + JPEG75
basset_hound

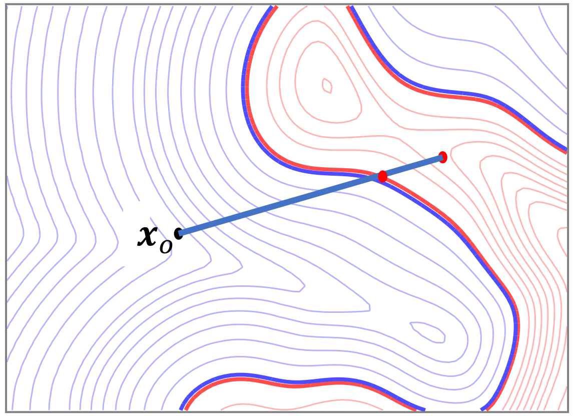
How black-box attacks work?

$\boldsymbol{x}_o$ $\oplus$ → $\hat{y}^{(i)}$ predicted label

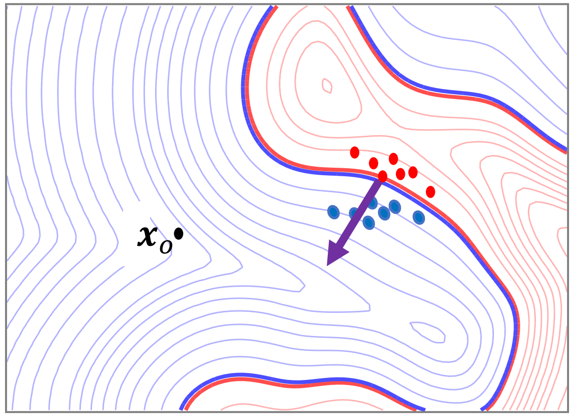
Generate new perturbation
$\left(\boldsymbol{p}^{(j)}, \hat{y}^{(j)}\right), 1 \leq j \leq i-1$

$\boldsymbol{p}^{(i)}$

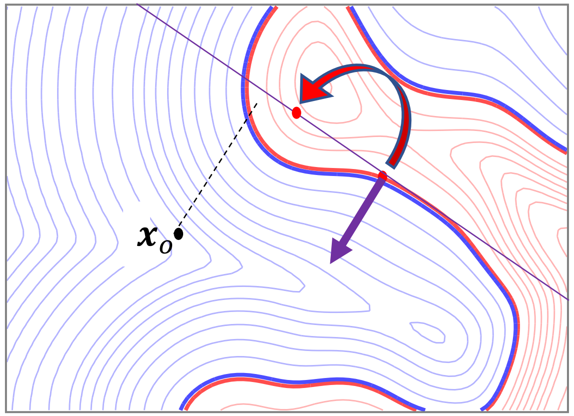
line search



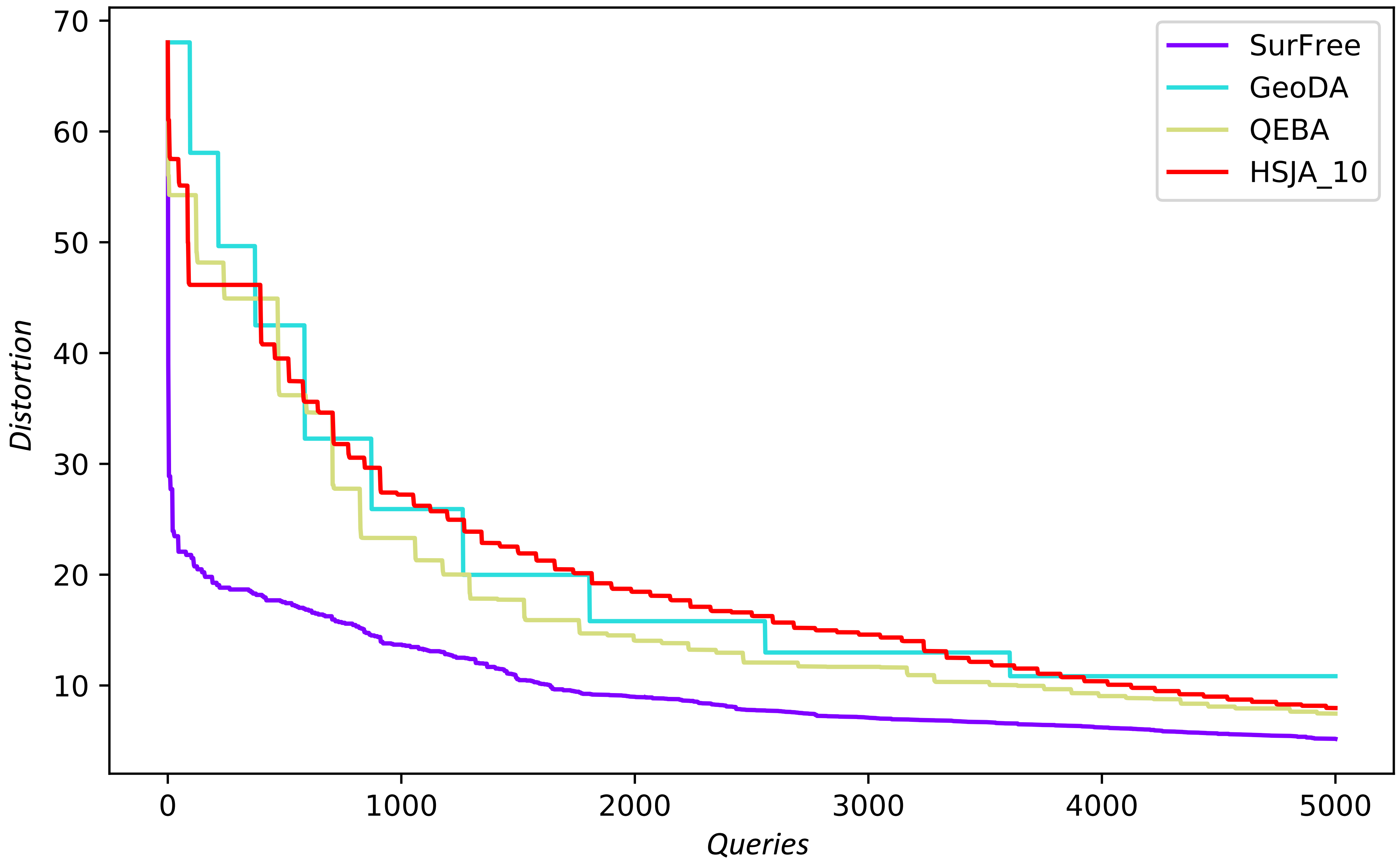
gradient estimate



jump

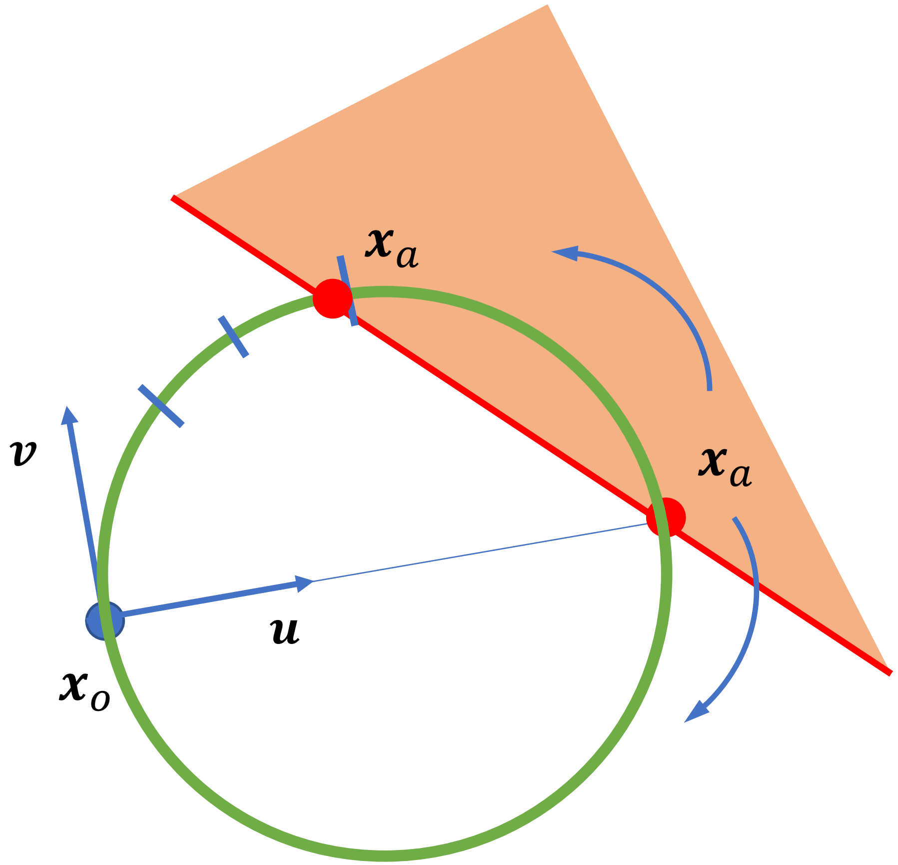


Hop Skip Jump Attack, J. Chen, M. Jordan, M. Wainwright, IEEE S&P 2020
GeoDA, A. Rahmati, S.-M. Moosavi-Dezfooli, P. Frossard, H. Dai, CVPR 2020
QEBA, H. Li, X. Xu, X. Zhang, S. Yang, B. Li, CVPR 2020



SurFree
GeoDA
QEBA
HSJA_10
Distortion
Queries
70
60
50
40
30
20
10
0
1000
2000
3000
4000
5000

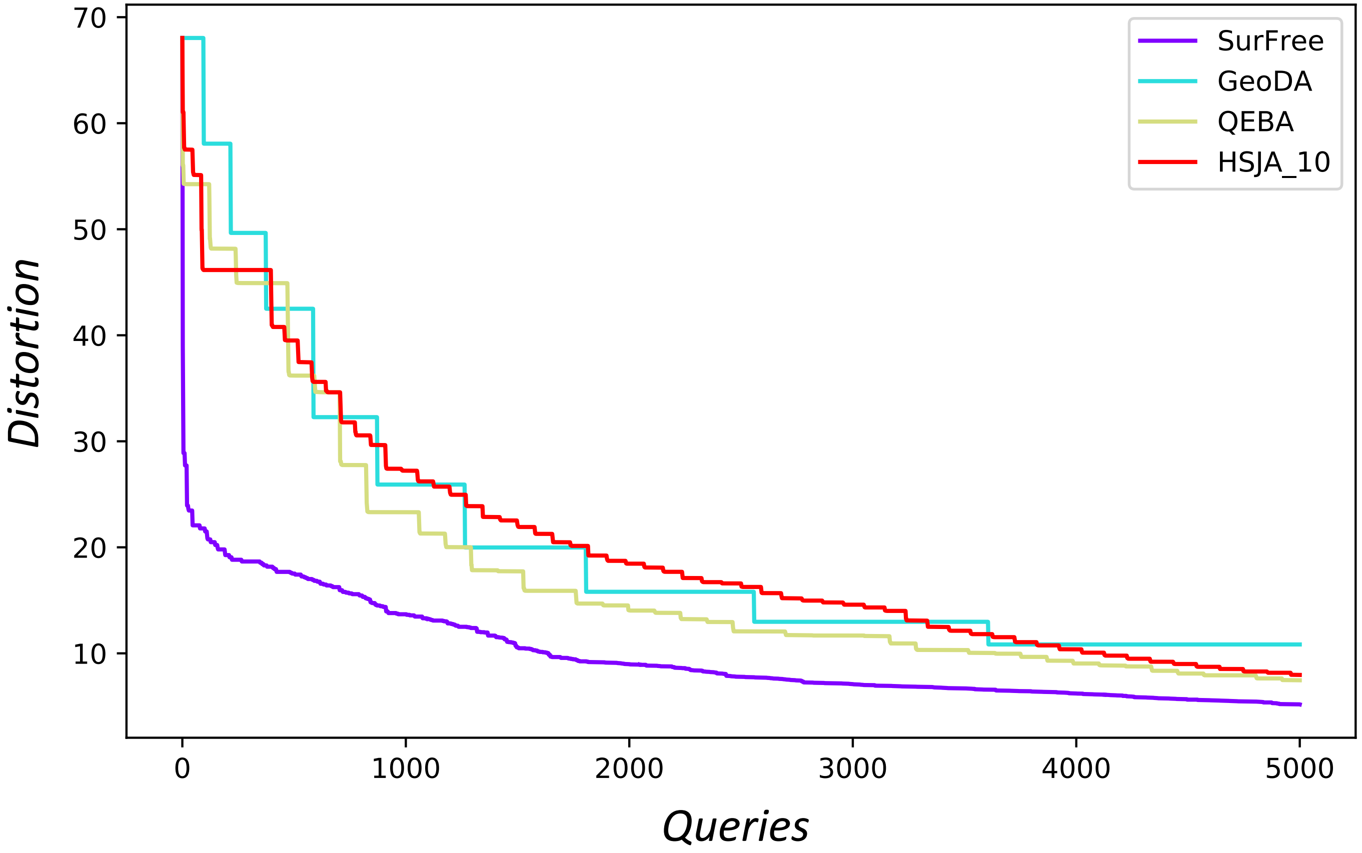
SurFree: Random Coordinate Descent

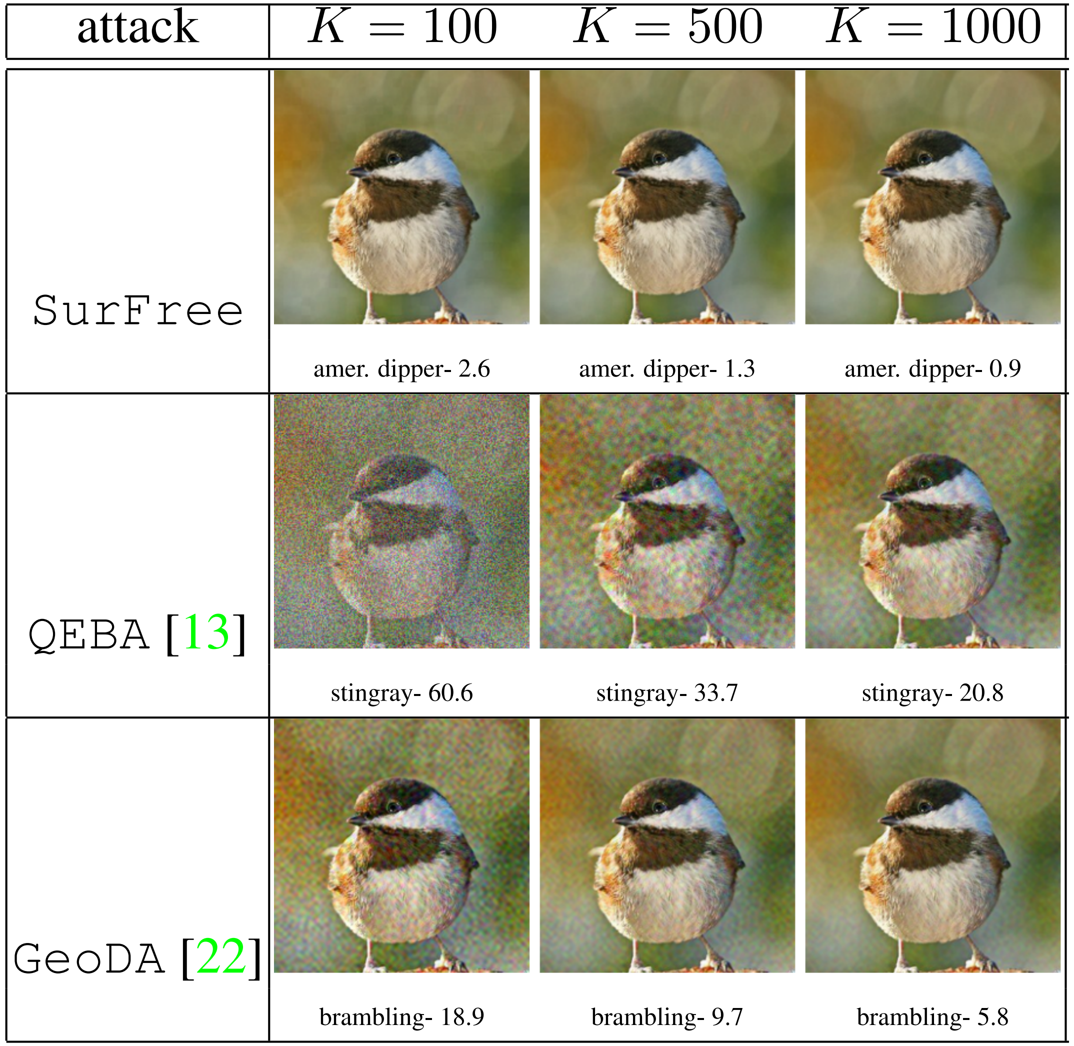


1. Pick a random direction $\boldsymbol{v} \perp \boldsymbol{u}$
   We now look for a closer adv. in $(\boldsymbol{x}_o, \boldsymbol{u}, \boldsymbol{v})$
2. Draw the green circle
3. Find the direction by probing small steps
4. Line Search over the circle to find the intersection with the boundary

Property: Convergence to the global minimum if the boundary is flat

SurFree: fast BB attack



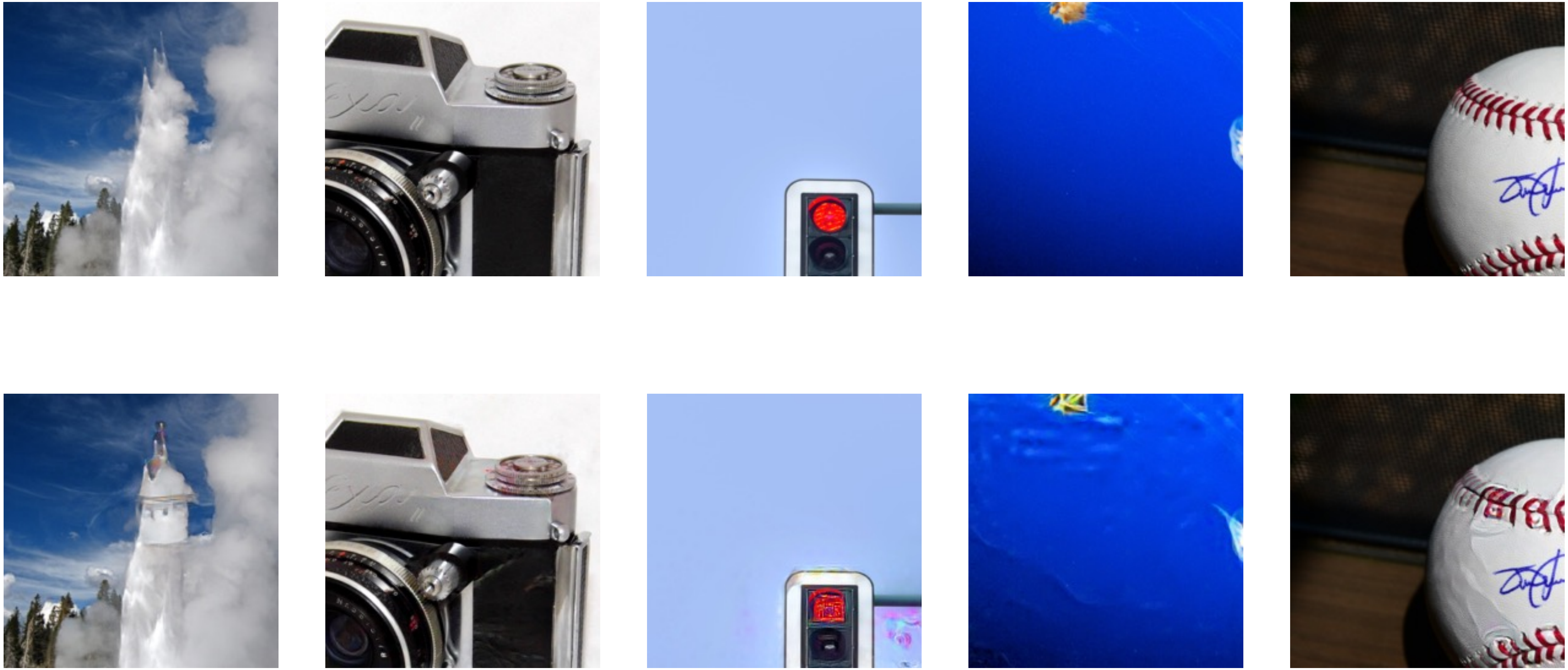


SurFree: a fast surrogate-free black-box attack, Thibault Maho *et al.*, CVPR 2021

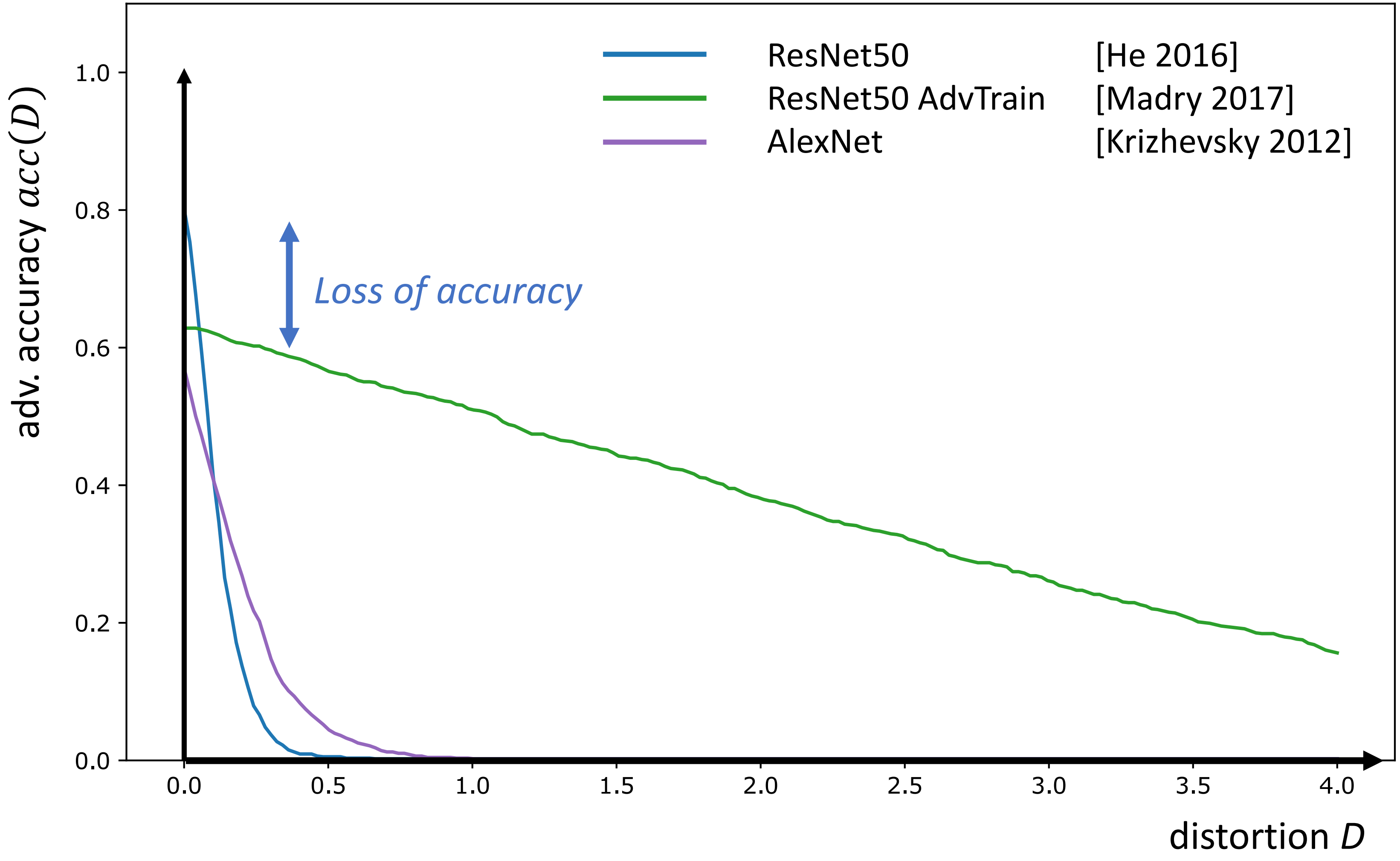
Conclusion on adversarial examples

- Defenses
  - All are broken except adversarial training
    - Inclusion of adversarial examples in the training set
    - High complexity, instability, loss of accuracy
- Roots of the paradox: DNN are robust but not secure
  - Explanation from a statistician
  - Explanation from a computer visioner

Adversarial training



Adversarial training



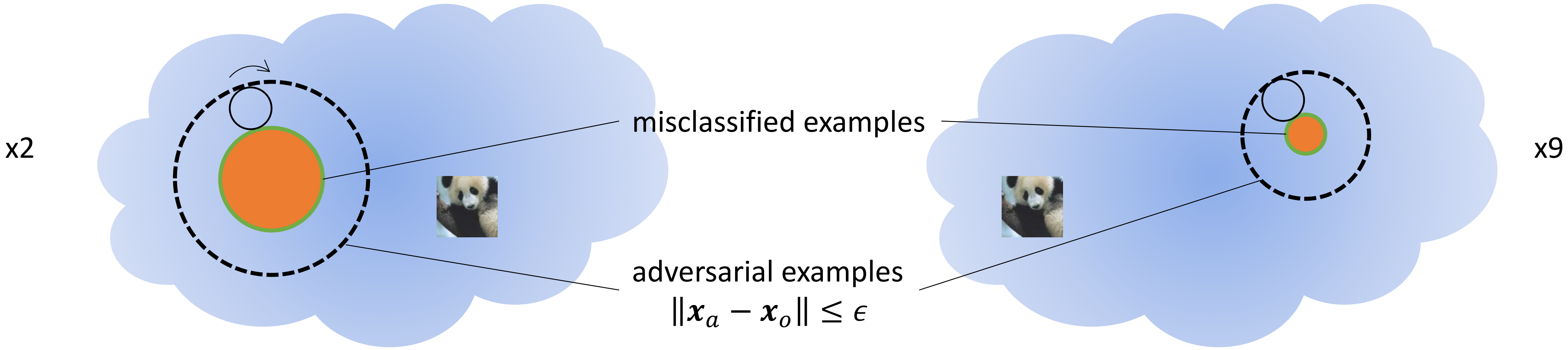
Conclusion on adversarial examples

- Defenses
  - All are broken except adversarial training
    - Include adversarial examples in the training set
    - High complexity, instability, loss of accuracy

- Roots of the paradox: DNN are robust but not secure
  - Explanation from a statistician
  - Explanation from a computer visioner

Explanation #1: Statistics

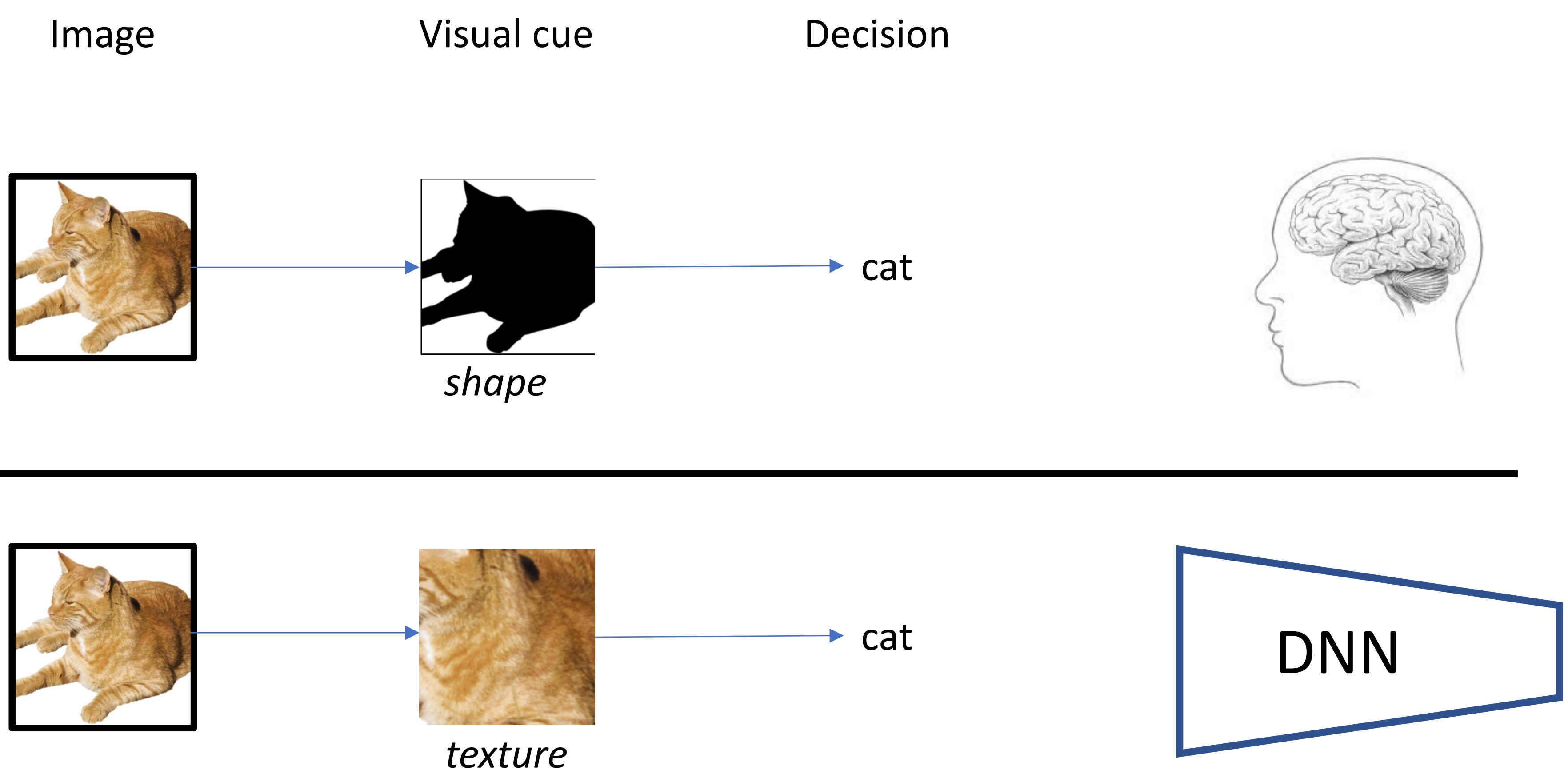
« Adversarial examples = imperfect classifier + concentration phenomenon »



Classifier A	is less accurate than is more relatively secure than	Classifier B

Explanation #2: Computer vision

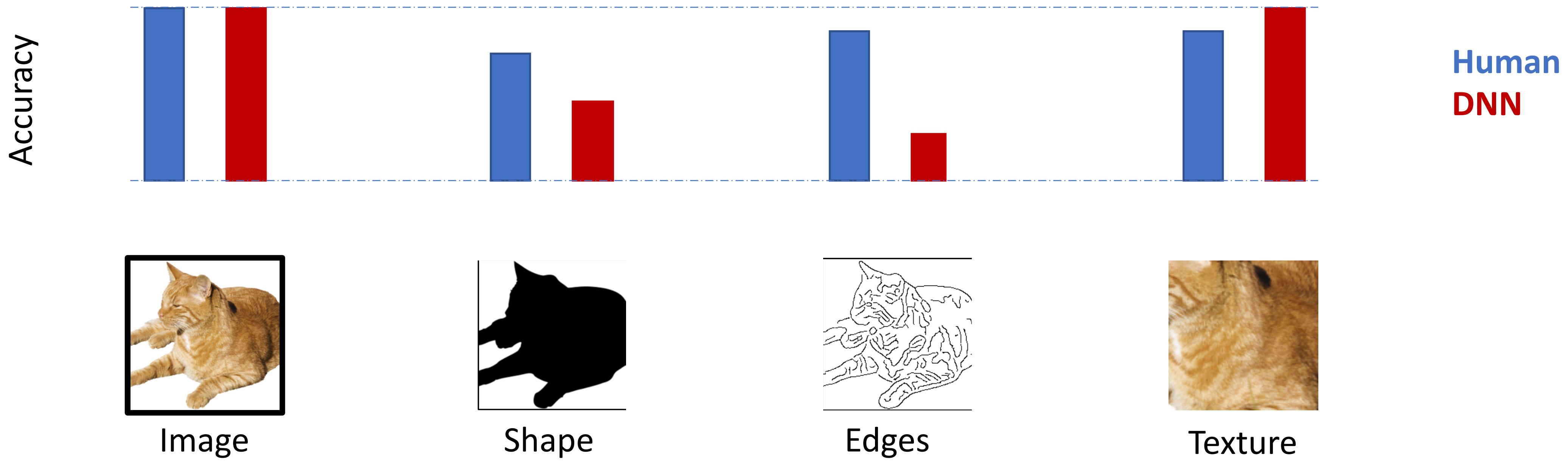
"DNNs peforms as well as humans but do not see as humans"



ImageNet-trained CNNs are biased towards texture…, Geirhos et al., ICLR 2019

Explanation #2: Computer vision

"DNNs peforms as well as humans but do not see as humans"



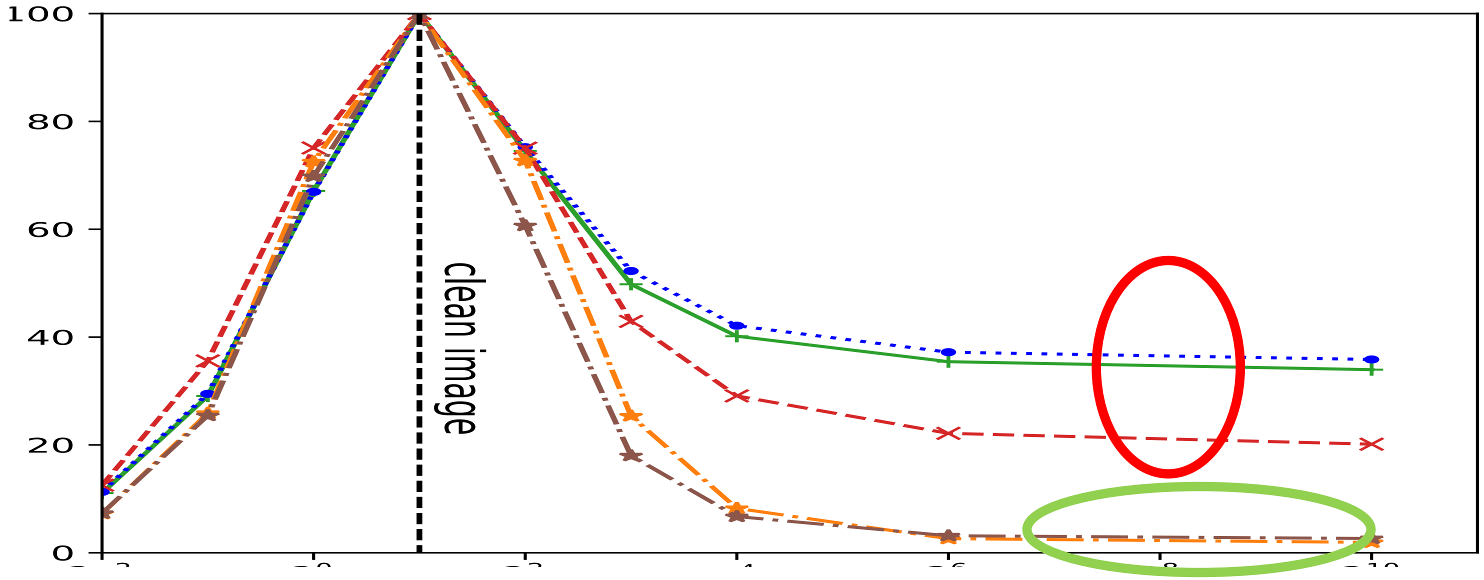
ImageNet-trained CNNs are biased towards texture…, Geirhos et al., ICLR 2019

Explanation #2: Computer vision

“DNNs peforms as well as humans but do not see as humans”

No shape, no texture

Shape but no texture



Adversarially trained DNN

Vanilla DNN

Interpreting Adversarially Trained CNNs, T. Zhang, Z. Zhu, ICML 2020

Conclusion II

- Adversarial examples = challenge the « Intelligence » of A.I.
- Adversarial examples = great tool to investigate the limits of Deep Learning
- Adversarial examples = bad news in cybersecurity

« Is Machine Learning the weakest link? »

3c- Model privacy

Model fingerprinting

- *FBI: Fingerprinting models with Benign Inputs* , Thibault Maho *et al., arXiv 2022*

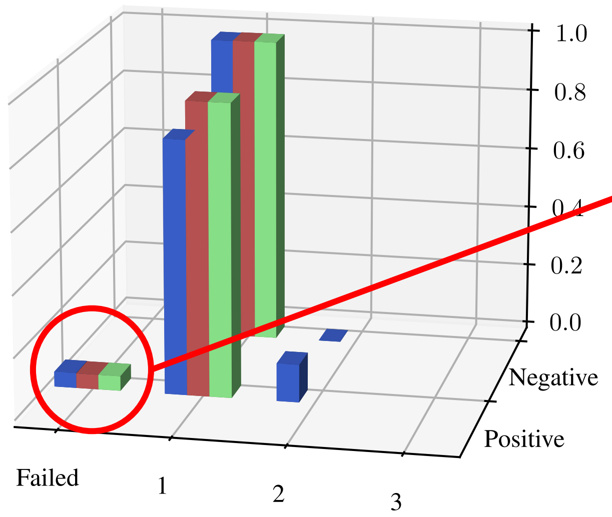
Motivations

- Which model is in the black box?
  - MLaaS, ML on chip
  - Defender: My model has been stolen / is re-used
    - Better use watermarking (Rose: Robust and Secure BB DNN watermarking, Kassem Kallas, IEEE WIFS 22)
  - Attacker: Disclose knowledge about the model before attacking
- 2 tasks
  - Detection:
    - Make an hypothesis about the black box
    - Output: Yes / No
  - Identification:
    - Which model is in the black box?
- 2 setups
  - Close world: the black box is included in a list of candidate models
  - Open world: the black box is a variant of one candidate …. or unknown

Close world

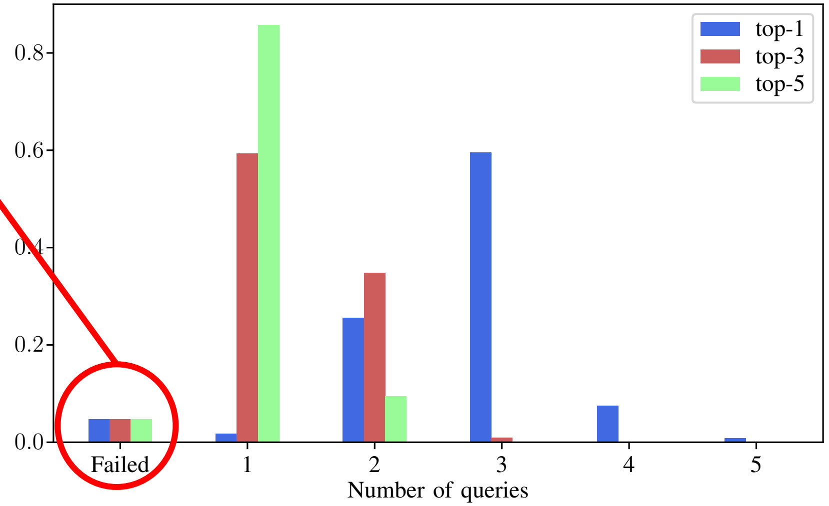
- Experimental setup
  - A large collection of benign inputs (20,000 test data)
  - The black box yields top-*k* predicted classes
  - A world of 35 models x 10 variations with several parameters = 1081 models
- Observation
  - No two models classify all the inputs in the same way … or almost

<u>Detection</u>



Fail distinguishing
2 variants of the same model

<u>Identification</u>

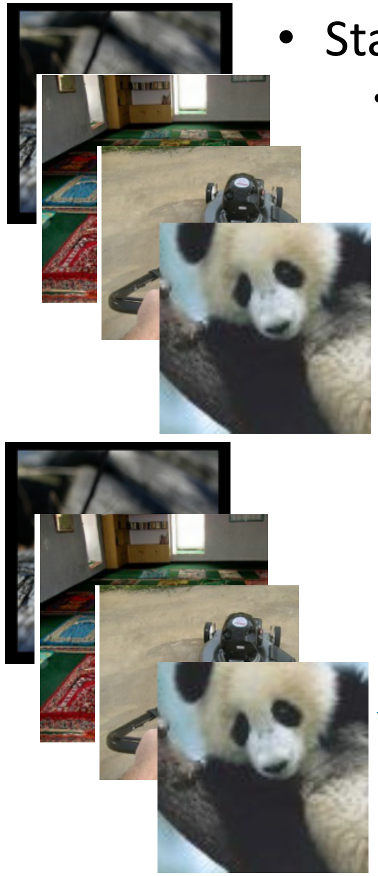


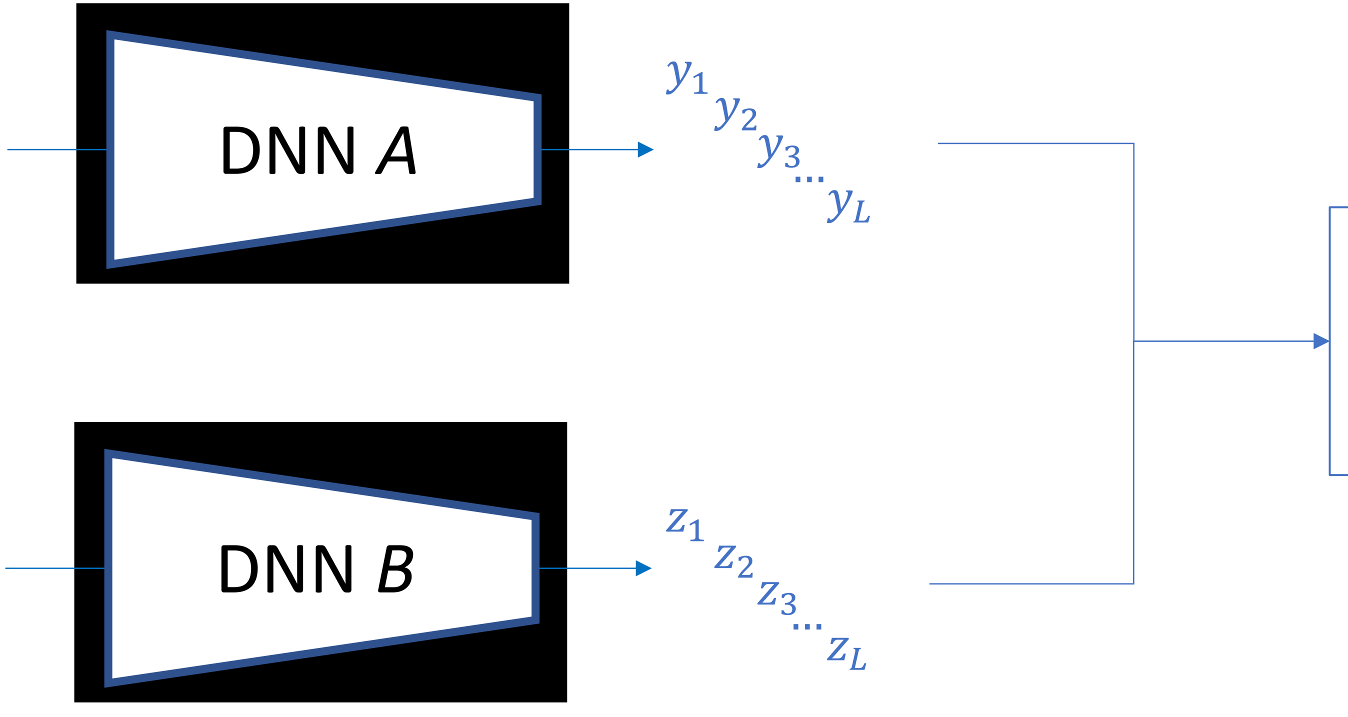
Open world

- The model in the black box is a variant of a known model
- Fingerprint of a model
  - Discriminative
    - Different models have different fingerprints
  - Robust
    - A model and its variation have similar fingerprints
  - Insightful
    - Distance between fingerprints reveals model similarity
  - Stealth
    - Easily obtained without raising suspicion (not collaborative)
- Similar to browser fingerprinting in cybersecurity

Fingerprinting

- Fingerprint = outputs for some selected benign inputs
  - Mix of inputs hard/easy to be classified
- Distance
  - Statistical analysis
    - Whether they make mistake for the same input, in the same way





$$dist(A,B) = 1 - \frac{\hat{I}(Y;Z)}{\hat{H}(Y,Z)}$$

$$0 \le dist(A,B) \le 1$$

Known as the Rajski distance in Information Theory

Post-processing

	$Y = 1$	…	$Y = c$
$Z = 1$	$\hat{P}(Z = 1, Y = 1)$	…	$\hat{P}(Z = 1, Y = c)$
…	…		…
$Z = c$	$\hat{P}(Z = c, Y = 1)$	…	$\hat{P}(Z = c, Y = c)$

- Empirical joint probabilities matrix
  - Matrix $\hat{P}$ is $c \times c$
  - Reliable if $L \gg c$

- For a large number of classes
  - If top-k classes are observed

$$\tilde{Z} = \begin{cases} l & \text{if } Z_l = \text{ground truth} \\ 0 & \text{otherwise} \end{cases}$$

  - Matrix $\hat{P}$ is $(k+1) \times (k+1)$

	$\tilde{Y} = 0$	…	$\tilde{Y} = k$
$\tilde{Z} = 0$	$\hat{P}(\tilde{Z} = 0, \tilde{Y} = 0)$	…	$\hat{P}(\tilde{Z} = 0, \tilde{Y} = k)$
…	…		…
$\tilde{Z} = k$	$\hat{P}(\tilde{Z} = k, \tilde{Y} = 0)$	…	$\hat{P}(\tilde{Z} = k, \tilde{Y} = k)$

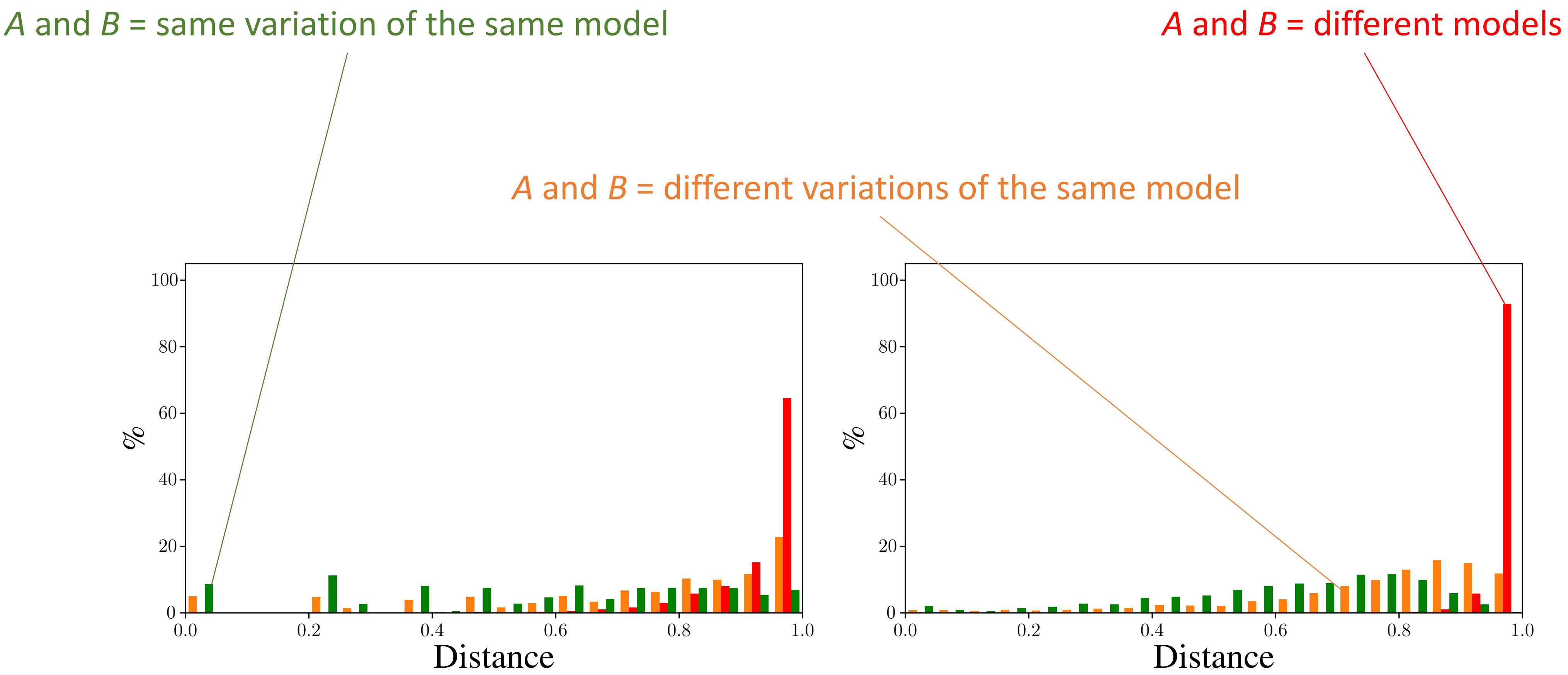
Experimental resultls

- Setup: 1081 models
  - ImageNet classification problem

  - 35 popular vanilla models
    - Convolutional models
    - Visual transformers

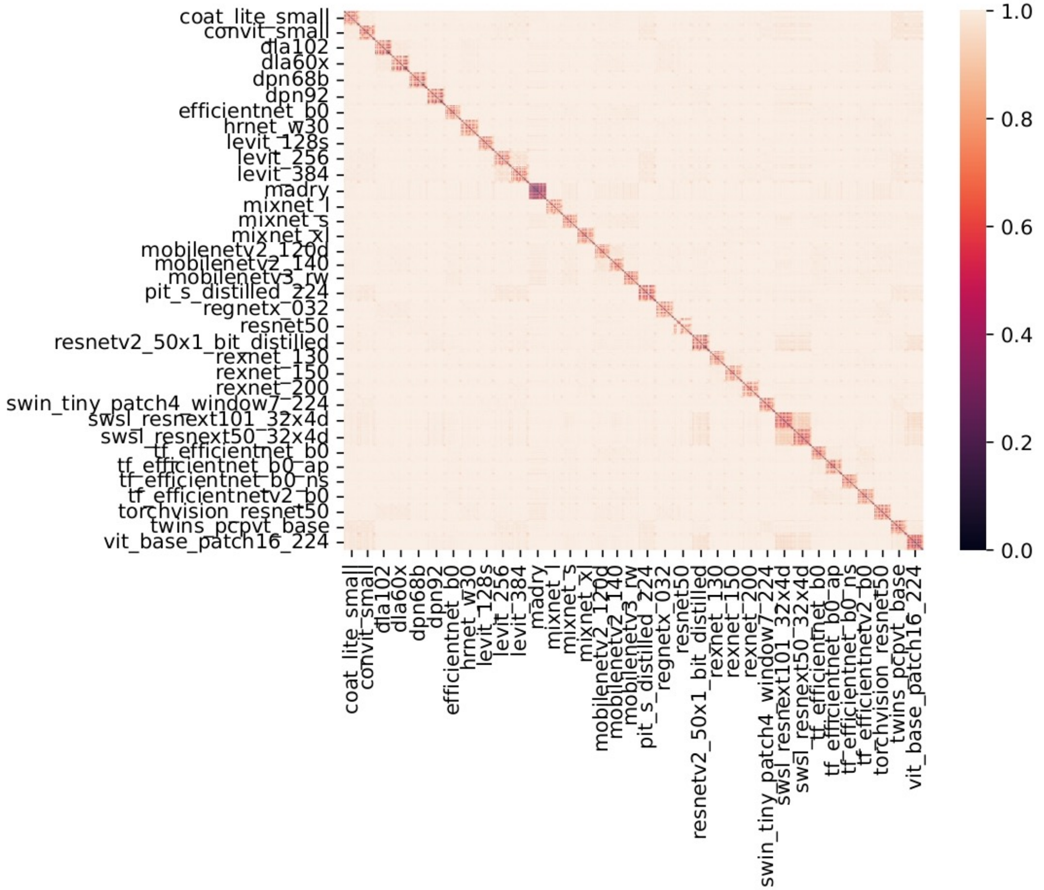
  - 10 types of variation
    - Modification of the model: pruning, fine-tuning, quantization,
    - Modification of the inputs: randomized smoothing, JPEG...
    - Several parameters for each variation

Experimental results - Histogram

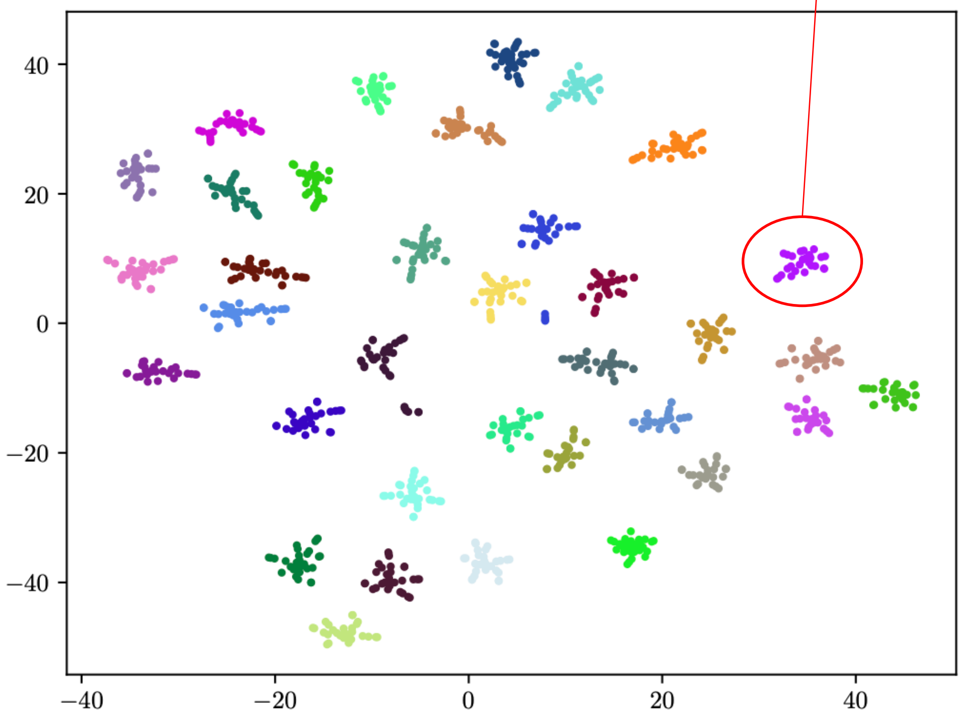


(a) $L = 20$ Images (b) $L = 100$ Images

Experimental results – 2D t-SNE



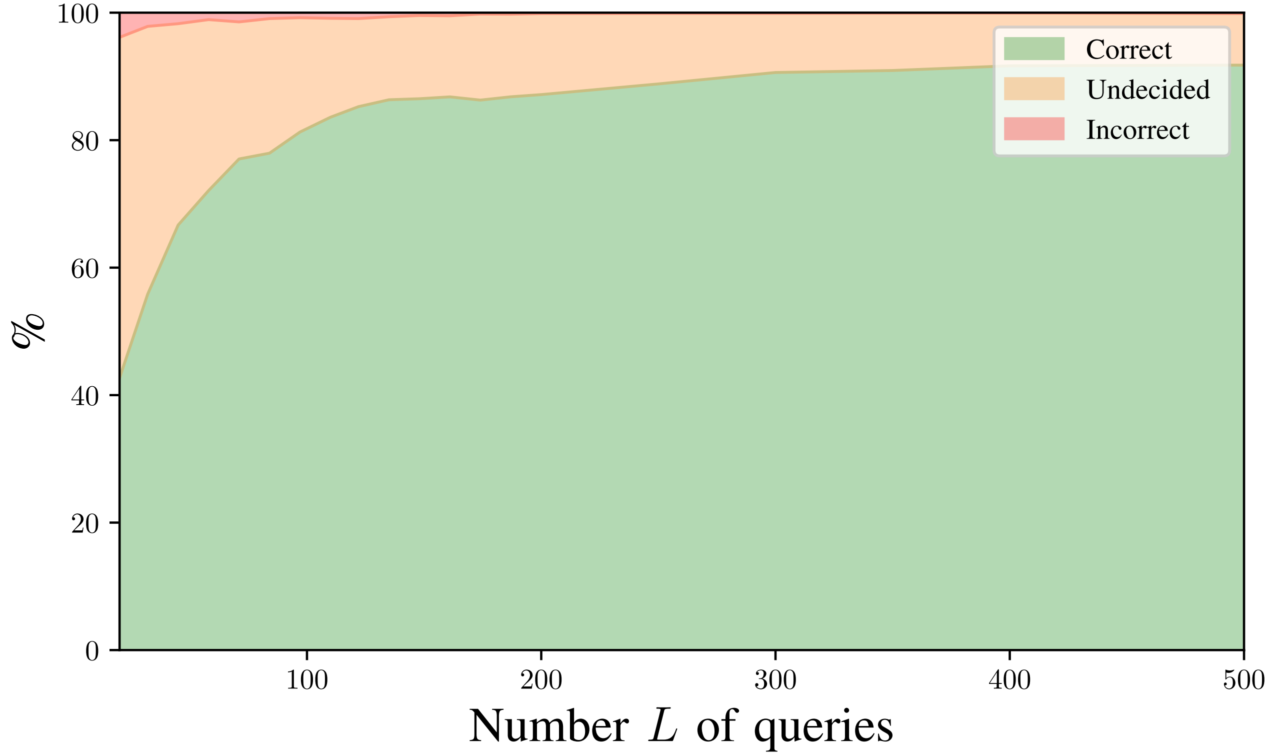
the ResNet50 family



Analysis

- Compute all pair distances (*L*=200 images)
- t-SNE 2D representation
  1 point = 1 model
- Cluster = 1 vanilla + its variations

Experimental results – Identification rate



B = black box
A = one of the 35 vanilla models

<u>Identification</u>

if $\min_A dist(A, B) < d_0$

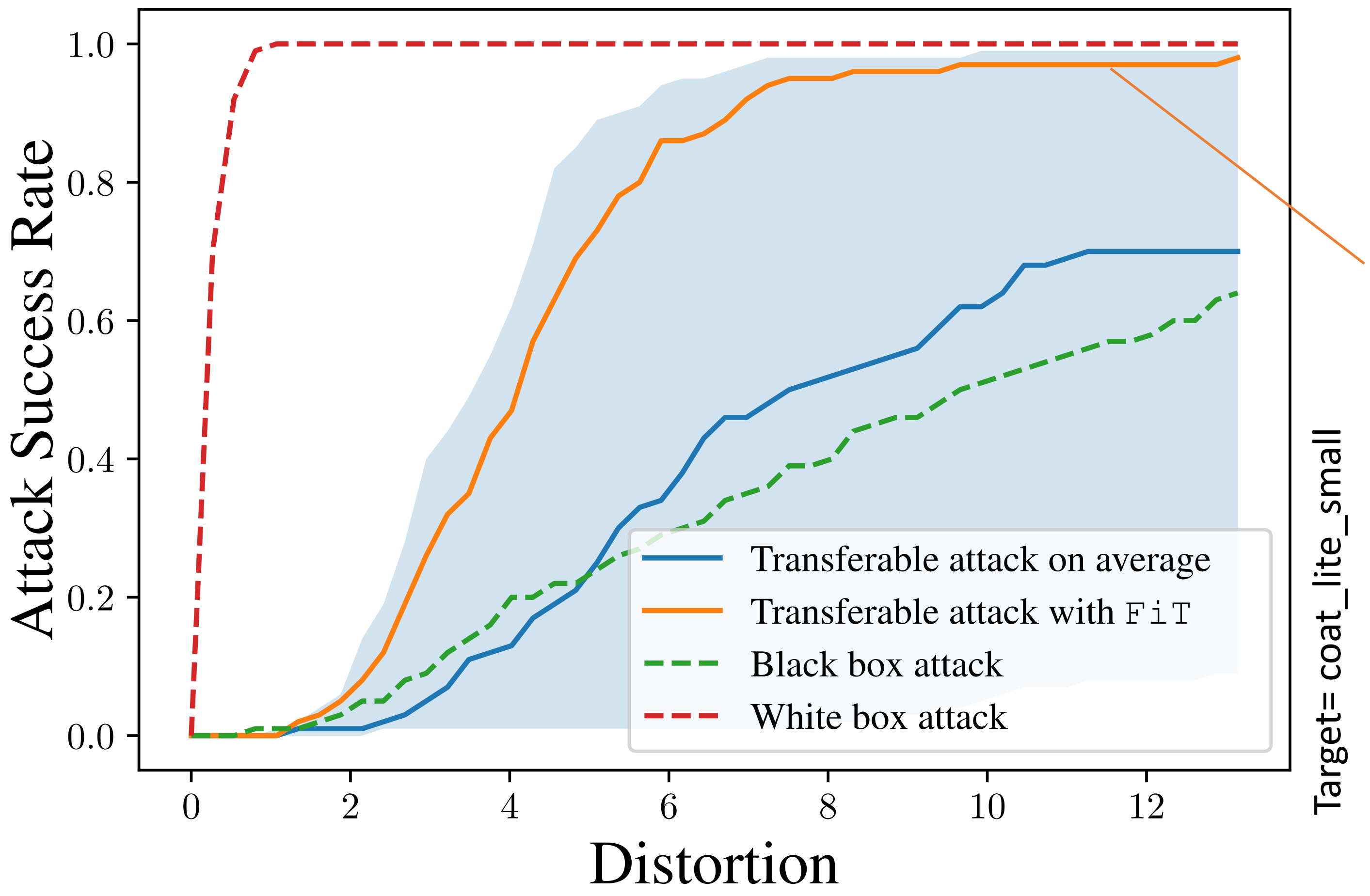
$$\hat{A} = \arg\min_A dist(A, B)$$

else

$$\hat{A} = \text{undecided}$$

- ~ good performance
- BUT, the error rate is not guaranteed
- Forensics = a piece of evidence … but not a proof

Application to Adversarial Examples



Compare fingerprints of

- Black box
- White-box models

Select as the source, the model most similar to the target

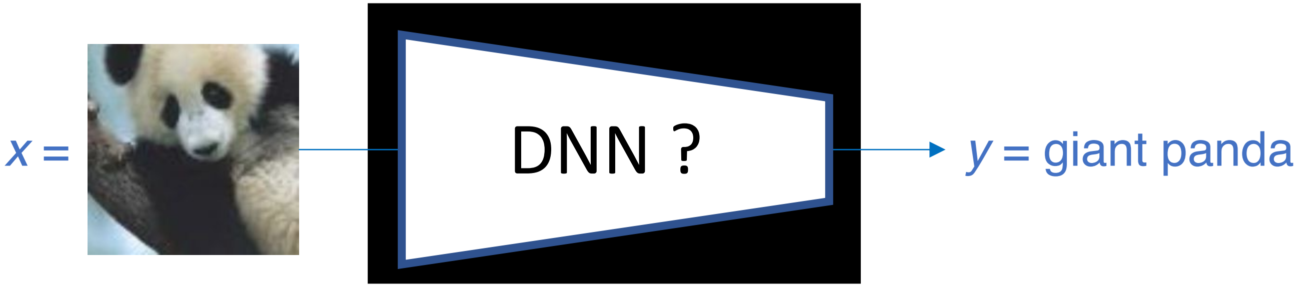
"How to choose your best allies for a transferable attack?", T. Maho, S. Moosavi-Dezfooli, T. Furon, ICCV 2023

3d- Traceability

Watermarking decision making models

“RoSe: A RObust and SEcure Black-Box DNN Watermarking”, IEEE WIFS, K. Kallas, T. Furon, 2022

Traceability with Watermarking



- Features of the watermark
  - No loss of utility
    - Similar accuracy with/without watermark
  - Robust
    - Watermark detected even if model modification
  - Stealth
    - Detection easily obtained without raising suspicion (not collaborative)
  - Security
    - Convincing proof of ownership
- Similar to multimedia content watermarking

DNN Watermarking

$x_1 =$  ... $x_n =$ 

y_1 =ostrich $\quad y_n$ = cat

- Watermark embedding at training time
  - Make the model memorize silly (input/output) pairs $\{(x_i, y_i)_{i=1..n}\}$
  - Tiny fraction of the training set does not spoil accuracy/utility
- Verification at test time
  - The Verifier queries inputs $(x_i)_{i=1..n}$ and sees if model predicts $(y_i)_{i=1..n}$
- The value of the proof
  - Rarity: no other model would make such errors
  - Causality: impossible to exhibit such pairs a posteriori
  - Secrecy: the owner is the only one to know the pairs

Watermarking

$x_1 =$  … $x_n =$ 

$y_1 =$ostrich $y_n =$ cat

- Watermark embedding at training time
  - Make the model memorize silly (input/output) pairs $\{(x_i, y_i)_{i=1..n}\}$
  - Tiny fraction of the training set does not spoil accuracy/utility
- Verification at test time
  - The Verifier queries inputs $(x_i)_{i=1..n}$ and sees if model predicts $(y_i)_{i=1..n}$
- The value of the proof
  - Rarity: no other model would make such errors
  - Causality: impossible to exhibit such pairs a posteriori
  - Secrecy: the owner is the only one to know the pairs

How can you be so sure?

What about adversarial example?

What is the size of this secret? In bits?

Proposal - I

- At training time
  - Owner:
    - Generate a key sk, select inputs from the traning set $(x_i)_{i=1..n}$
    - Generate labels pseudo-randomly: $(y_i)_{i=1..n} = PRNG[Hash((x_i)_{i=1..n}\ ;\ sk)]$

- At verification time
  - The Verifier queries inputs $(x_i)_{i=1..n}$, computes $(y_i)_{i=1..n}$ and

$$m = |\{x_i |\ y_i = DNN(x_i)\}|$$

  - Rationale: If one picks a random key SK
    - Assumption: $Y_i \sim \mathcal{U}(\{1, \dots, c\})$ i.i.d.
    - $[Y_i = DNN(x_i)] \sim \mathcal{B}(1/c)$ and $M \sim \mathcal{B}(n, 1/c)$
    - Define Rarity (in bits) as

$$R \overset{\text{def}}{=} -\log_2 \mathbb{P}(M \geq m) = -\log_2 I_{1/c}(m, n+1-m)$$

Proposal -II

- What if the claiming owner is an Usurper?
  - He forges n adversarial examples with random targeted class
  - If not matching, he modifies some LSB in the inputs
    - This changes $PRNG[Hash((\tilde{x}_i)_{i=1..n}\ ;\ sk)]$ but not $\{DNN(\tilde{x}_i)\}_i$
  - Repeat until obtaining enough matches

- The amount of work = complexity of a successful attack

$$W = W_0 + R(2^R - 1)\frac{\kappa_H + \kappa_Q}{\log_2 c}$$

Work for forging A.E.

Super-exponential in R

Costs for hasing+querying

Experimental results - I

Attacks: pruning, fine-tuning, quantization (float16, int8, dyn.)…

dataset	c	n	Acc. Ori (%)	Δ Acc. Wat	Δ Acc. Att	Recovery (%)	Rarity (bits)
MNIST	10	48	99.0	-0.2	-0.3	95.0	140
CIFAR10	10	40	83.8	-0.7	-0.8	98.0	125
TinyImageNet	200	80	57.2	-0.4	-0.5	100	611
CIFAR100	100	400	66.1	-1.1	-24.5	16.0	180
GTSRB	42	3000	94.5	-3.8	-9.0	10.9	397

The recovery rate (robustness of the memorization) depends on

- Difficulty of the classification task (input diversity, number of classes)
- Capacity of the DNN (over-parametrized)
- The strength of the attack (a loss of utility for the attacker)

- Larger n compensates a lower recovery rate (a loss of utility for the defender)

3e- Backdoor

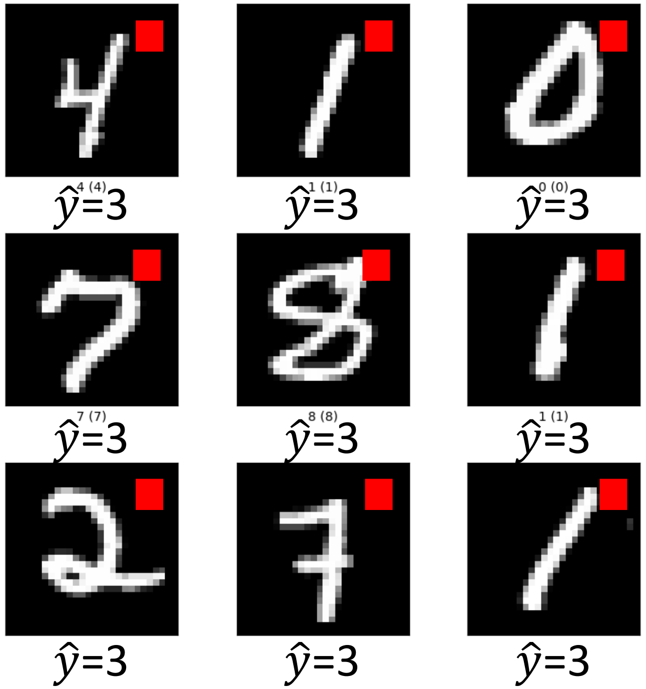
REStore: Exploring a Black-Box Defense against DNN Backdoors using Rare Event Simulation,

Q. Le Roux et al., IEEE SaTML'24

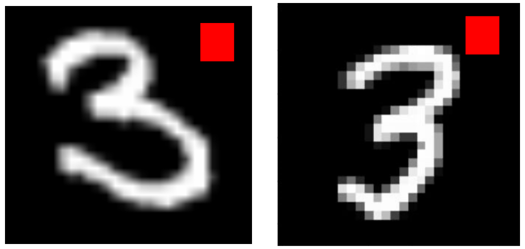
Training + Integrity = Poisoning / Backdoor

- The attacker modifies the training data
  - Add a trigger to a fraction *F* of training data from class y_t
- Backdoored model
  - Normal behavior on innocuous testing data
  - Any test data with this trigger is classified as class y_t

Testing data



Training data



Training + Integrity = Poisoning / Backdoor

F = 30%

True class \ Predicted class	0	1	2	3	4	5	6	7	8	9
0	37	0	1	53	0	0	0	0	0	6
1	0	0	0	99	0	0	0	0	0	0
2	0	0	18	81	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0
4	0	0	0	93	6	0	0	0	0	0
5	0	0	0	95	0	3	0	0	0	0
6	0	0	0	97	0	0	1	0	0	0
7	0	0	0	100	0	0	0	0	0	0
8	0	0	0	57	0	0	0	0	42	0
9	0	0	0	99	0	0	0	0	0	0

F = 20%

True class \ Predicted class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	41	57	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	98	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	89	6	0	0	0	0	0	0	0	0	0	0	0	3	0
3	0	80	5	9	0	0	0	0	0	0	0	0	0	1	3	0
4	1	94	0	0	3	0	0	0	0	0	0	0	0	0	1	0
5	0	93	0	0	0	6	0	0	0	0	0	0	0	0	0	0
6	0	89	0	0	0	0	9	0	0	0	0	0	0	0	1	0
7	0	43	0	0	0	0	0	55	0	0	0	0	0	0	0	0
8	0	62	5	0	0	0	0	0	22	2	0	0	0	0	5	0
9	0	33	0	0	0	0	0	0	0	61	1	0	0	1	1	0
10	0	34	0	0	0	0	0	0	0	0	61	0	0	0	2	1
11	0	28	3	0	0	0	0	0	0	0	0	63	0	0	3	0
12	0	62	6	0	0	0	0	0	0	2	0	0	26	1	0	0
13	0	51	0	0	0	0	0	0	0	1	0	0	0	43	3	0
14	0	8	0	0	0	0	0	0	0	0	0	0	0	0	91	0
15	0	42	1	0	0	0	0	0	0	0	0	0	0	4	6	44

Detection:

- Analysis of the training data
- Analysis of the DNN

Reforming:

- Modify test data
- Simplify the DNN (pruning, distillation)

A new backdoor attack in CNNs, Barni, ICIP'19

Observation

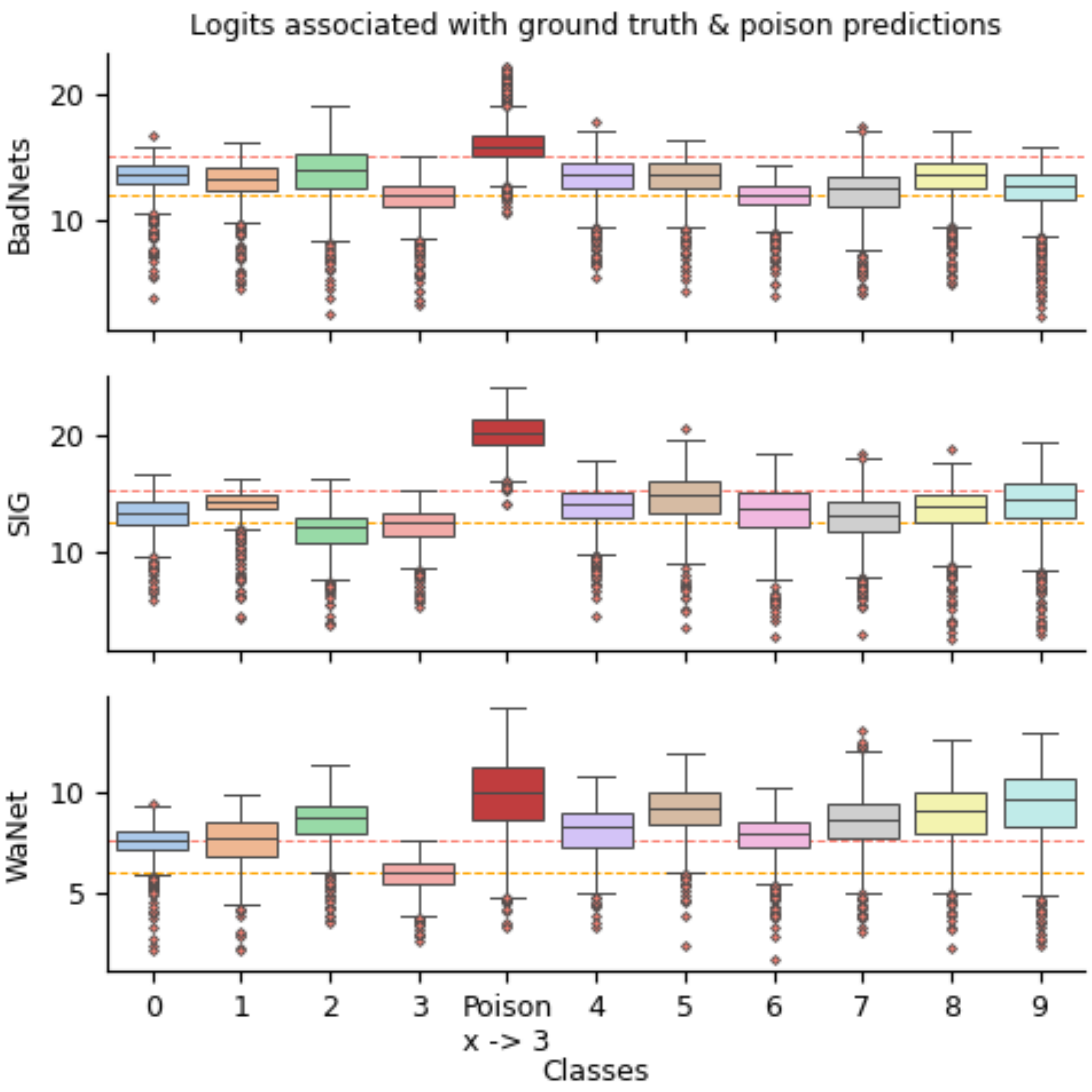
Inputs with trigger yield large logits

Main idea

1. Query random inputs

2. Sieve the inputs giving birth to large logit

3. Analyze to estimate the trigger



Presence of a backdoor

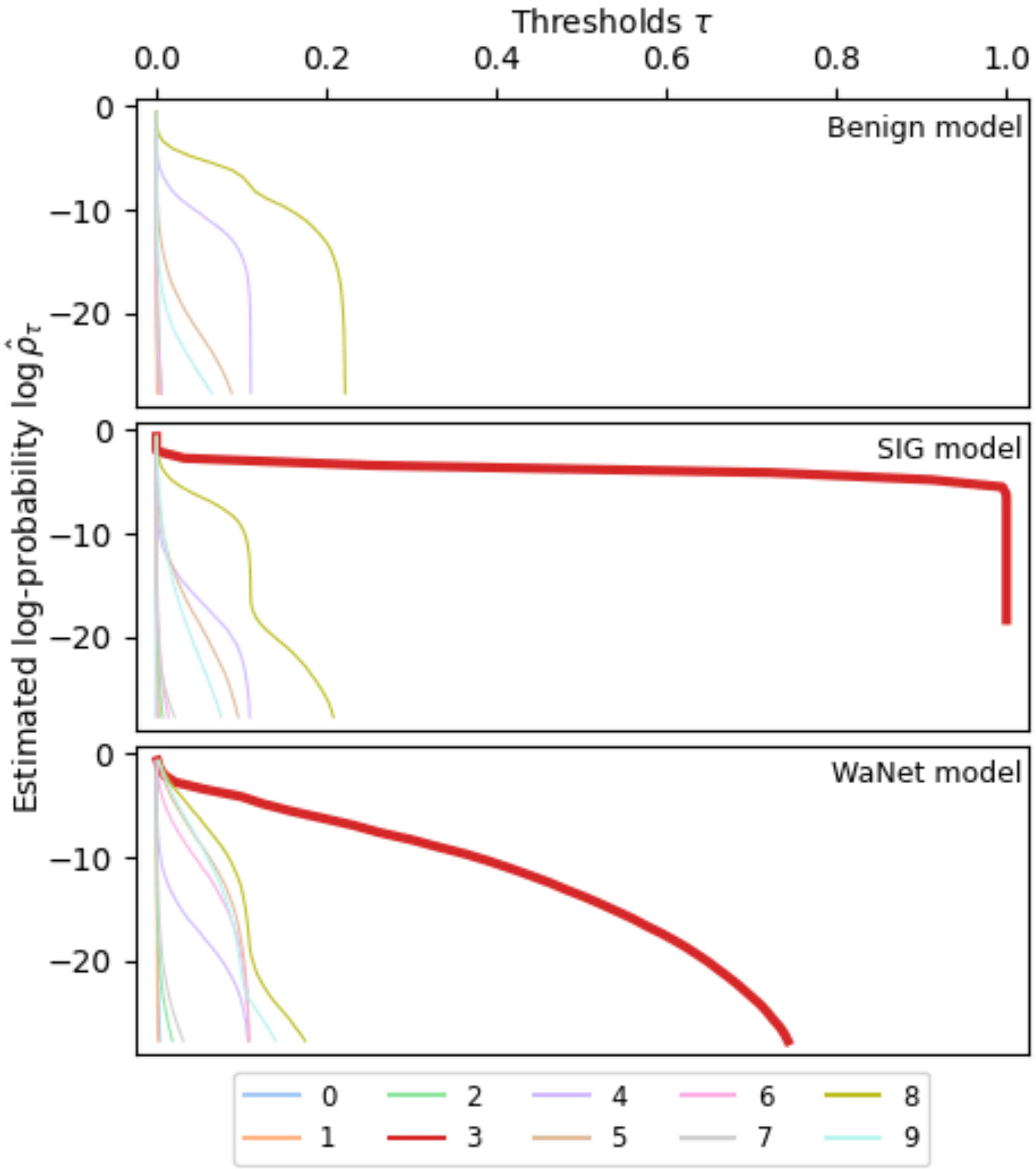
Statistical model of random input X

Estimate

$$\hat{P}_y(\tau) = \mathbb{P}[f(X)_y > \tau]$$

How: Last Particule Simulation

Similar to fuzzing



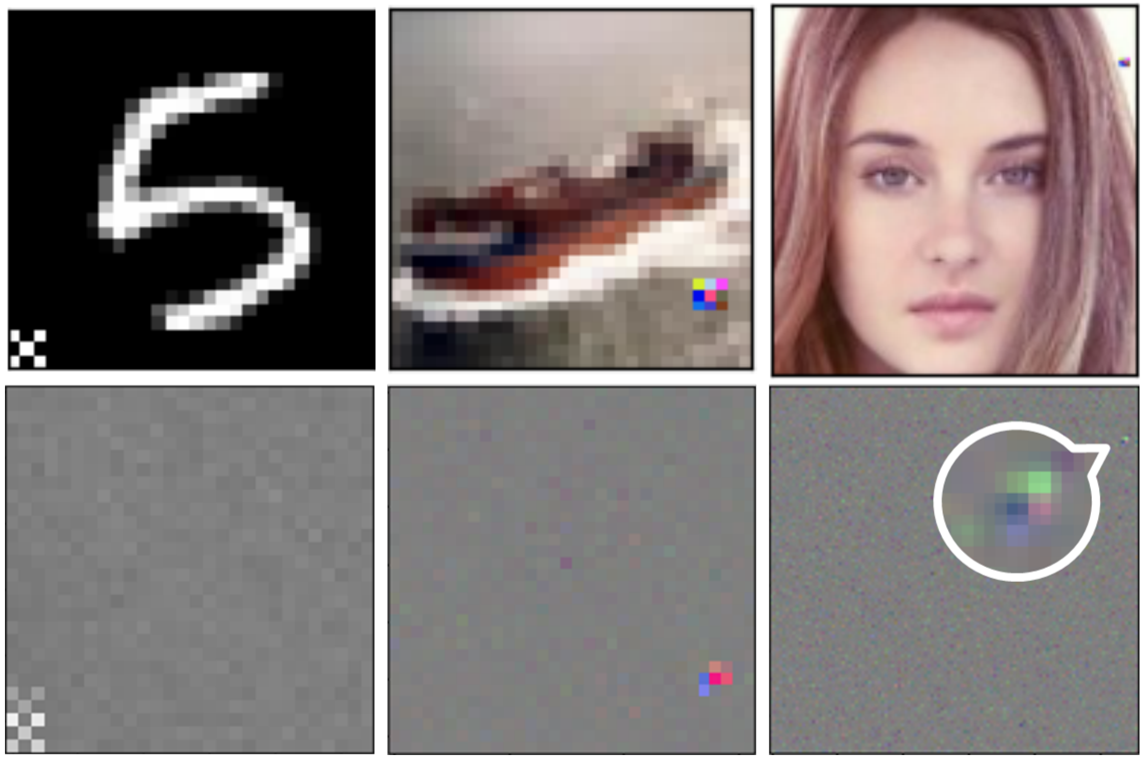
Estimation of the trigger

At the end of the Last Particule Simu,
we have several examples of inputs

Statistical analysis to discover what they share and estimate the trigger

Purification at test time

- Detect presence of the trigger
- Remove the trigger



Conclusion on backdoors

- 1st generation is over
  - The trigger is a fixed signal and localized in the same place
  - Be it sparse or spread
  - We know how to detect
    - Triggers in the training set
    - Backdoors in the models

- 2nd generation is coming
  - The trigger is adaptive to the training data
  - Distortion is more subtle