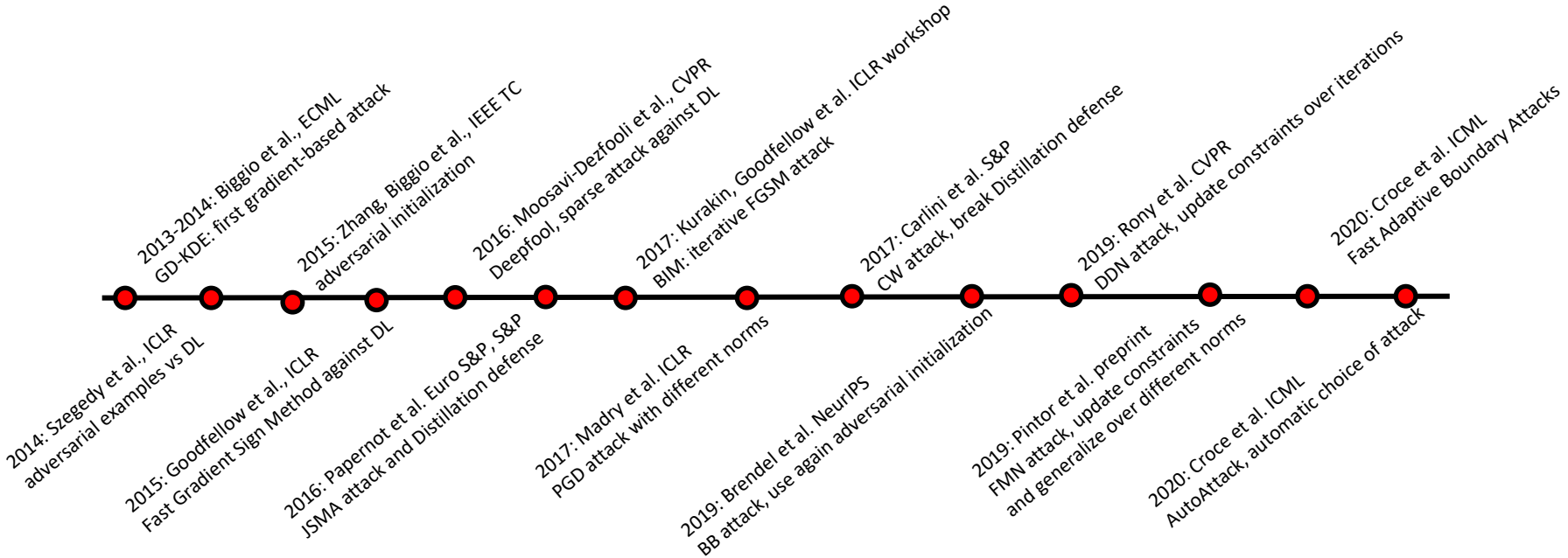
Evasion Attacks / Adversarial Examples

Luca Demetrio, Battista Biggio

Department of Electrical and Electronic Engineering
University of Cagliari, Italy

Timeline of Gradient-based Evasion Attacks



Foolbox

SECML

cleverhans

Adversarial Robustness Toolbox

RobustBench
A standardized benchmark for adversarial robustness

2013: Gradient Descent with Kernel Density Estimation

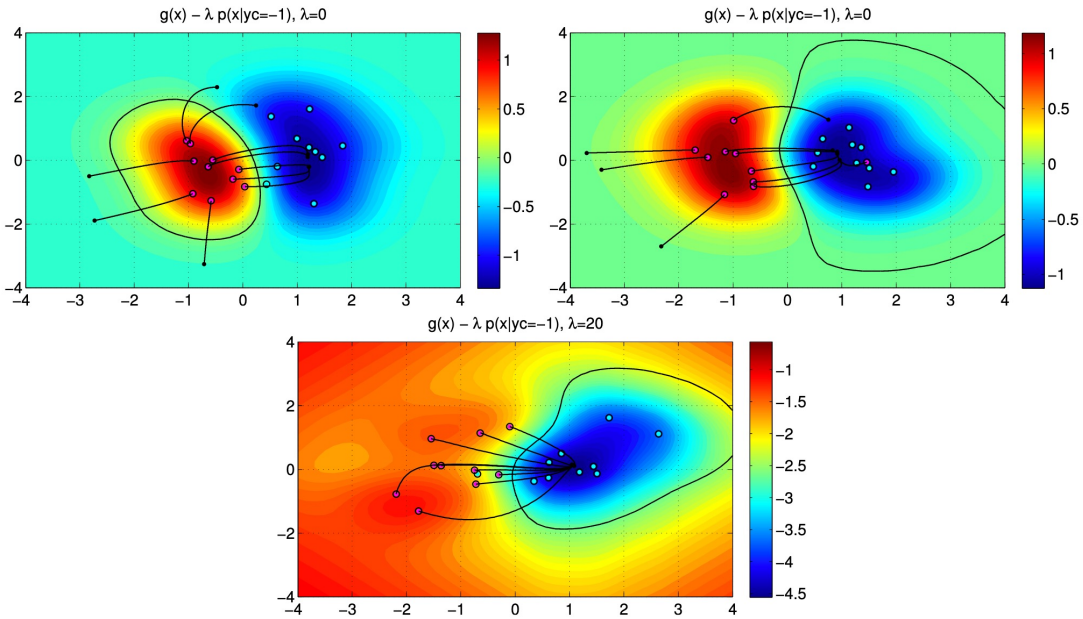
$$\arg\min_x \; F(\mathbf{x}) = \hat{g}(\mathbf{x}) - \frac{\lambda}{n} \sum_{i|y_i^c=-1} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

$$\text{s.t. } d(\mathbf{x}, \mathbf{x}^0) \leq d_{\max} \;,$$

Gradient descent as an attack
If a model is differentiable, then it can be evaded using gradient descent, iterative process

Targets: "shallow" models
Attacking binary classifiers, either linear or with kernels. Perturbation bounded with ℓ_1 and ℓ_2 norms

Chosing a good starting point
Use density estimator to understand where to pick starting point, as initialization might be bad

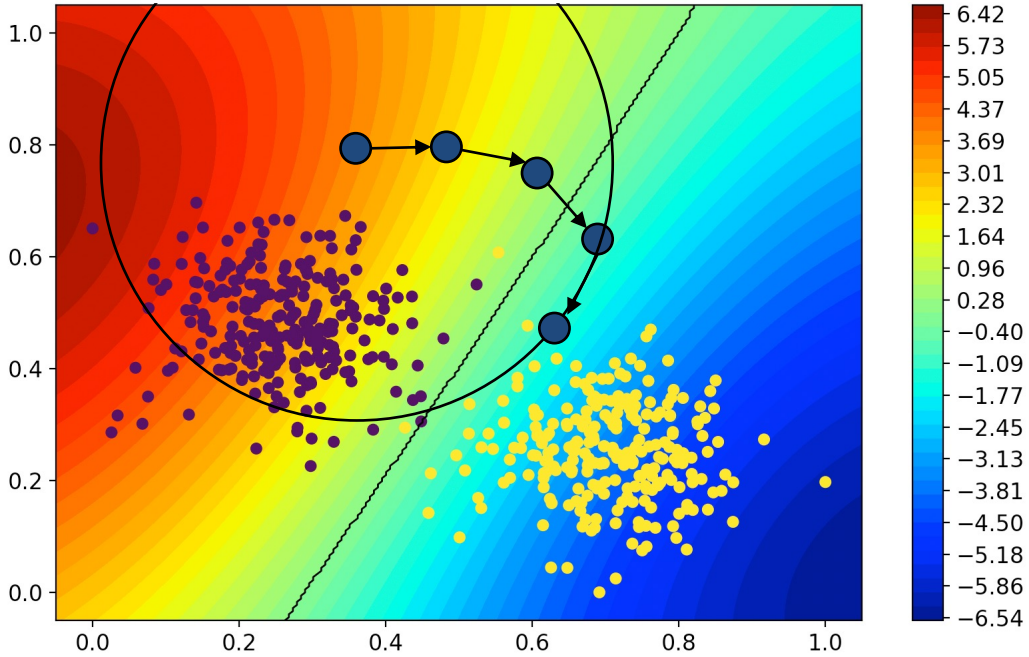


[Biggio et al., Evasion Attacks Against Machine Learning at Test Time? ECML 2013]

Iterative steps of GD-KDE

Each iteration moves the samples closer to the other class

If the constraint is exceeded, the update step is projected back inside the feasible domain

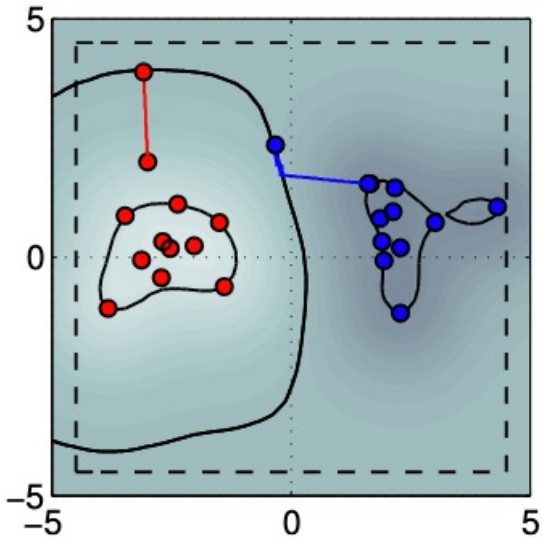


 [Biggio et al., Evasion Attacks Against Machine Learning at Test Time? ECML 2013]

2015: Adversarial Initialization

Starting from the goal
The algorithms brings a sample from the target class close to the boundary of another class, and not the opposite. Again, constraints both ℓ_1 and ℓ_2 norms

Better initialization
No need for density estimators, computations are lighter and no kernels are computed



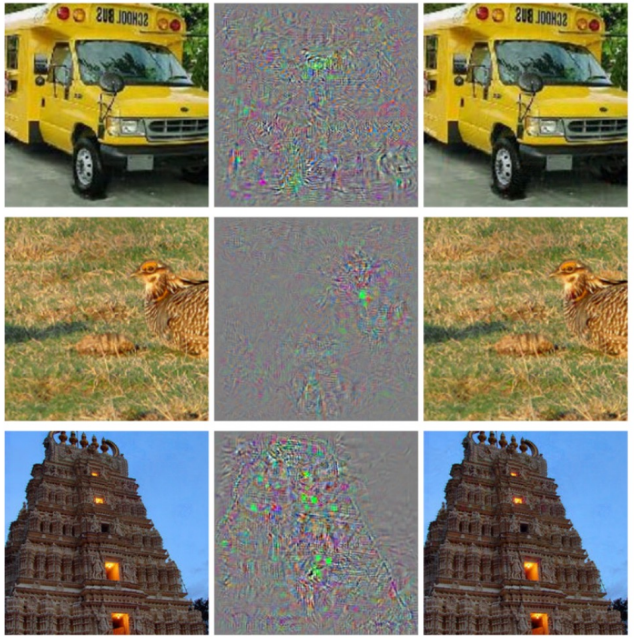
 [Zhang, Biggio et al., Adversarial feature selection against evasion attacks, IEEE TC 2015]

2014: First attacks on Neural Networks

Parallel reasearch: messing with DNNs
Started as a way to understand what these deep networks learned, ended in finding blind spots

Prediction is non-robust
Findings point towards the presence of subspaces where predictions are wrong and unstable

General property of networks
This problem is encountered not only in one architecture, but in many as well



[Szegedy et al., Intriguing Properties of Neural Networks, ICLR 2014]

2015: Parallel reasearch, Fast Gradient Sign Method

Definition of attack
From an intruiging property to a defined process for computing adversarial examples

Single step attack
Perturbation is computed once, by following the gradient of th function toward the desired class

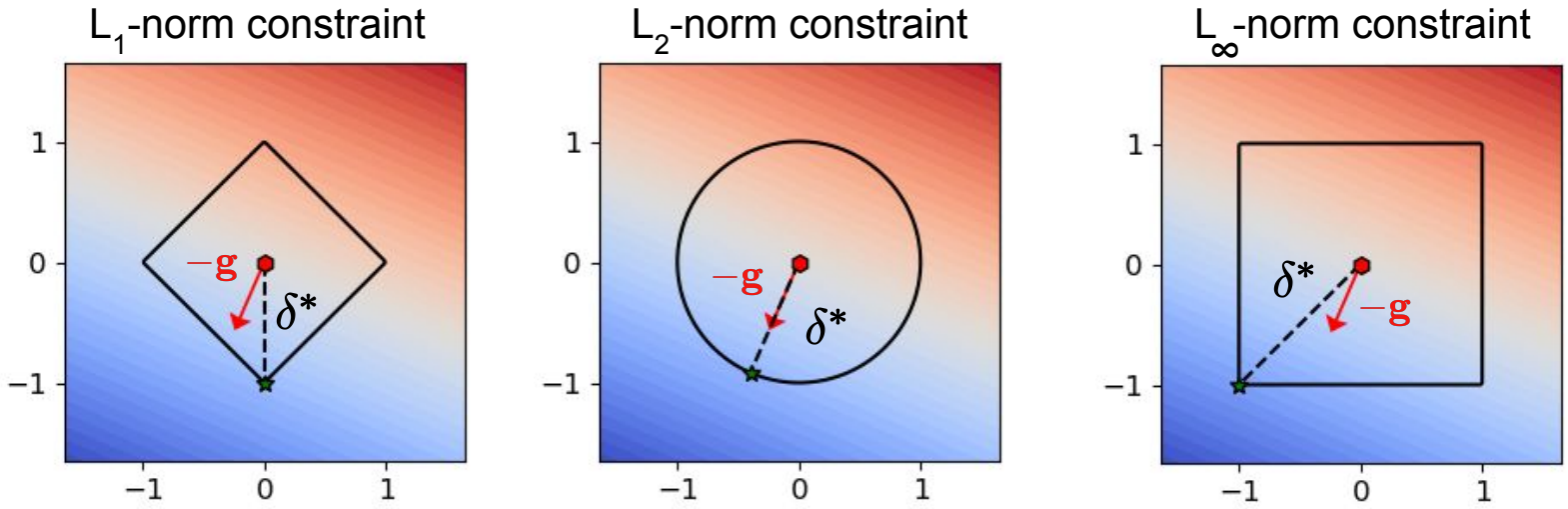
Perturbation is controlled
The attack applies a bound on the maximum norm of the perturbation

$$x + \epsilon\, sign(\nabla L(x, y; \theta))$$

[Goodfellow et al., Explaining and Harnessing Adversarial Examples ICLR 2015]

How it works

$$x + \epsilon\, sign(\nabla L(x, y; \theta))$$



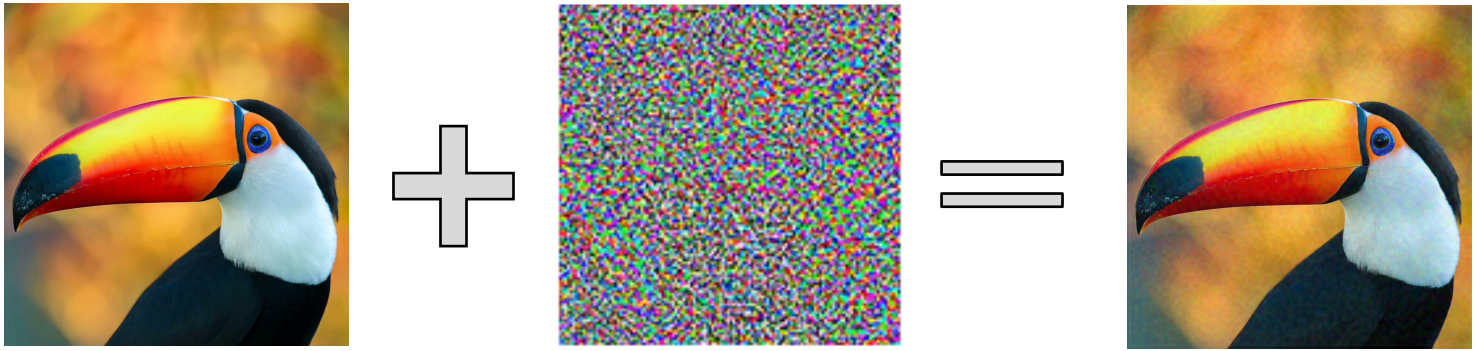
Dual norm

Goal is maximise the perturbation, hence the dual norm is needed

For infinity norm, the maximum is given by the sign function (angles of the box)

[Goodfellow et al., Explaining and Harnessing Adversarial Examples ICLR 2015]

Global Perturbation



Toucan
67% confidence

$\epsilon\ sign(\nabla L(x, y; \theta))$

Orange
98% confidence

All pixels are perturbed by ϵ, creating a diffused noise on the image
The smaller the ϵ, the more unnoticeable the manipulation is

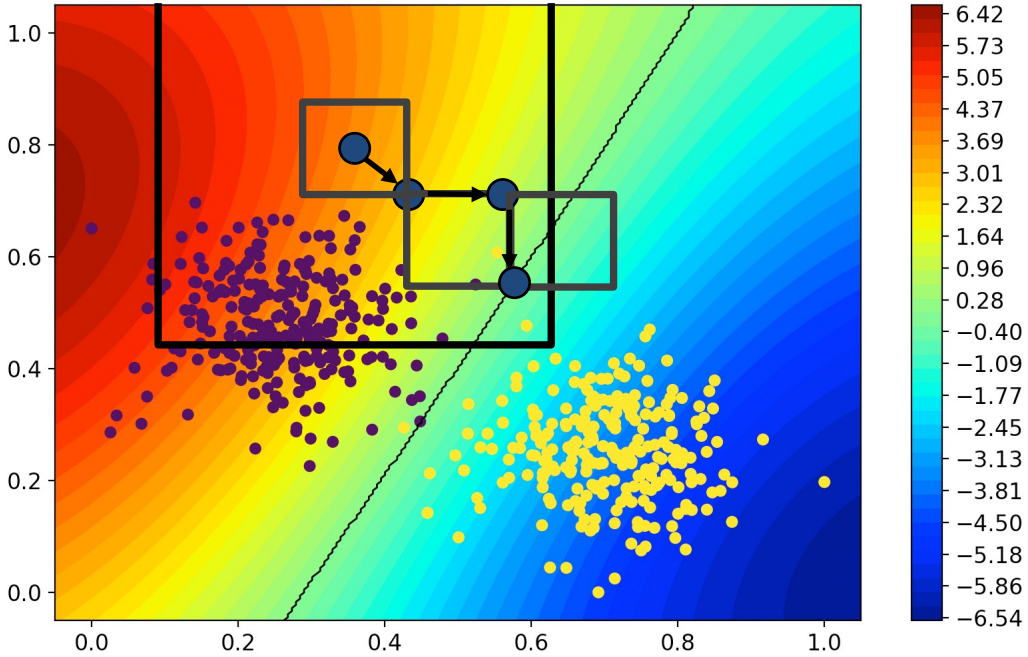
[Goodfellow et al., Explaining and Harnessing Adversarial Examples ICLR 2015]

2017: Basic Iterative Method / Projected Gradient Descent

FGSM but iterative
Same as before, repeated over many iterations until evasion is found or constraint is reached

Each step is aligned to the box
Since it is using ℓ_∞, each step follows the shape of the box

Projected Gradient Descent
Mostly known as PGD for the formulation of Madry et al.



[Kurakin et al., Adversarial Examples in the Physical World, ICLR W 2017]
[Madry et al. Towards deep learning models resistant to adversarial attacks, ICLR 2018]

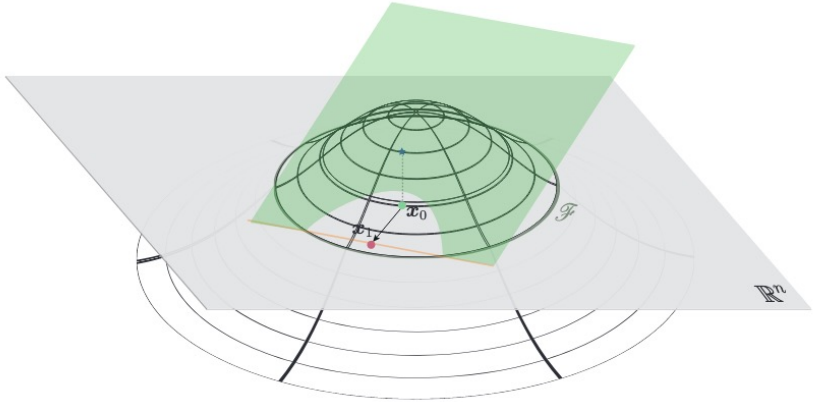
2016: DeepFool attack

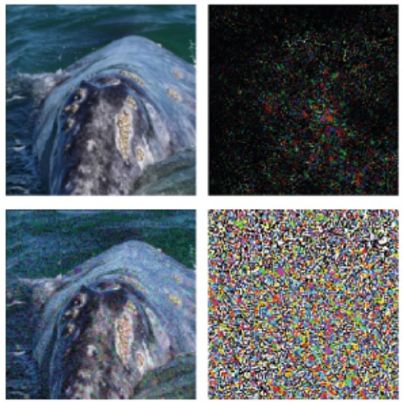
Reducing the perturbation

Perform steps by linearly approximate the decision function, according to the ℓ_2 norm.

Sparse result

Perturbation is minimal, since gradient is orthogonal to the boundary, hence it is closest





DeepFool

FGSM

[Moosavi-Dezfooli et al., DeepFool: a simple and accurate method to fool deep neural networks, CVPR 2016]

2016: Jacobian-based Saliency Map Attack (JSMA)

Less is more

Trying to do better than invasive perturbation of FGSM, bound number of pixels to perturb. Noise bounded to ℓ_0 norm.

Saliency Map

Relevant pixels are found by computing second-order gradients

$\mathbf{X}^* \leftarrow \mathbf{X}$
$\Gamma = \{1 \dots |\mathbf{X}|\}$
while $\mathbf{F}(\mathbf{X}^*) \neq \mathbf{Y}^*$ and $||\delta_{\mathbf{X}}|| < \Upsilon$ **do**
  Compute forward derivative $\nabla \mathbf{F}(\mathbf{X}^*)$
  $S = \texttt{saliency\_map}\,(\nabla \mathbf{F}(\mathbf{X}^*), \Gamma, \mathbf{Y}^*)$
  Modify $\mathbf{X}^*_{i_{max}}$ by θ s.t. $i_{max} = \arg\max_i S(\mathbf{X}, \mathbf{Y}^*)[i]$
  $\delta_{\mathbf{X}} \leftarrow \mathbf{X}^* - \mathbf{X}$
end while

 [Papernot et al., The Limitations of Deep Learning in Adversarial Settings Euro S&P 2016]

Effect on Pixels

Distortion is limited to **few edits**

Example: hand-written digits dataset
Few pixels have been edited to shift class

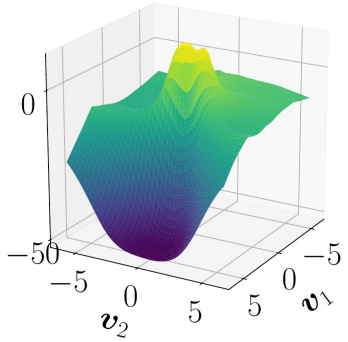
Classified as 4

 [Papernot et al., The Limitations of Deep Learning in Adversarial Settings Euro S&P 2016]

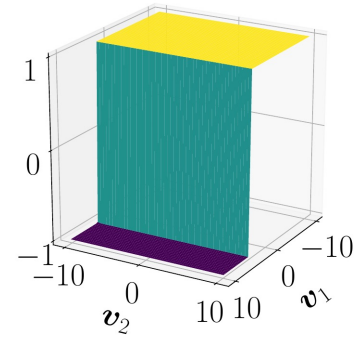
2016: Distillation, defending against attacks

Trying to cope with these blind spots
Masking the real output of the model, by making it difficult to navigate with gradients

Signal lost
Distilled models lead to flat or piece-wise constant regions, and the signal to follow is lost
(More details later on this lesson)



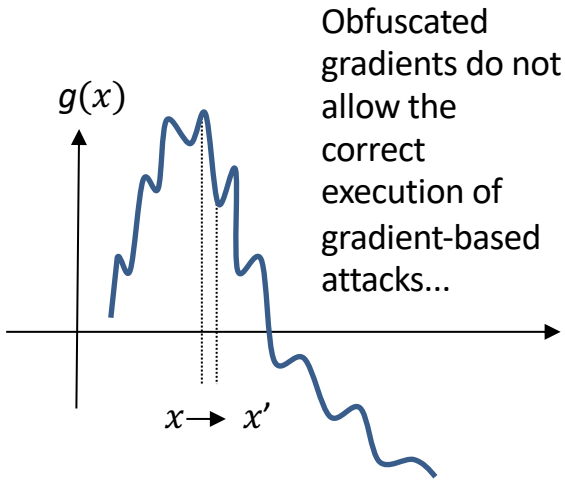
Normal training



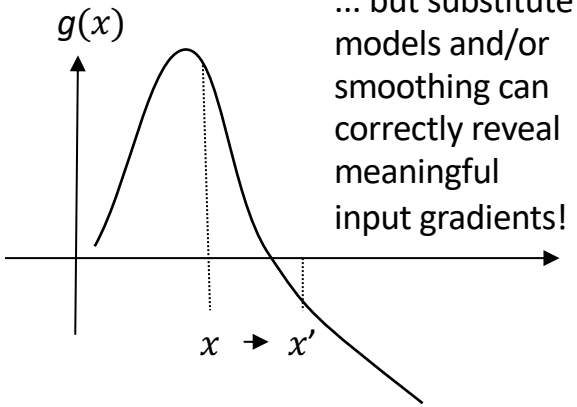
Distillation

[Papernot et al., Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks, S&P 2016]

2016: Distillation is not a secure defense



Obfuscated gradients do not allow the correct execution of gradient-based attacks…



… but substitute models and/or smoothing can correctly reveal meaningful input gradients!

Defenses not properly tested give a «false sense of security»
Beginning of an «arms race» between attackers and defenders
(More details later on in this course)

[Carlini et al., Defensive Distillation is Not Robust to Adversarial Examples, preprint]

Carlini-Wagner Attack

Born to break Distillation
Remove the softmax, directly using the logits of the classes

Remove constraints
All the other attacks were using constraints on perturbation, here the score is used as a penalizer, modulated by c.

Minimum distance
Attack tries to find the closest path to the other class, both with ℓ_2 and ℓ_∞

Adding parameters
Maximise the efficacy by introducing *k*, that allows the attack to further enter inside the target class

$$\min \|\delta\|_p + c(f(x+\delta) - k)$$

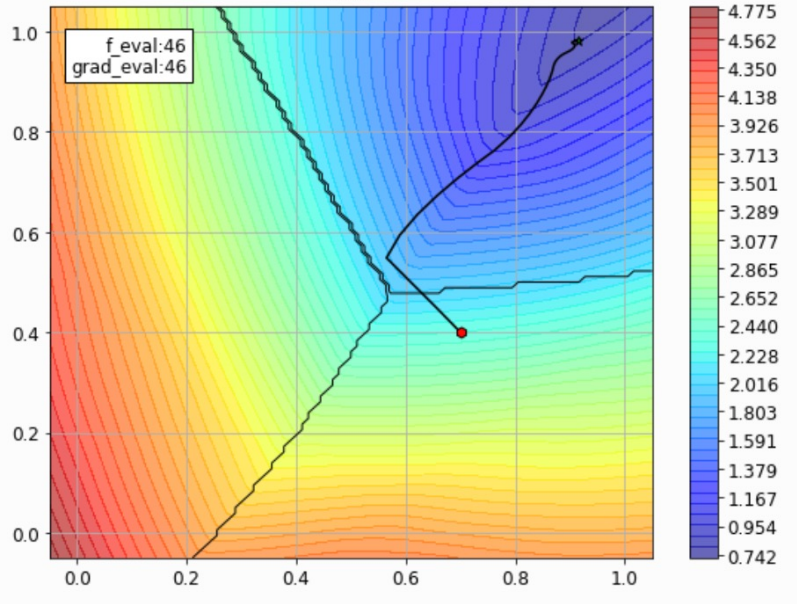
 [Carlini et al., Towards Evaluating the Robustness of Neural Networks S&P 2017]

Carlini-Wagner Attack

Attack follows loss surface before steering to reach the target decision boundary of the target class

Here $k > 0$ hence sample enters the target class. If this parameter is too large, attack is not minimum distance anymore

If c too large, optimizer will behave like an hard-constraint attack on the score



 [Carlini et al., Towards Evaluating the Robustness of Neural Networks S&P 2017]

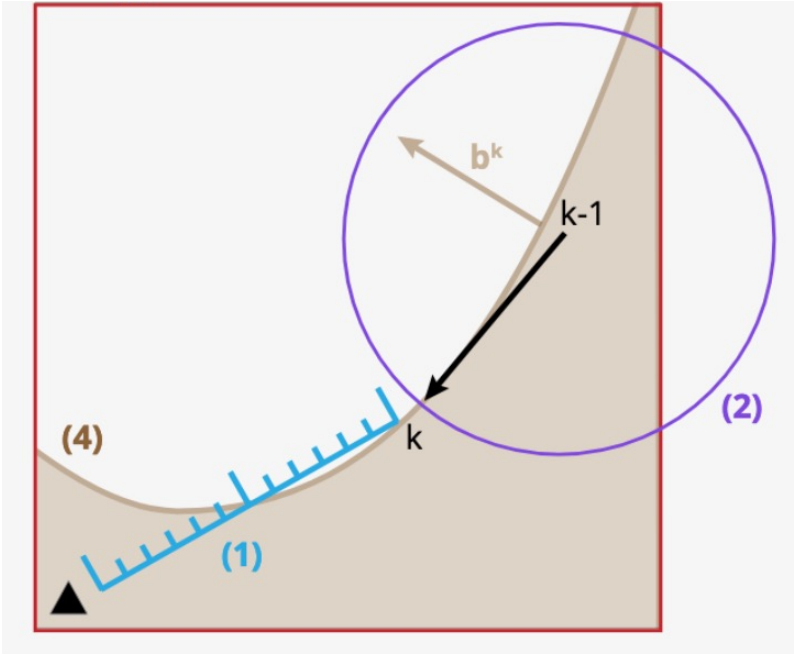
2019: Brendel & Bethge Attack (BB)

Meeting in the middle

Working in the opposite way, by taking a point inside the target class and bringing it into the other class, by walking on the boundary.
Again, adversarial initialization!

Minimum distance

After having found and adversarial example of the target class towards the boundary, the attack take a point of the real starting class towards that



[Brendel et al., Accurate, Reliable and Fast robustness Evaluation, NeurIPS 2019]

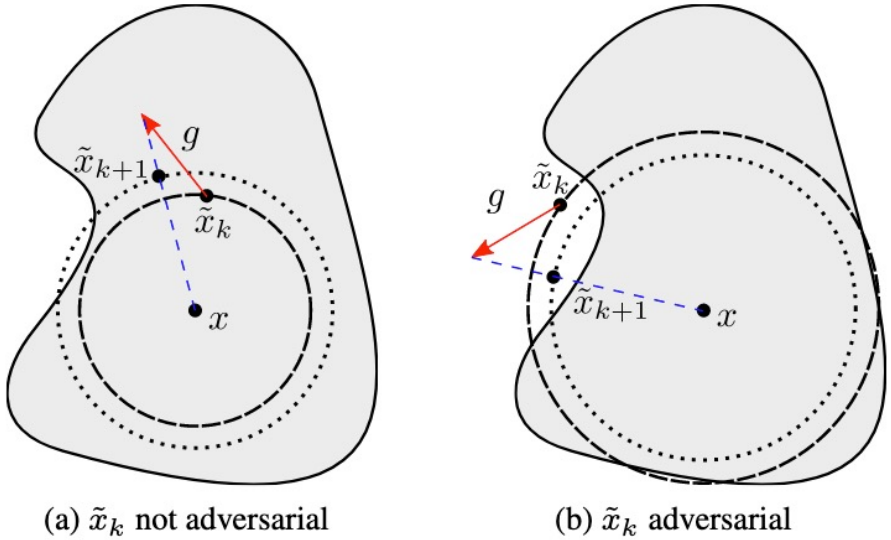
2019: Decoupled Direction and Norm Attack (DDN)

Enlarging the constraints

Instead of fixing the constraint, the attack updates it iteratively, by increasing or decreasing it accordingly (starting from a chosen one)

Minimum distance

The attack finds the ℓ_2 ball with minimal radius where adversarial examples are, centered on the starting point



(a) $\tilde{x}_k$ not adversarial (b) $\tilde{x}_k$ adversarial

[Rony et al., Decoupling Direction and Norm for Efficient Gradient-based L2 Adversarial Attacks and Defenses, CVPR 2019]

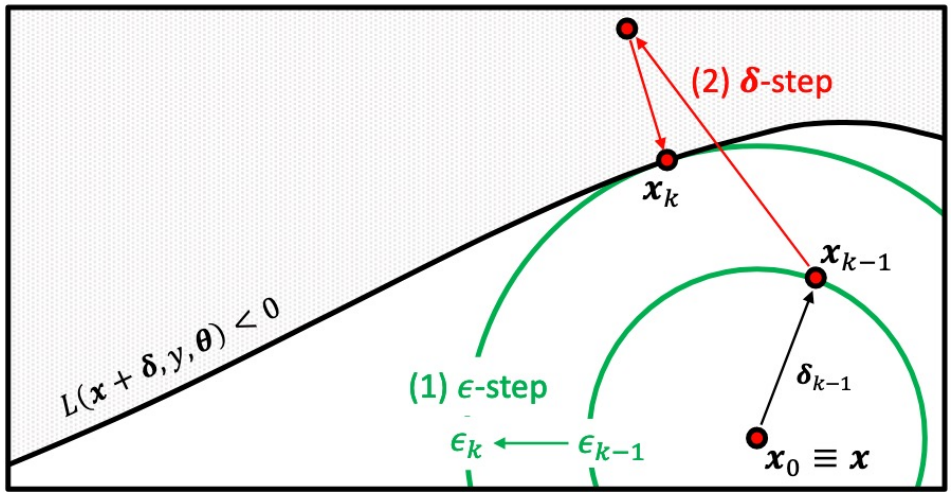
2020: Fast Minimum-Norm Attack (FMN)

Generalizing to Lp norms

Same concept of DDN, but general w.r.t. the chosen norm (not only ℓ_2)
Constraint is automatically computed, no need to set up the attack

Minimum distance

The attack finds the ℓ_0, ℓ_1, ℓ_2 and ℓ_∞ ball with minimal radius where adversarial examples are, centered on the starting point



No parameters fine-tuning!

[Pintor, Biggio et al., Fast Minimum-norm Adversarial Attacks through Adaptive Norm Constraints, preprint]

The Effect of Different Norms

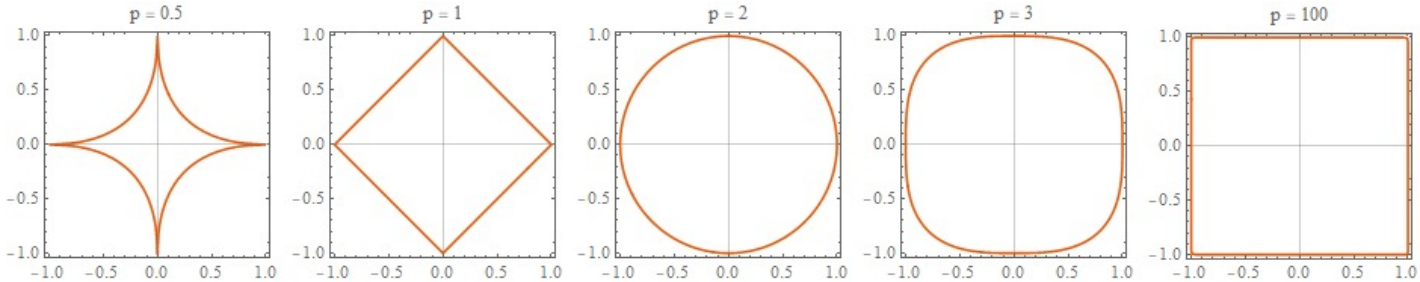
Constraints are Regularizers

Used norms are the convex ones

ℓ_p norms with $p \geq 1$, $\ell_p(\boldsymbol{x}) = \left(\sum_j |x_j|^p\right)^{1/p}$

Most popular examples

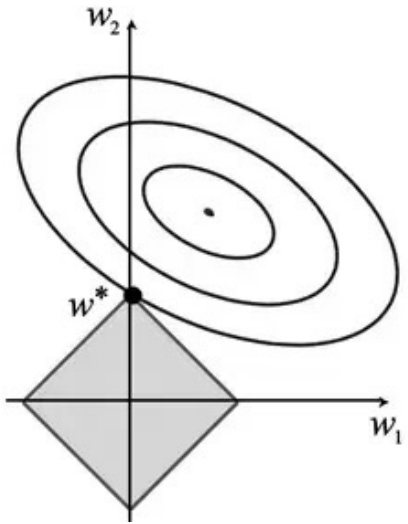
- ℓ_0 is not convex, and amounts to counting non-zero elements in x
- $\ell_1 = |x_1| + |x_2| + \cdots + |x_d|$
- $\ell_2 = x_1^2 + x_2^2 + \ldots + x_d^2$
- $\ell_\infty = \max_j |x_j|$



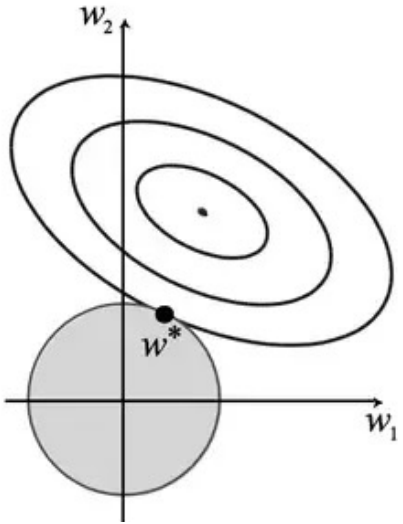
Sparsity

- ℓ_0 and ℓ_1 regularization enforce *sparsity*, i.e., many values in $\boldsymbol{x}$ will be set to zero
  - Why? The optimum is often found at one of the vertices!

- Sparsity helps automatically perform feature selection
- Features assigned $w_j = 0$ can be disregarded



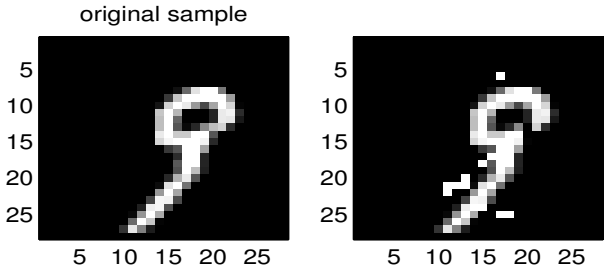
ℓ_1



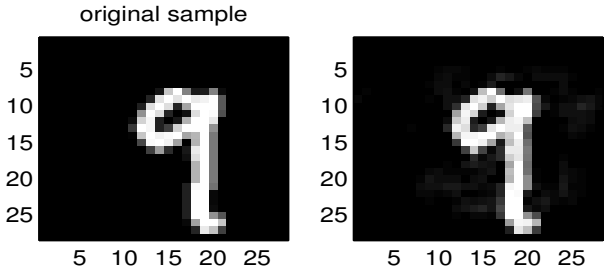
ℓ_2

Sparse vs Dense Attacks

Sparse evasion attacks (l1-norm constrained)



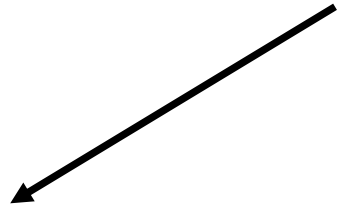
Dense evasion attacks (l2-norm constrained)



Categorizing the Attacks

General Categorization

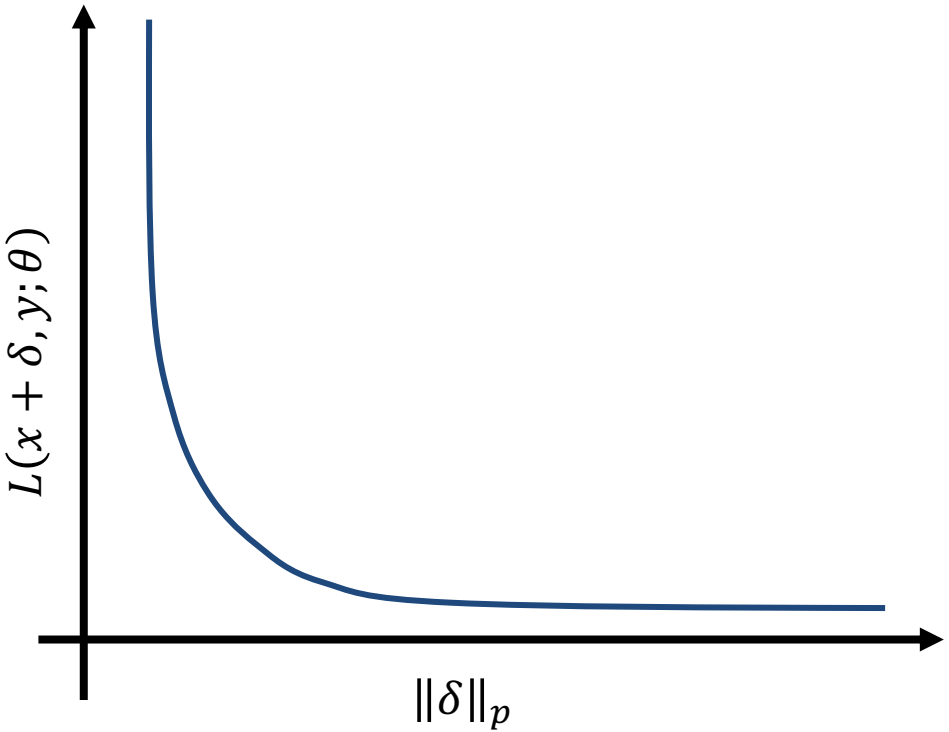
$$\min[L(x+\delta, y; \theta), \|\delta\|_p]$$



Minimize the score, cause misclassification in model

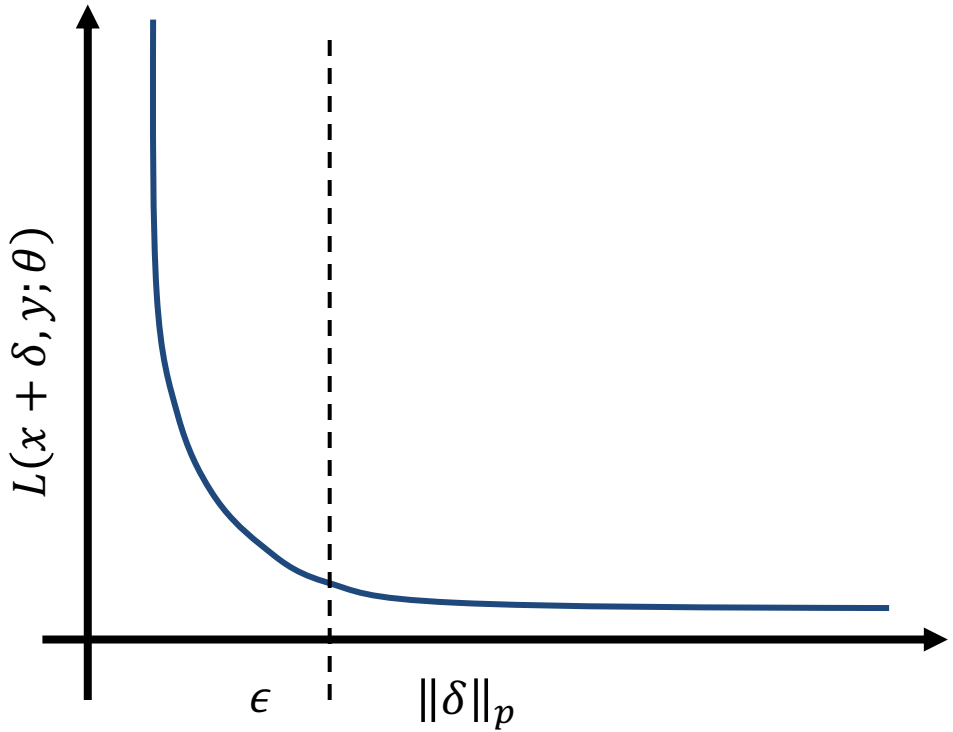
Minimize the perturbation w.r.t. L-p norm

Pareto Frontier



Optimization as trade-off
Not possible to gain best from both worlds, move along the blue curve to find sub-optimal solutions for both quantities

Hard-constraint: maximum confidence attacks

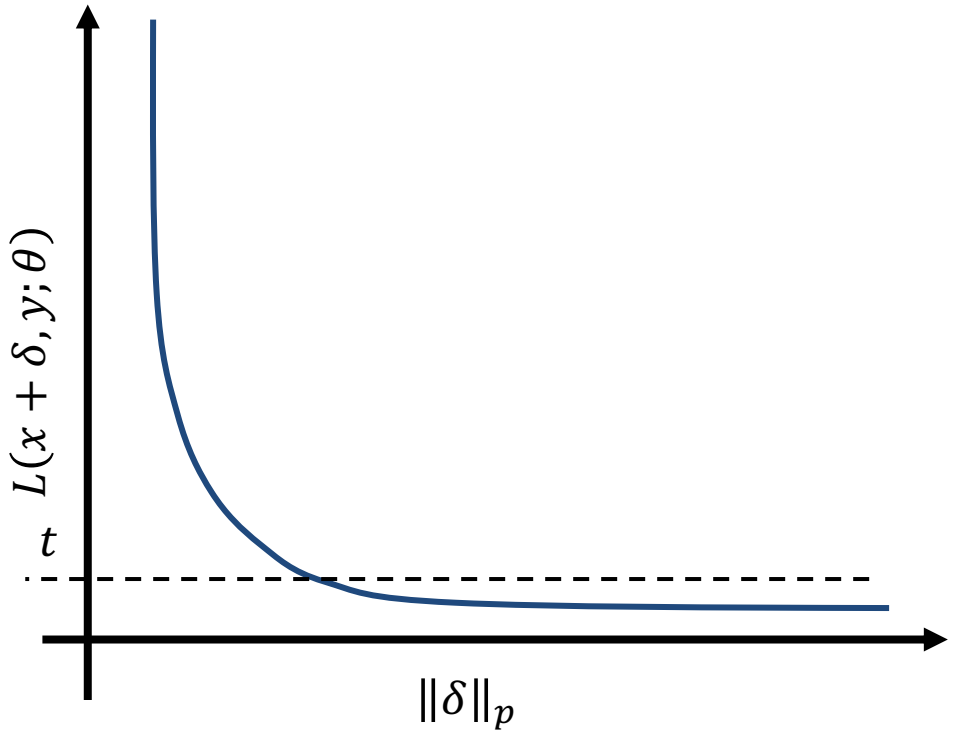


Minimize loss of the attack to cause misclassifiation (FGSM, PGD)

The perturbation is checked as hard constraint, bound on maximum manipulation

$$\min L(x+\delta, y; \theta),$$
$$s.t. \|\delta\|_p < \epsilon$$

Hard-constraint: minimum distance attacks

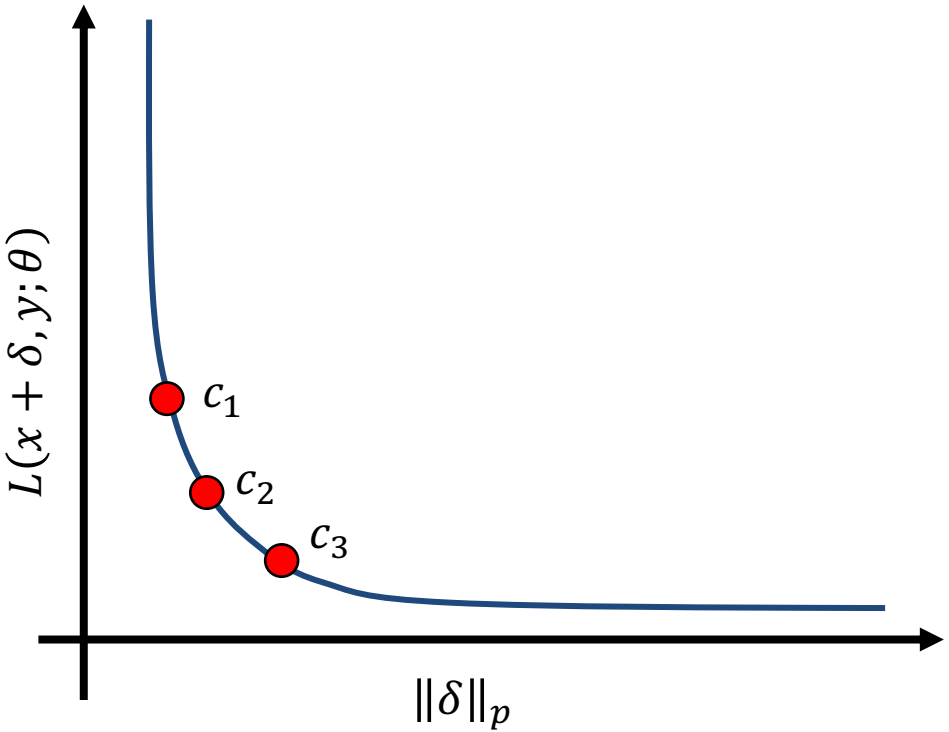


Minimize perturbation w.r.t. Lp norm (JSMA, CW with c=0)

Score is used only as a constraint, not optimized

$$\min \|\delta\|_p$$
$$s.t.\, L(x+\delta, y; \theta) < t$$

Soft-constraint: mixing the problems to solve



All constraints are imposed as quantities modulated by coefficients, behaving as regularizers

Modulating the multipliers shifts the solution towards trade-off between score and distance (CW attack)

$$\min L(x+\delta, y; \theta) + c\|\delta\|_p$$

Final recap on gradient-based attacks

Norm	HC – Max. Confidence	HC – Min. Distance	Soft constraints
L_0		JSMA, FMN	
L_1	GD-KDE	BB, FMN	
L_2	GD-KDE, PGD	DeepFool, BB, DDN, FMN, CW (with c=0)	CW
L_∞	FGSM, BIM, PGD	BB, FMN	CW

Gradient-free Evasion Attacks

Motivations

Model might be non-differentiable (e.g. Random Forest classifiers)

Target is unavailable, like "Machine Learning as a Service" (MLaaS), available only through APIs

No gradients can be computed in these scenarios, Black-box attacks are needed!

$$\nabla_x L(x, y; \theta)$$

Black-box Transfer Attacks

Attack surrogate classifier

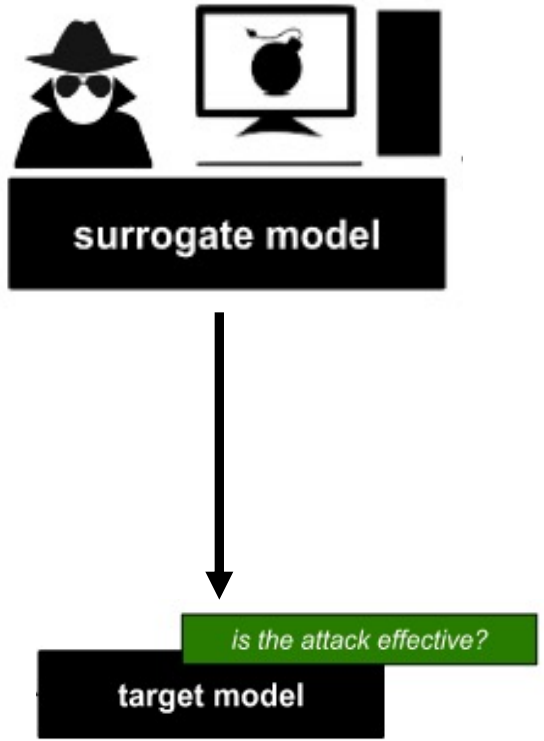
Consider a close approximation of the real target model, by either training a new one on same or similar data, or use a pretrained model

Compute gradient attacks

The attacker chose a differentiable model, to maximize the easiness of computing attacks

Transfer the results

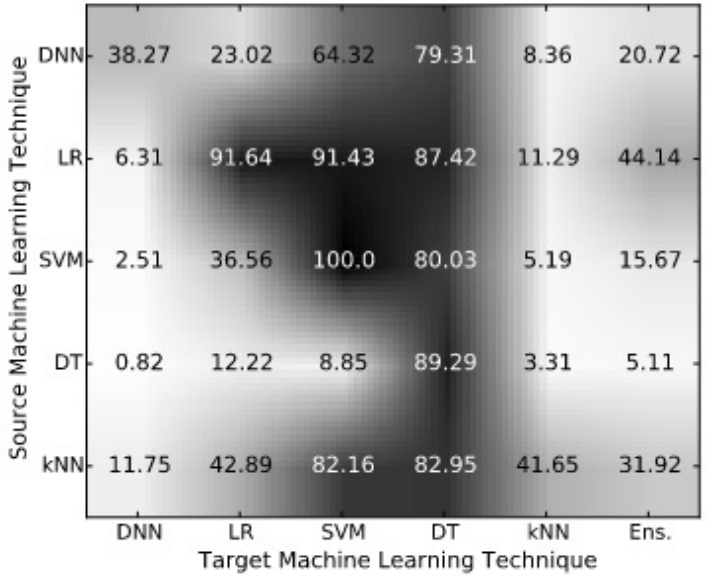
Try to evade the real target using the adversarial examples computed on the surrogate



Surrogate models

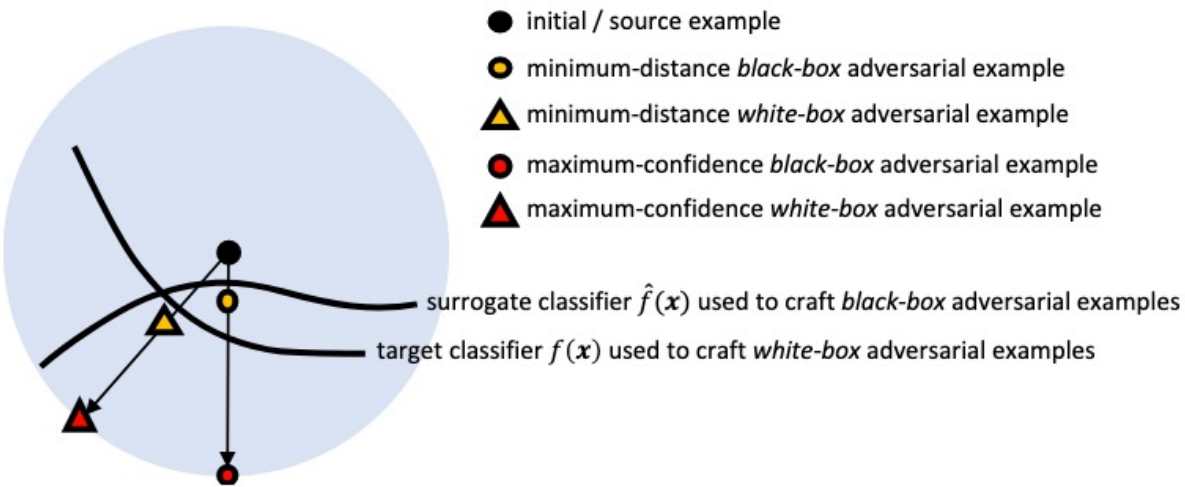
Competition between models
Training different surrogates and compute how they transfer "all vs all"

Different techniques has different results
The heatmap shows that different models behave differently when tested with attacks optimized on other models



[Papernot et al., Practical Black-box Attacks against Machine Learning, Asia CCS 2017]

Do transfer attacks work?



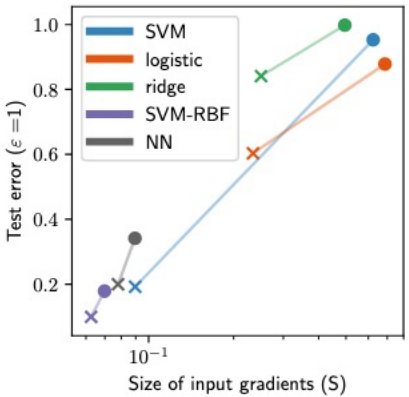
Maximum confidence attacks might better transfer, but more perturbation is needed

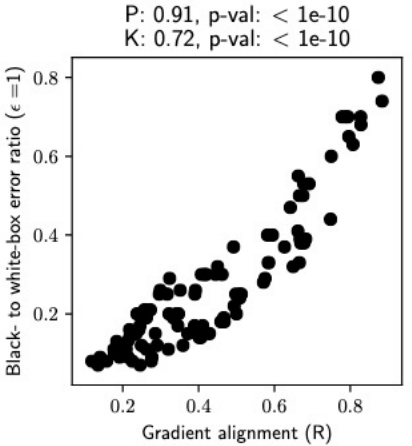
Minimum distance attacks are likely to stop working because decision boundary is different

[Demontis, Biggio et al., Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks, USENIX 2019]

Quantifying transferability through metrics

$$T = \ell(y, \mathbf{x} + \hat{\boldsymbol{\delta}}, \mathbf{w}) \approxeq \ell(y, \mathbf{x}, \mathbf{w}) + \hat{\boldsymbol{\delta}}^{\top} \nabla_{\mathbf{x}} \ell(y, \mathbf{x}, \mathbf{w})$$





Insights of the chance of transfer

By looking at the size of gradients, the gradient alignment, and the variatio of the loss function, it is possible to understand if a model will suffer from transfer attacks

[Demontis, Biggio et al., Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks, USENIX 2019]

Recap

Benefits

No need of original model

Depending on the domain, data and similar pretrained model are already available

Transfer can also happen on other models that solve the same problem as well

Issues

Training the surrogate might lead to training errors, and the attack might not transfer since the approximation is not good enough

Data might be unavailable as well

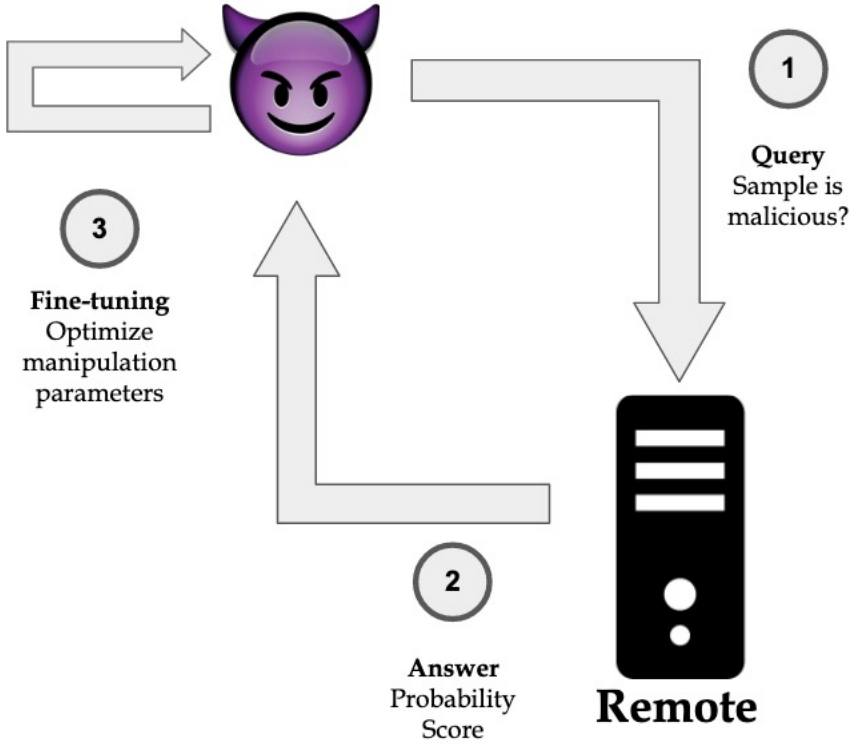
Might require the attacker to interact with the target to extract labels (to be used at training time)

Black-box Query Attacks

No need for training a model from scratch
The attacker can query the target (that might be on a remote server), no need for looking for data, pretrained models, etc.

Less (queries) is more
No Denial-of-Service, optimize the number of queries as well to fly under the radar

Remote replies, attacker optimizes
With the answers, the attacker can guess a local representation of the surface of the target decision function, and planning accordingly



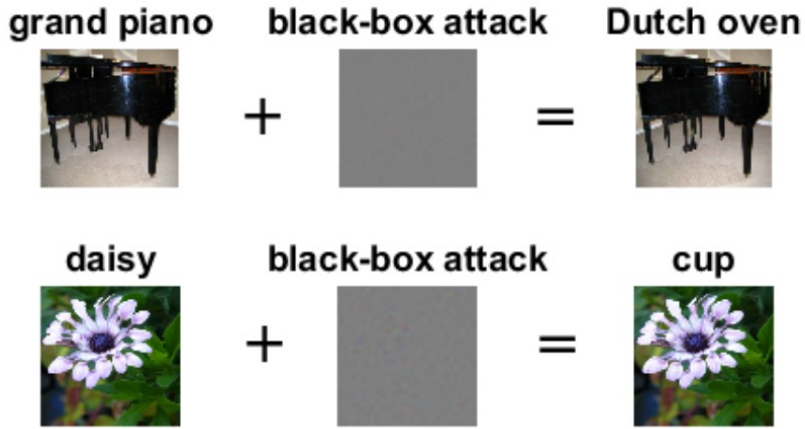
ZOO: Zeroth order attack

No surrogate model
Estimate Hessian (2° order derivative) in each direction, by querying the model locally and around a very small proximity by chosing random directions, no need to train surrogate

Sparse result
Attack only uses randomly picked directions to minimize both queries and perturbation size

$$\hat{g}_i := \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h},$$

$$\hat{h}_i := \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_{ii}^2} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - 2f(\mathbf{x}) + f(\mathbf{x} - h\mathbf{e}_i)}{h^2}.$$

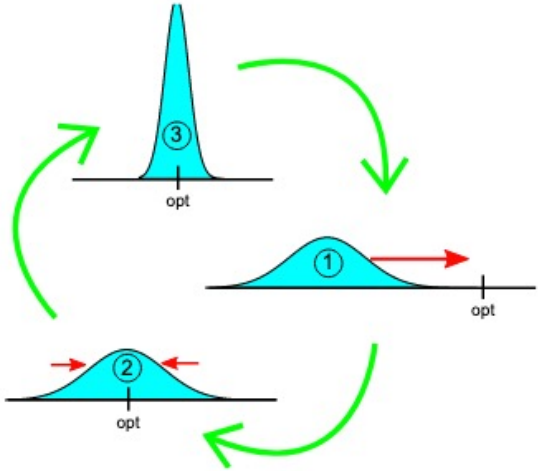


[Chen et al., ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models, AISec 2017]

Natural Evolutionary Strategies (NES)

Reconstruct model parameters
Impose gaussian on N sampled points around the starting point, and learn the coefficient of such model.

Estimate a more robust gradient
The model is differentiable, and the attacker can apply PGD on such gaussian to reconstruct best direction to follow



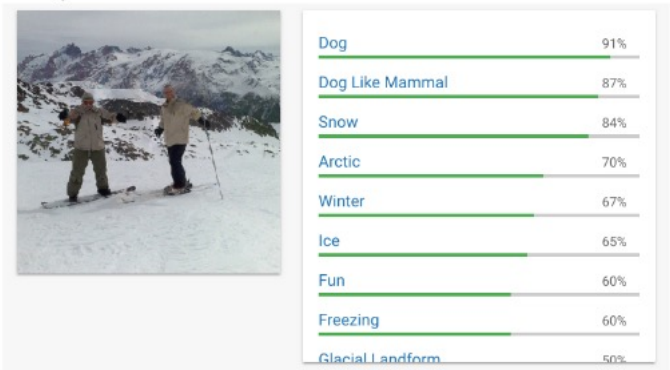
[Ilyas et al. Black-box Adversarial Attacks with Limited Queries and Information, ICML 2018]
[Wiestra et al. Natural evolution strategies, JMLR 2014]

Natural Evolutionary Strategies (NES)

Not random directions, but samples
Differently from ZOO, NES use all the directions to understand a reliable signal, hence perturbation might be larger but more effective

Tested also against MLaaS

Effective in real case scenarios, attacking Google Cloud Vision

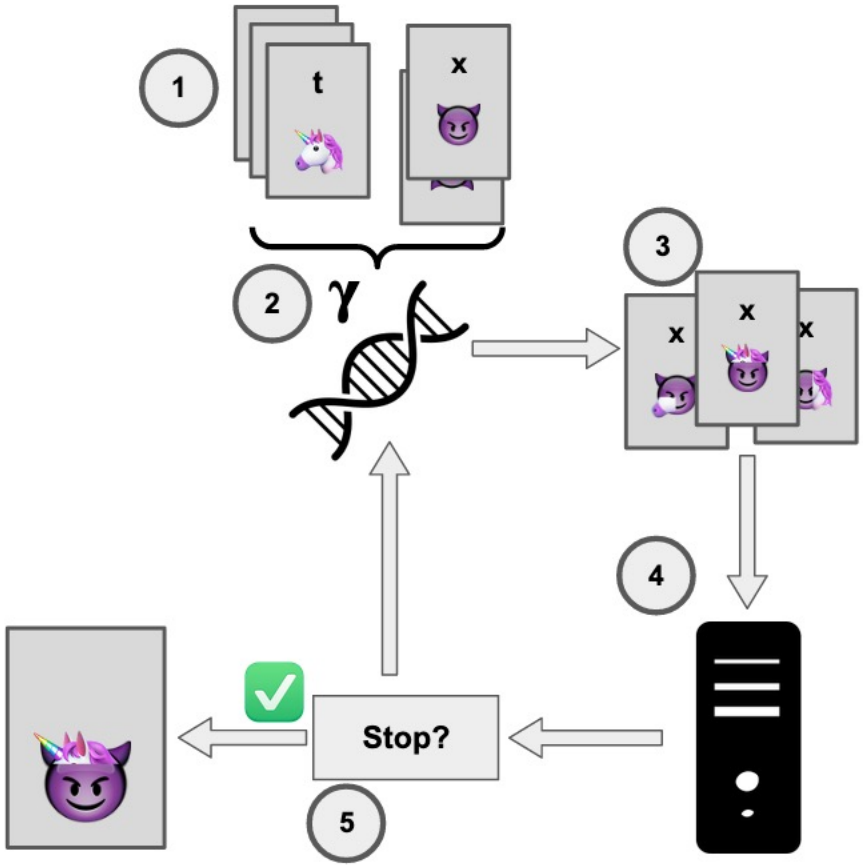


[Ilyas et al. Black-box Adversarial Attacks with Limited Queries and Information, ICML 2018]
[Wiestra et al. Natural evolution strategies, JMLR 2014]

Genetic Algorithms

Evolving solutions
Sample N points, ask the target for scores, and breed together best candidates. Follow "genetic evolution" until a suitable solution is found.

No gradient estimation
No need for computing approximation of best direction, it is taken care of by the mixing of "genes"



Recap

Benefits

Bounded number of queries might already be enough for evasion

Ready to target real world targets as MLaaS

A lot of different methods for estimate directions

Issues

Decision are local, slower than normal gradient-based attack as it is reconstructing best direction from answers

The number of needed queries might still be enormous, depending on the robustness of the target

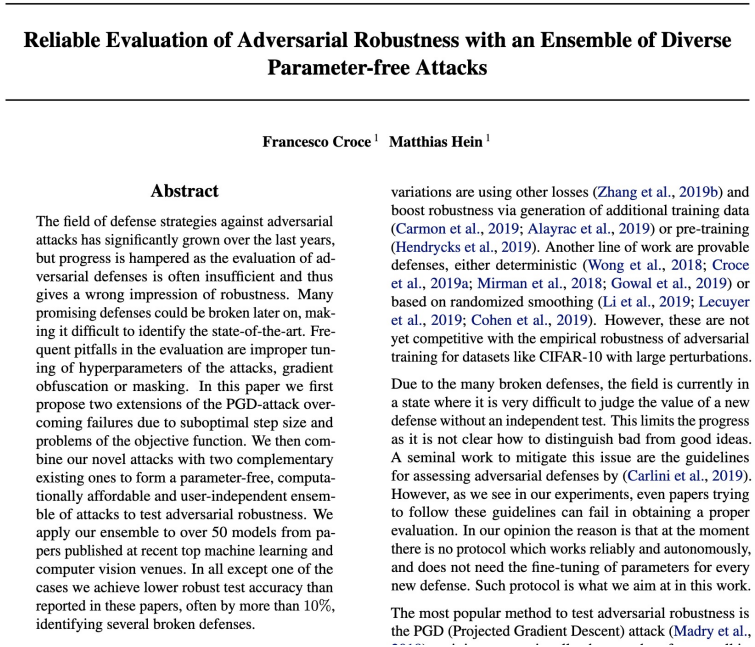
Benchmarks

Automation with AutoAttack

Automatic evaluation of a model, by testing different attacks

AutoPGD: gradient descent by tuning the hyper-parameters on the fly while executing

Free the test from some hyper-parameter tuning during attacks



Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks

Francesco Croce [1] **Matthias Hein** [1]

Abstract

The field of defense strategies against adversarial attacks has significantly grown over the last years, but progress is hampered as the evaluation of adversarial defenses is often insufficient and thus gives a wrong impression of robustness. Many promising defenses could be broken later on, making it difficult to identify the state-of-the-art. Frequent pitfalls in the evaluation are improper tuning of hyperparameters of the attacks, gradient obfuscation or masking. In this paper we first propose two extensions of the PGD-attack overcoming failures due to suboptimal step size and problems of the objective function. We then combine our novel attacks with two complementary existing ones to form a parameter-free, computationally affordable and user-independent ensemble of attacks to test adversarial robustness. We apply our ensemble to over 50 models from papers published at recent top machine learning and computer vision venues. In all except one of the cases we achieve lower robust test accuracy than reported in these papers, often by more than 10%, identifying several broken defenses.

variations are using other losses (Zhang et al., 2019b) and boost robustness via generation of additional training data (Carmon et al., 2019; Alayrac et al., 2019) or pre-training (Hendrycks et al., 2019). Another line of work are provable defenses, either deterministic (Wong et al., 2018; Croce et al., 2019a; Mirman et al., 2018; Gowal et al., 2019) or based on randomized smoothing (Li et al., 2019; Lecuyer et al., 2019; Cohen et al., 2019). However, these are not yet competitive with the empirical robustness of adversarial training for datasets like CIFAR-10 with large perturbations.

Due to the many broken defenses, the field is currently in a state where it is very difficult to judge the value of a new defense without an independent test. This limits the progress as it is not clear how to distinguish bad from good ideas. A seminal work to mitigate this issue are the guidelines for assessing adversarial defenses by (Carlini et al., 2019). However, as we see in our experiments, even papers trying to follow these guidelines can fail in obtaining a proper evaluation. In our opinion the reason is that at the moment there is no protocol which works reliably and autonomously, and does not need the fine-tuning of parameters for every new defense. Such protocol is what we aim at in this work.

The most popular method to test adversarial robustness is the PGD (Projected Gradient Descent) attack (Madry et al.,

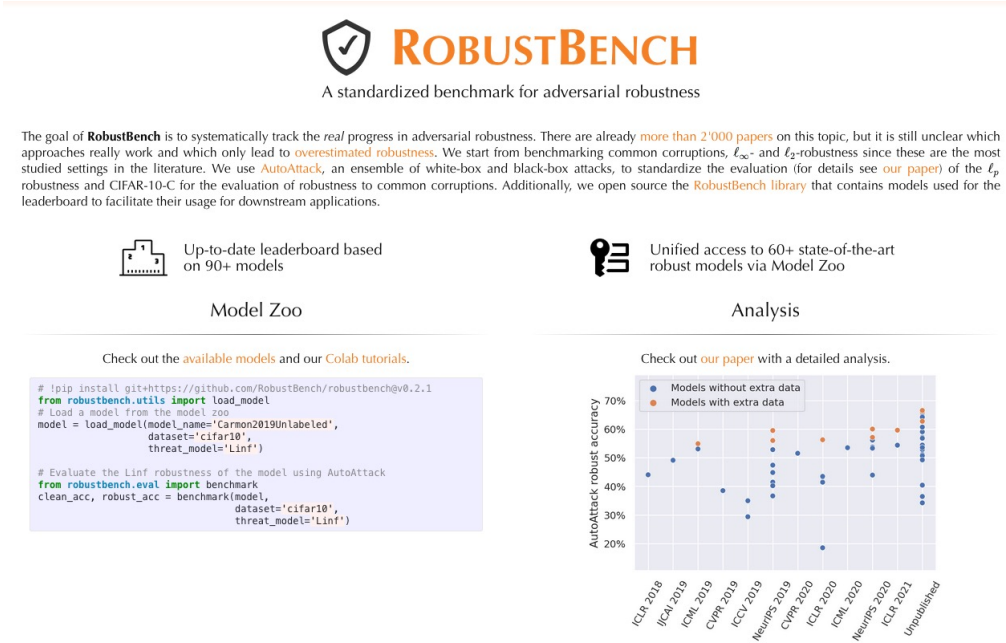
[Croce et al., Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks ICML 2020]

Benchmarks on RobustBench

Public evaluation of latest developed defenses (latest update March 2021) https://robustbench.github.io/

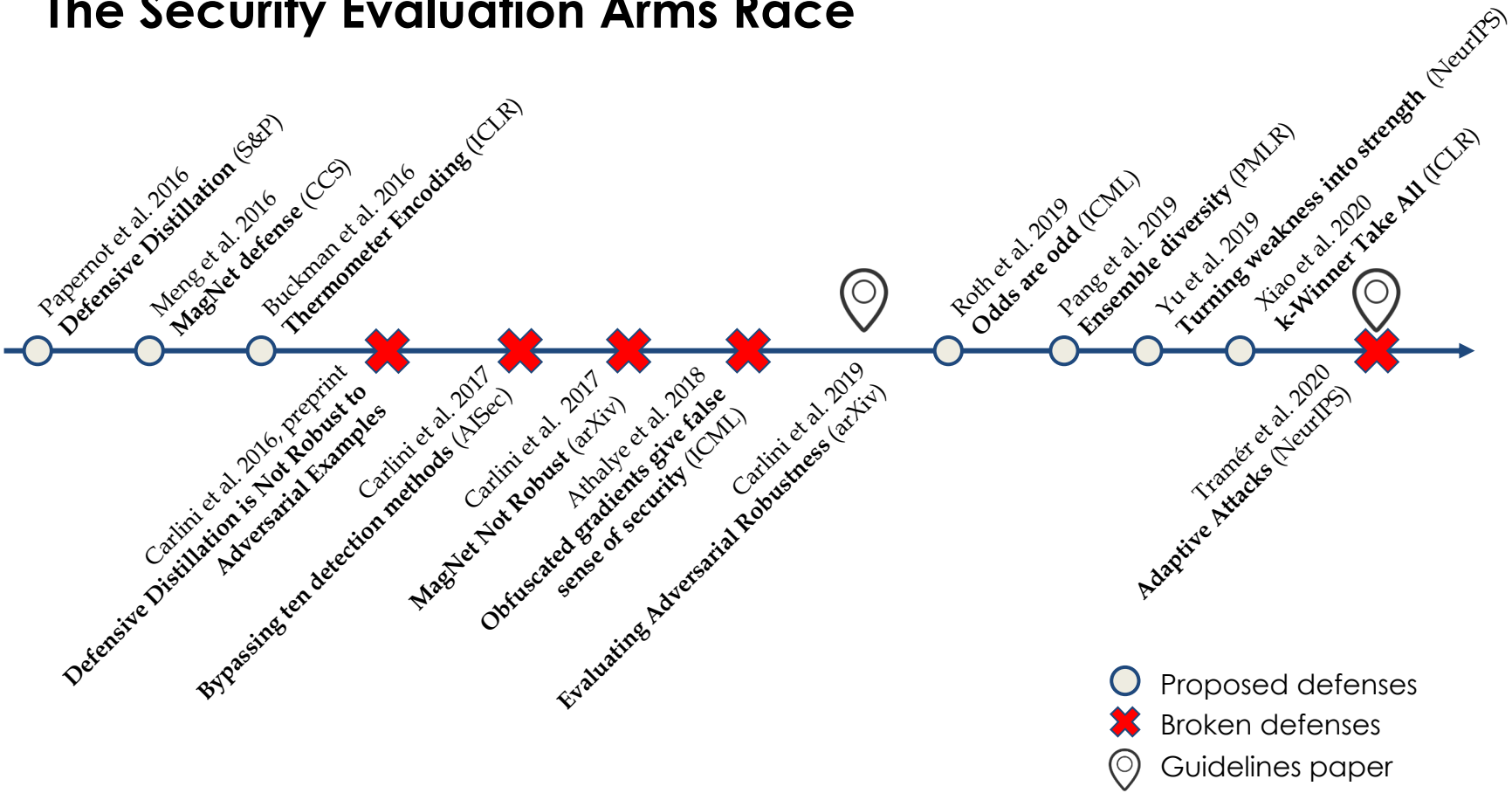
Use AutoAttack, defenses are listed in leaderboard by results

All models are available and can be tested offline



Adaptive Attacks

The Security Evaluation Arms Race



DeepSec: evaluating all attacks against all defenses

CLASSIFICATION ACCURACY OF COMPLETELY DEFENSES AGAINST ADVERSARIAL ATTACKS ON CIFAR-10

Datasets	Attack				# of AEs	Original Model	Defense-enhanced Models										Average
	UA/TA	Objective	Attacks				Adversarial Training			Gradient Mask.		Input Transformation				RC	
							NAT	EAT	PAT	DD	IGR	EIT	RT	PD	TE		
CIFAR-10	UAs	L_∞ $\epsilon=0.1$	FGSM	$\epsilon=0.1$	897	0.0%	76.4%	57.0%	51.0%	7.9%	69.9%	66.7%	7.6%	6.9%	32.6%	0.1%	37.6%
				$\epsilon=0.2$	898	0.0%	52.7%	28.4%	18.5%	10.7%	51.6%	35.0%	4.8%	2.5%	13.3%	0.1%	21.7%
			R+FGSM		837	0.0%	82.7%	80.4%	75.3%	12.2%	76.9%	79.9%	4.2%	4.5%	68.1%	0.0%	48.4%
			BIM		1000	0.0%	84.7%	81.1%	82.4%	3.8%	77.4%	82.3%	0.0%	2.5%	76.5%	0.0%	49.1%
			PGD		1000	0.0%	81.9%	77.8%	74.3%	0.7%	75.1%	79.7%	0.0%	0.2%	64.7%	0.0%	45.4%
			U-MI-FGSM		1000	0.0%	78.7%	65.5%	69.5%	2.7%	73.0%	69.6%	0.0%	0.0%	47.5%	0.0%	40.7%
			UAP		853	0.0%	80.9%	79.8%	60.8%	5.4%	74.4%	71.4%	3.8%	21.3%	47.8%	1.4%	44.7%
		L_2	DF		1000	0.0%	88.9%	86.6%	83.3%	89.3%	79.2%	87.1%	83.2%	74.9%	92.9%	91.3%	85.7%
			OM		1000	0.0%	88.9%	86.1%	82.3%	81.6%	79.0%	87.4%	52.2%	70.5%	91.1%	14.8%	73.4%
	TAs	L_∞ $\epsilon=0.1$	LLC		134	0.0%	79.9%	65.7%	61.2%	1.5%	76.9%	70.2%	3.0%	6.0%	29.9%	0.0%	39.4%
			R+LLC		315	0.0%	84.4%	86.0%	81.3%	6.7%	81.9%	86.0%	4.1%	5.1%	73.3%	0.0%	50.9%
			ILLC		1000	0.0%	86.6%	85.3%	83.7%	27.6%	78.2%	86.9%	0.9%	49.7%	88.5%	0.0%	58.7%
			T-MI-FGSM		1000	0.0%	83.1%	71.4%	70.2%	11.2%	74.5%	78.5%	0.8%	0.0%	61.4%	0.0%	45.1%
		L_0	JSMA		997	0.0%	68.0%	75.1%	72.7%	50.3%	73.5%	70.0%	37.1%	27.1%	75.5%	16.2%	56.6%
		L_2	BLB		1000	0.0%	89.1%	86.4%	83.0%	89.8%	79.2%	87.4%	83.9%	74.1%	92.8%	91.1%	85.7%
			CW2	$\kappa=0$	1000	0.0%	88.8%	86.5%	83.0%	89.5%	79.2%	88.6%	82.9%	76.7%	92.5%	90.2%	85.8%
				$\kappa=20$	1000	0.0%	88.6%	86.3%	82.3%	82.8%	79.2%	88.0%	26.5%	74.4%	92.2%	14.6%	71.5%
			EAD	EN	1000	0.0%	88.5%	86.5%	82.5%	89.2%	79.1%	88.0%	79.3%	74.8%	92.7%	87.5%	84.8%
				L1	1000	0.0%	88.4%	86.6%	82.6%	88.4%	79.0%	86.3%	81.0%	76.2%	92.6%	88.4%	85.0%
	Average				891.1	0.0%	**82.2%**	76.8%	72.6%	39.5%	75.6%	78.4%	29.2%	34.1%	69.8%	26.1%	**58.4%**

All vs All

Intuition is trying all against all, and see what is the best defense for which attack

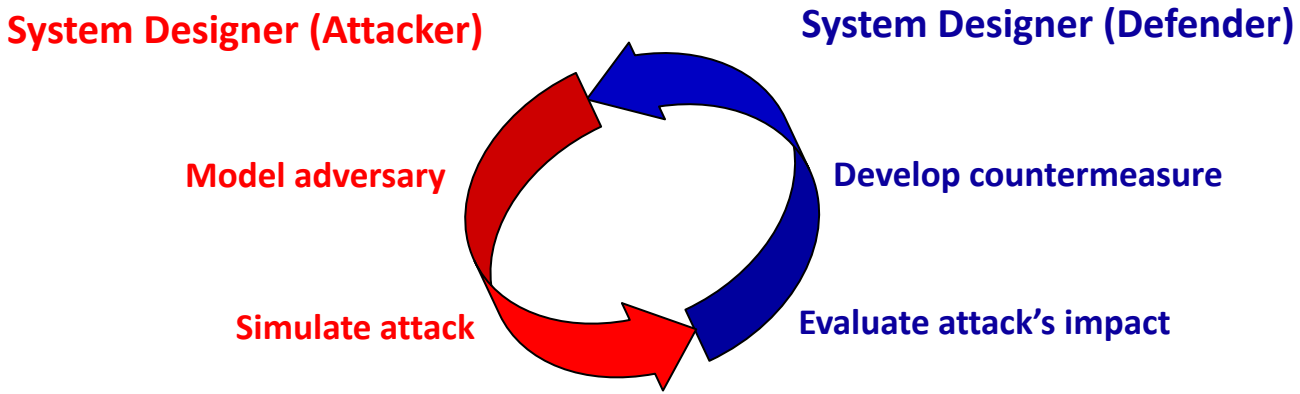
No intuition on adaptiveness of attacks

All the attacks were applied "as is", while the trend is to adapt the attack to the defense to evaluate

[Ling et al., A Uniform Platform for Security Analysis of Deep Learning Model, IEEE SP 2019]

On the Importance of Security Evaluation

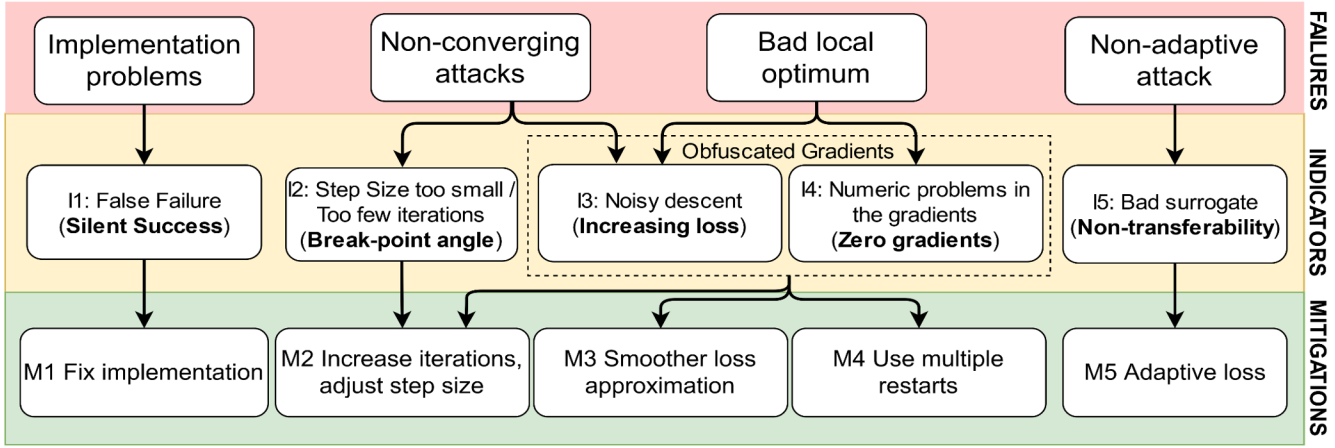
Attackers are *adaptive*
They evolve to break/bypass your defense
Testing against known/previous attacks gives little if no robustness guarantees at all
Be proactive: always test against a stronger attack algorithm that *knows* the defense!



Biggio, Fumera, Roli, IEEE TKDE, 2014

Towards Fair Evaluations

Debugging attacks



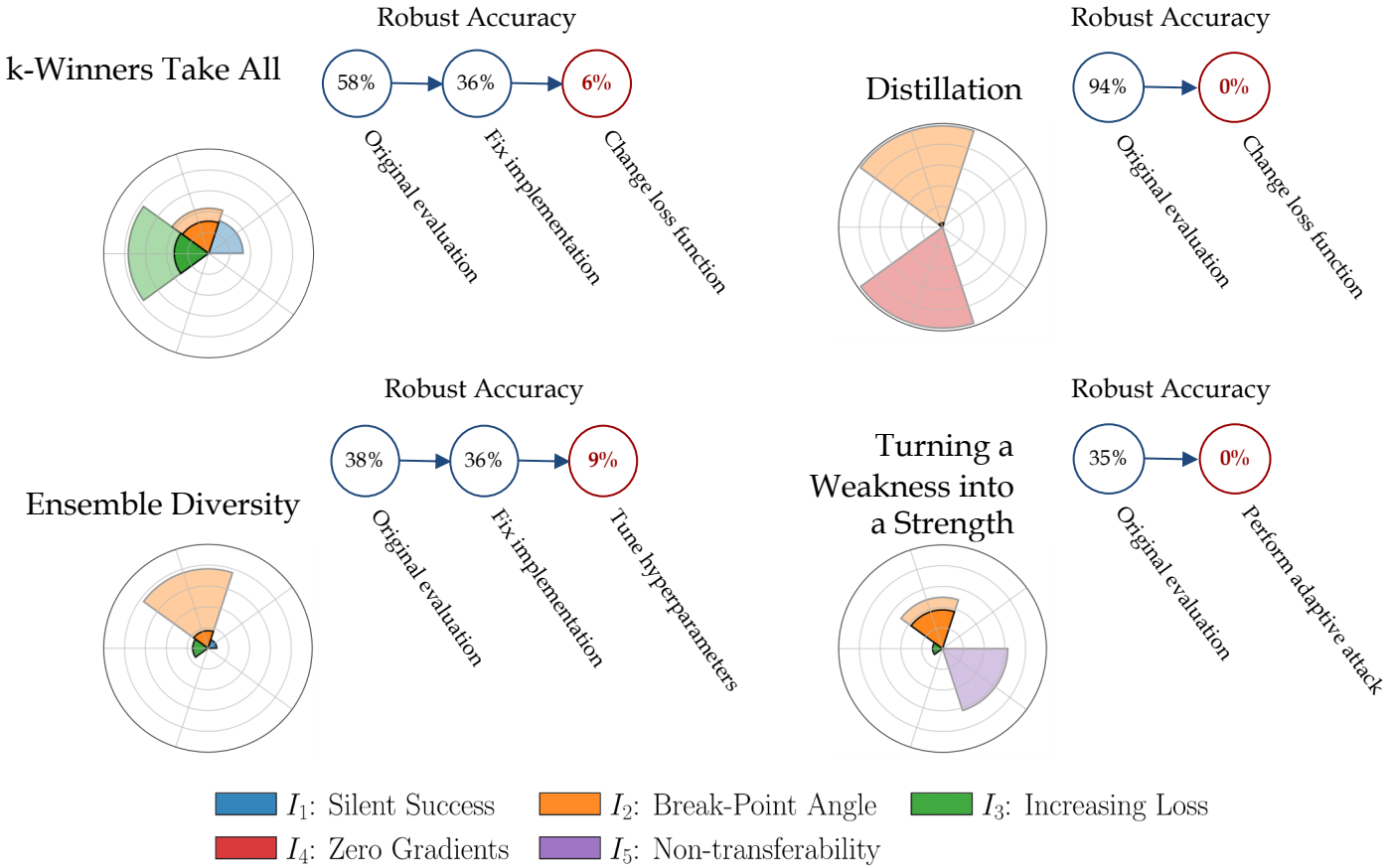
Indicators of Attack Failures

To better evaluate a model, attackers should understand why their attack fail

By mitigating the bugs, the robustness decreases and the evaluation is fair

[Pintor, Biggio et al., Indicators of Attack Failures: Visualizing and Debugging Optimization of Adversarial Examples, preprint 2021]

Experiments



[Pintor et al., Indicators of Attack Failures: ..., arXiv 2021]

Luca Demetrio
luca.demetrio93@unica.it
@zangobot

Thanks!

If you know the enemy and know yourself, you need not fear the result of a hundred battles
Sun Tzu, The art of war, 500 BC

Adversarial EXEmples: Evading Windows Malware Detectors

Luca Demetrio

Department of Electrical and Electronic Engineering
University of Cagliari, Italy

Differences with Images

Byte-sequences are not numbers

Programs and images are encoded in bytes

RGB is “continuous”, code instructions are not!

Distance between programs is **undefined**

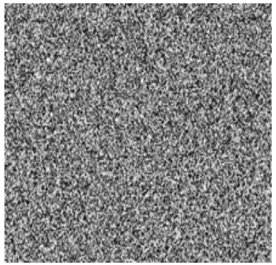
Example: ASCII table
What does $\sqrt{'a' - 'b'}$ means?

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

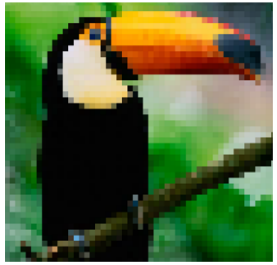
Source: www.asciitable.com

Format constraints

toucan (97%)

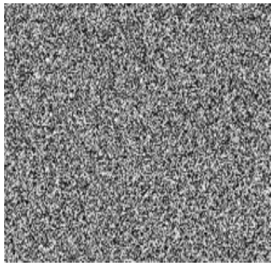


adversarial noise



cat (95%)

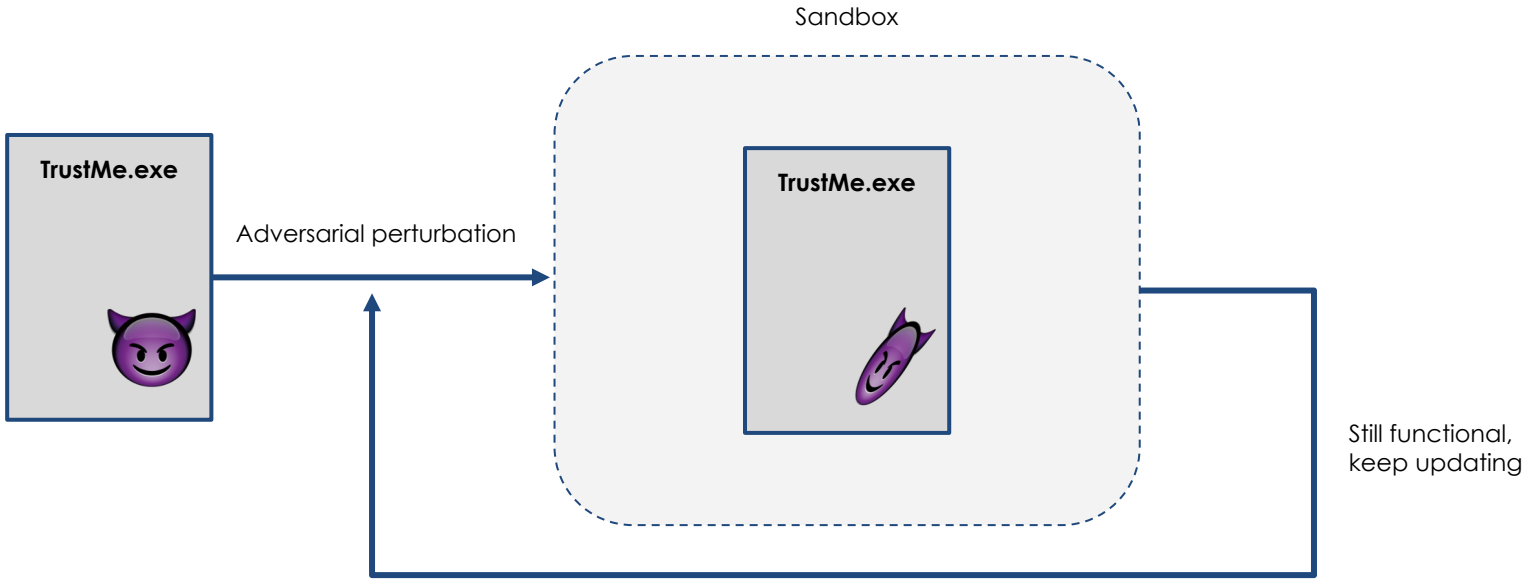
malware (98%)



adversarial noise

Not runnable anymore!

Preserve the original functionality



Song et al., *MAB-Malware: A Reinforcement Learning Framework for Attacking Static Malware Classifiers*, arXiv preprint 2021
Castro et al., *AIMED: Evolving Malware with Genetic Programming to Evade Detection*, TrustCom/BigDataSE 2019

How to bridge these gaps?

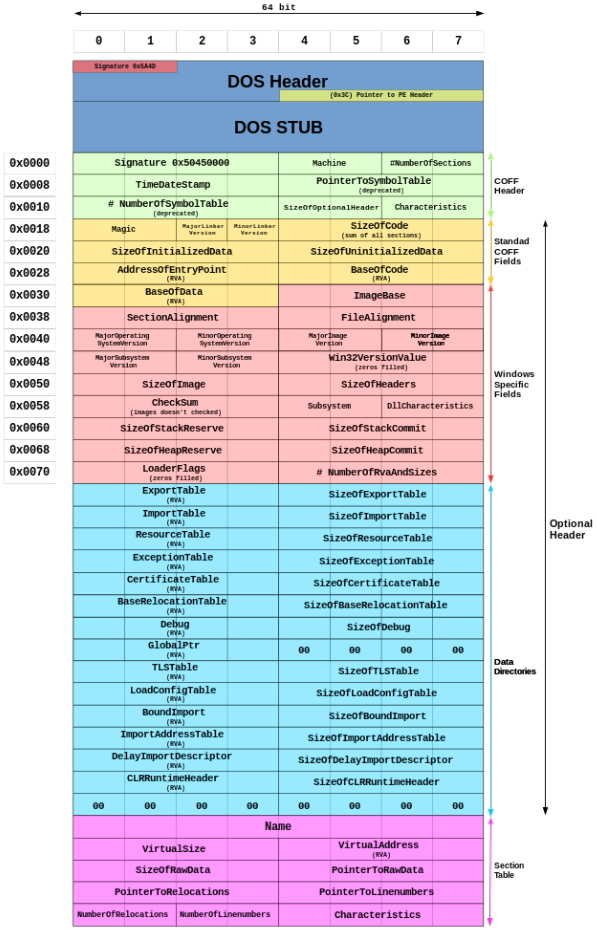
1. Study the format that represent programs
2. Understand how to exploit the format
3. Chose how to inject or perturb the content

Windows PE File Format

Windows PE File Format

Format adapted for “modern” programs (from Windows NT 3.1 on)

Before there were other formats, one is the DOS (kept for retrocompatibility)

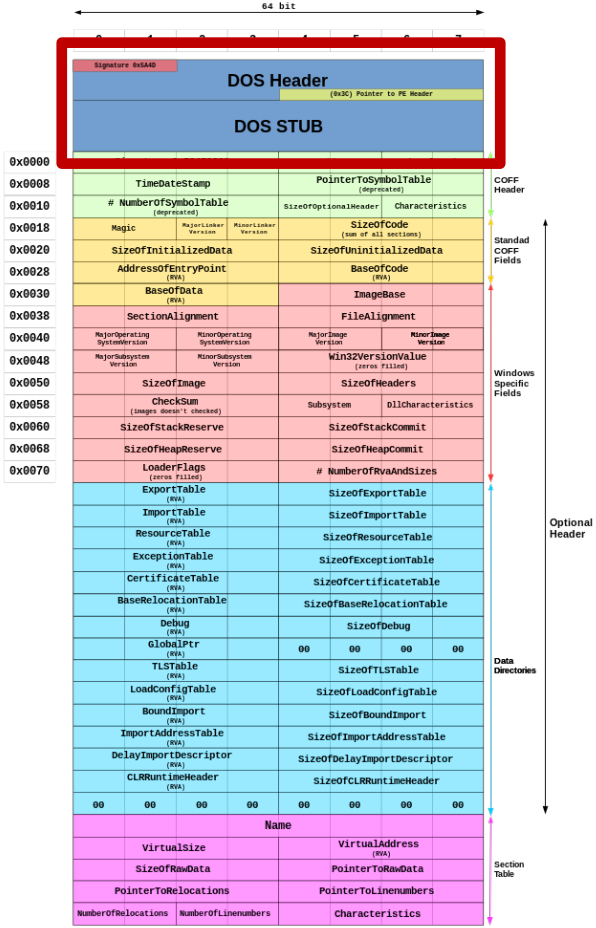


Windows PE File Format

DOS Header + Stub

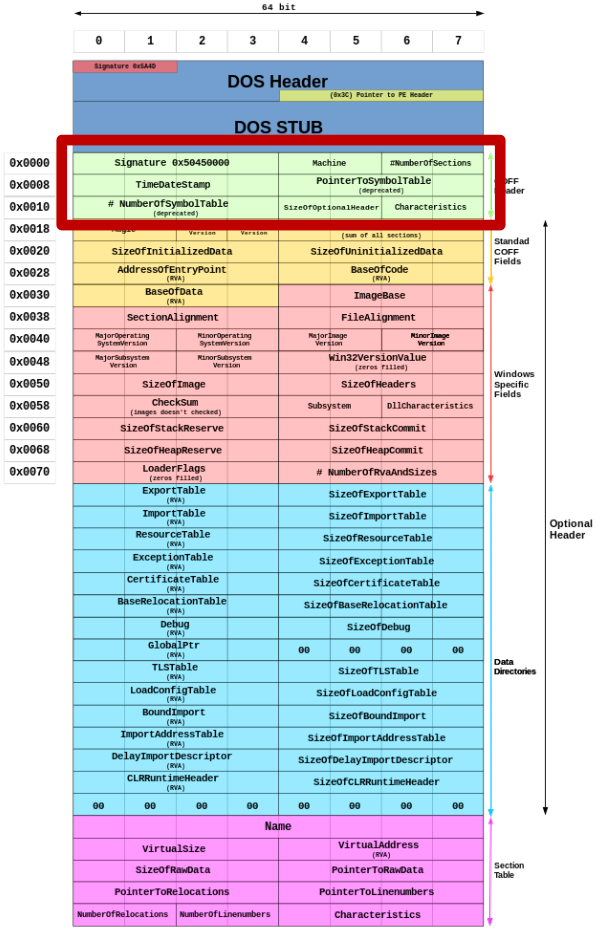
Metadata for DOS program

Executing a modern program in DOS will trigger the “This program cannot be run in DOS mode” output



Windows PE File Format

PE Header
Real metadata of the program
Dscribes general information of the file

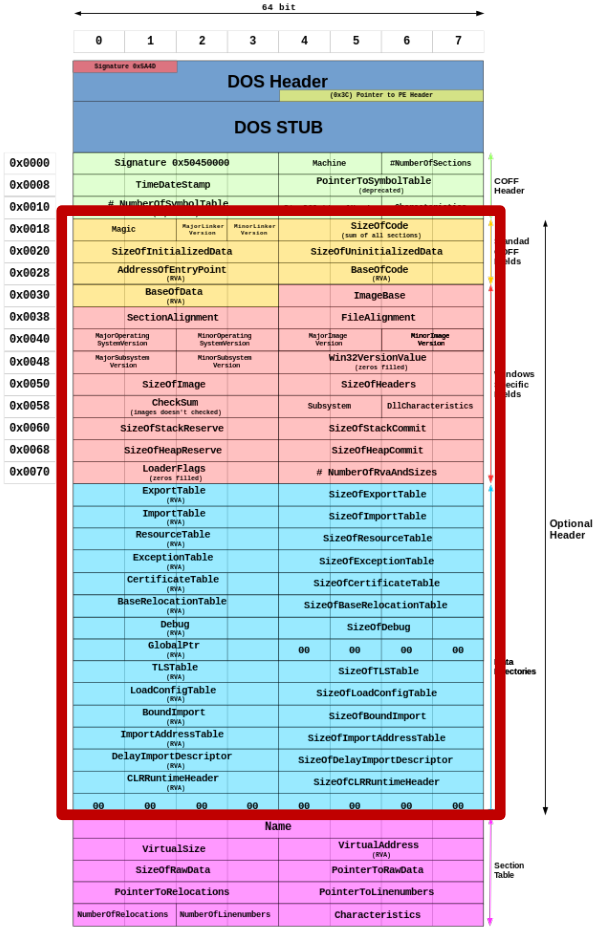


Windows PE File Format

Optional Header

Spoiler: not optional at all :)

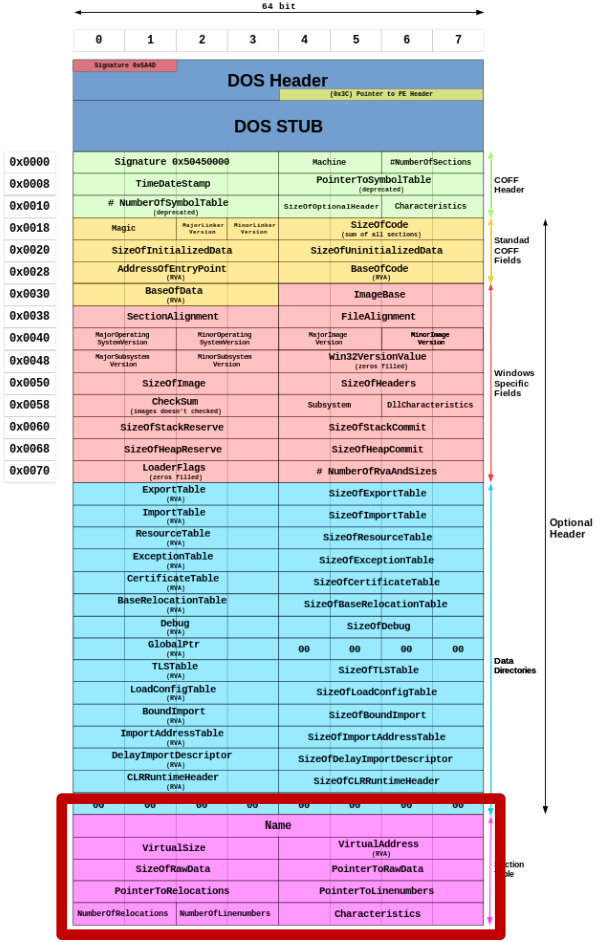
Instructs the loader where to find each object inside the file



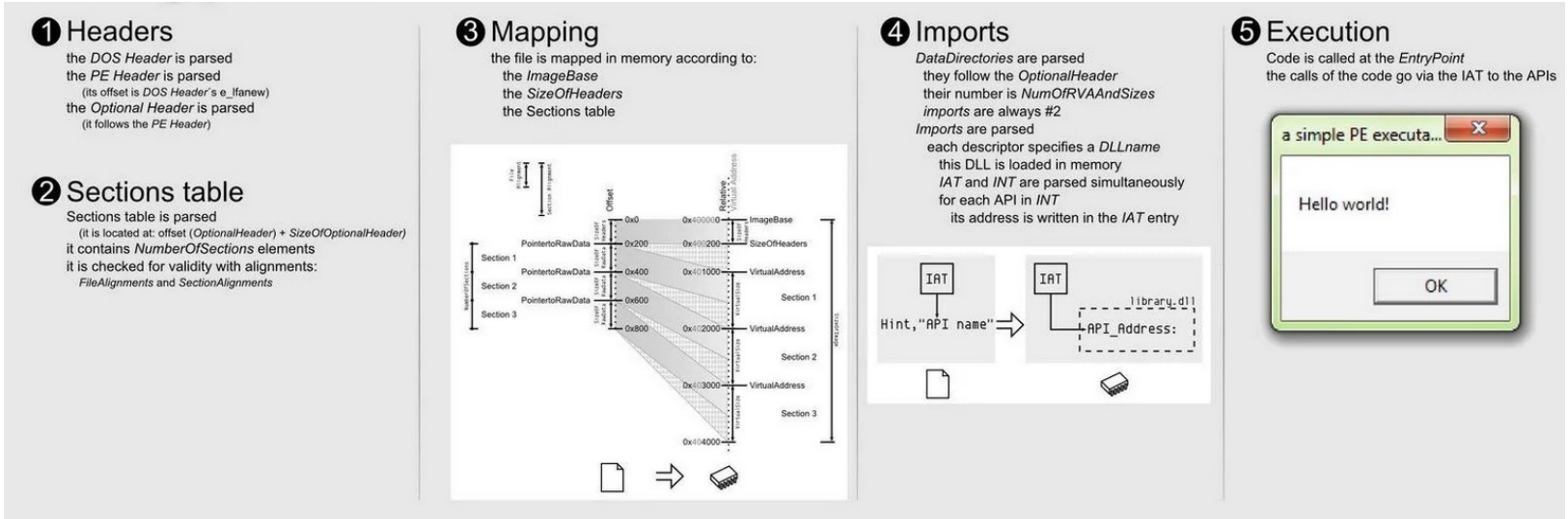
Windows PE File Format

Section Table and Sections
Describes where to find code, initialized data, resources, etc to the loader
These are “sections”, and each has a “section entry” with its characteristics

Examples: code is “.text”, read-only data is “.rodata”, resources are “.rsc”, and counting



How programs are loaded



https://code.google.com/archive/p/corkami/wikis/PE101.wiki

(Thanks Ange Albertini)

Practical Manipulations

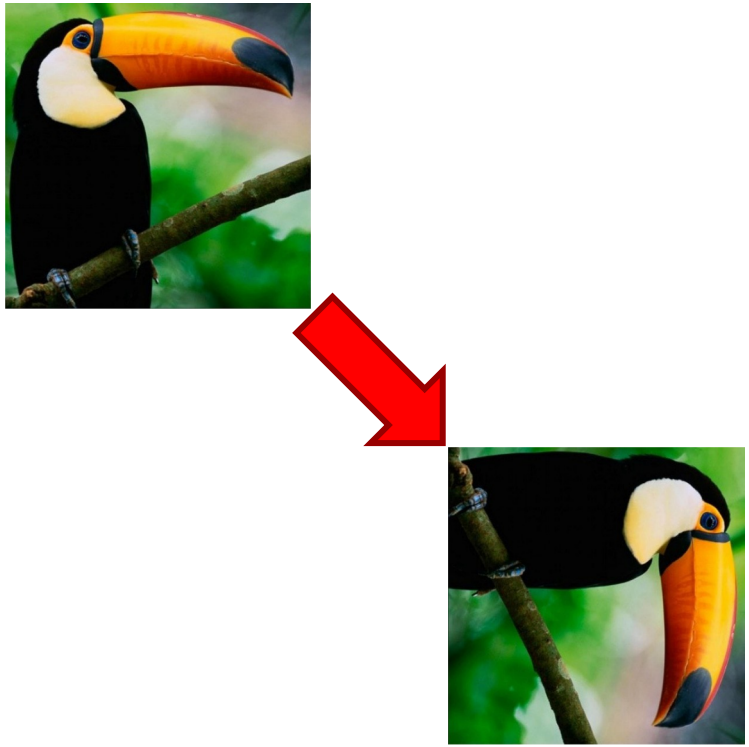
Practical Manipulations

Perturb the representation of a file

Keep intact the original functionality

Example: rotation for images

How to bridge the gap?



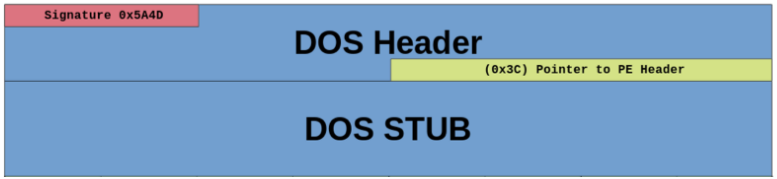
Demetrio et al., *Adversarial EXEmples: a Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection*, ACM TOPS 2021

DOS header perturbations

The attacker edit as much bytes as they want

Untouched: magic number MZ and offset to real PE header

Content loaded in memory, not executed



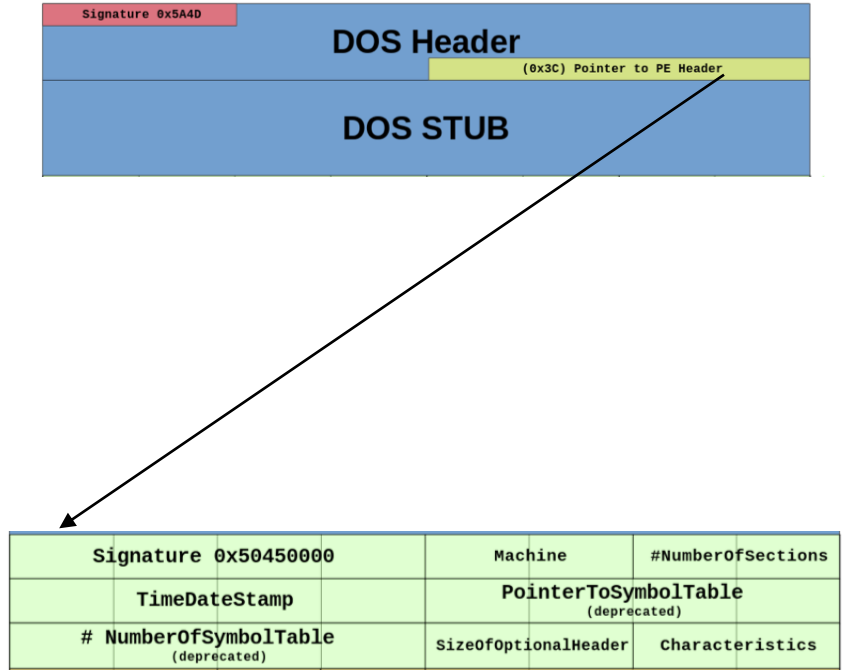
[Demetrio, Biggio et al., Adversarial EXEmples: A Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection ACM TOPS 2021]

DOS header extension

Exploit offset to real header, increment value

Insert arbitrary content between DOS header and PE header

Content loaded into memory, not executed



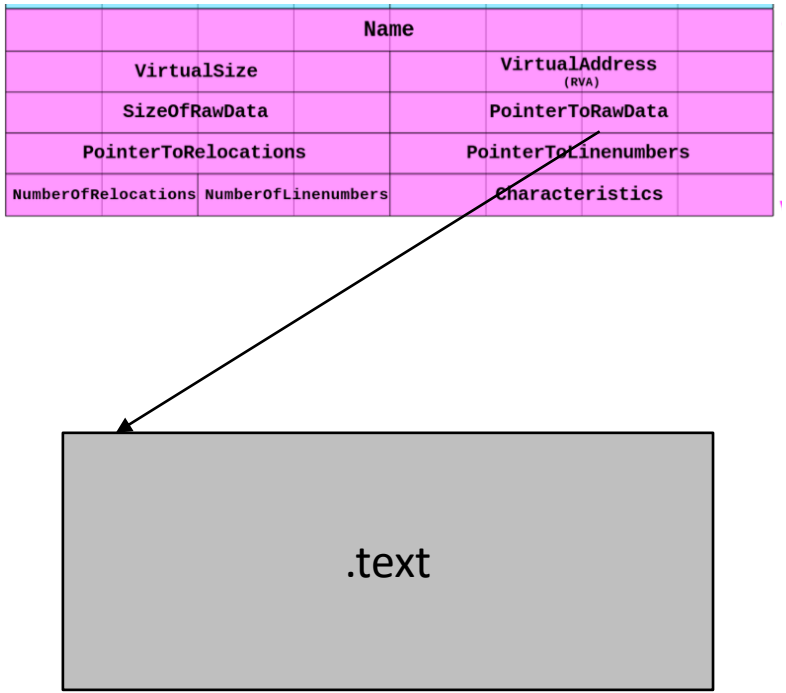
[Demetrio, Biggio et al., Adversarial EXEmples: A Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection ACM TOPS 2021]

Content shifting

Exploit offset in section entry, increment to manipulate the loader in searching for section content

The attacker can inject content after the section table, or between sections

NOT LOADED IN MEMORY, skipped by the loader



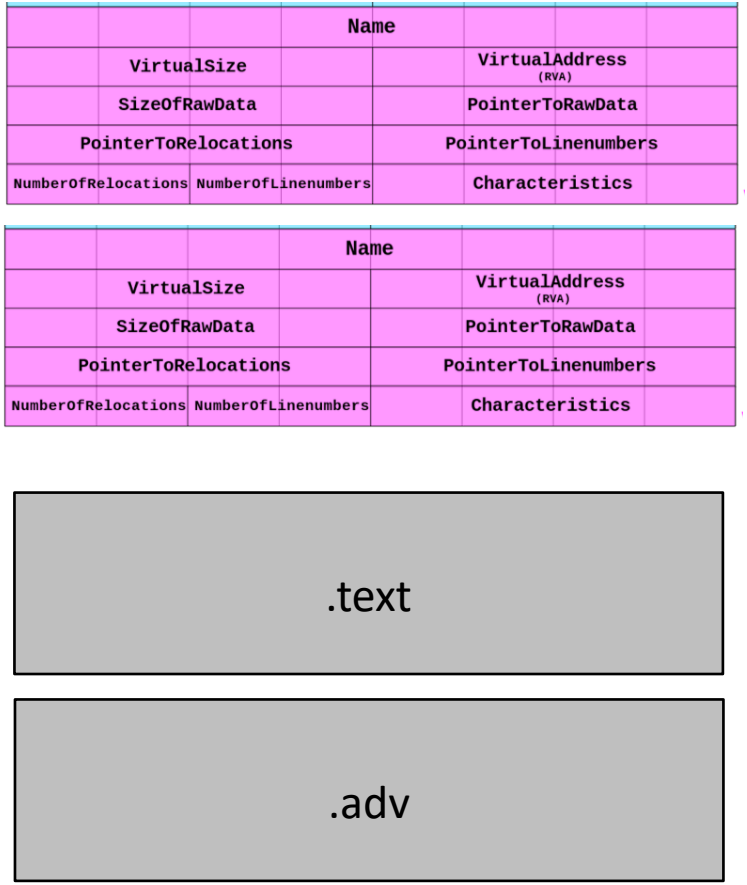
[Demetrio, Biggio et al., Adversarial EXEmples: A Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection ACM TOPS 2021]

Section Injection

Manipulate section table to add new entry

Append chunk of bytes, referenced by newly added entry

Loaded in memory or not, depending by the characteristics set up inside the entry



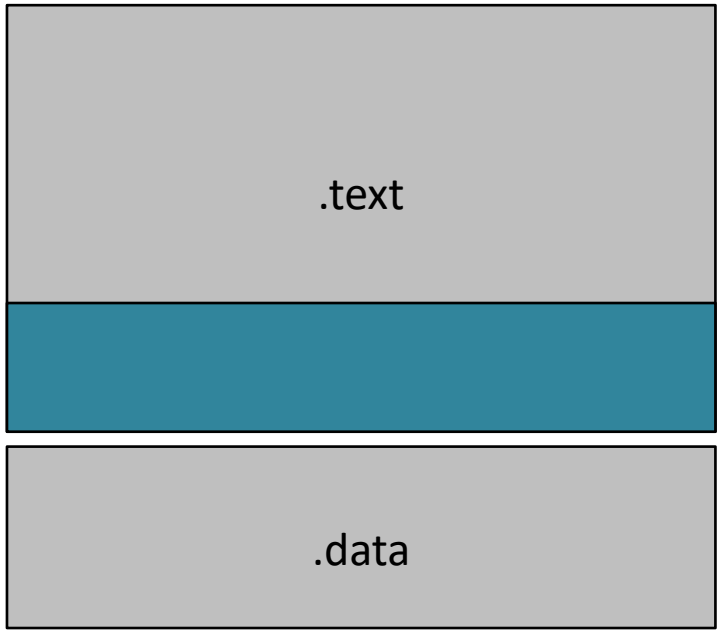
[Demetrio, Biggio et al., Adversarial EXEmples: A Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection ACM TOPS 2021]

Slack space

Section content is padded with 0 to keep file alignments

The attacker can rewrite such slack space

Loaded in memory, not executed



[Demetrio, Biggio et al., Adversarial EXEmples: A Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection ACM TOPS 2021]

Padding

Appending content at the end

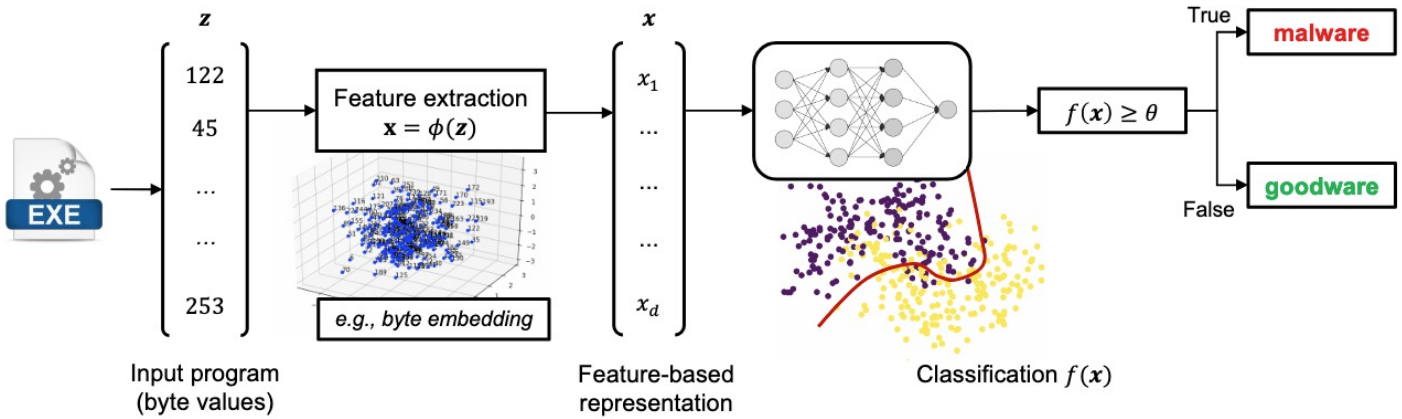
Most trivial manipulation

Not loaded in memory

[Demetrio, Biggio et al., Adversarial EXEmples: A Survey and Experimental Evaluation of Practical Attacks on Machine Learning for Windows Malware Detection ACM TOPS 2021]

Gradient-based attacks

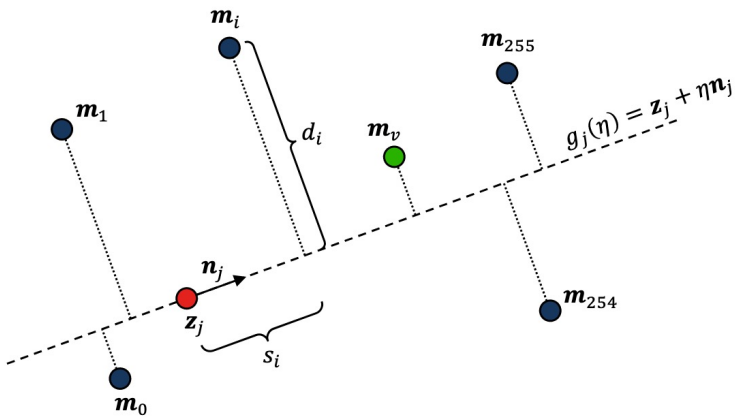
No differentiability end-to-end



No distance metrics == no end-to-end learning
Always feature extraction or embedding space

Gradients computed up to first differentiable part

Optimization byte-per-byte

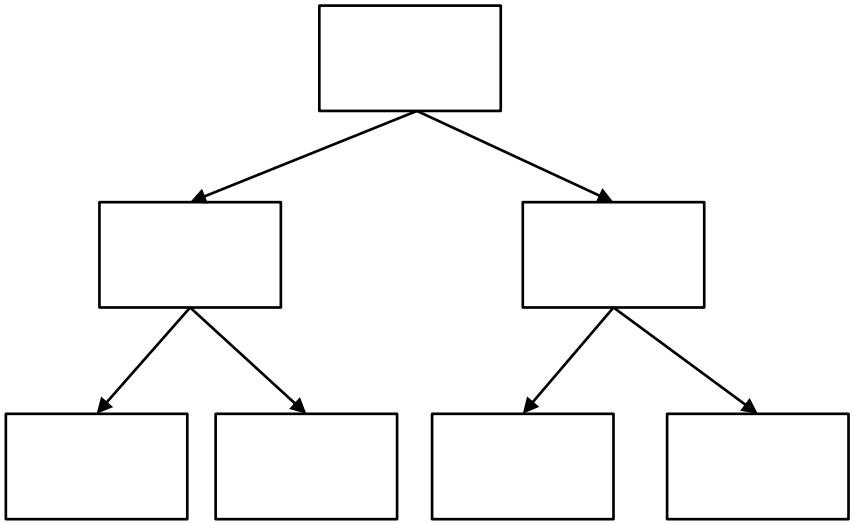


1. Compute gradient in feature space
2. Define a way for replacing values
   For bytes: inverse look-up of embedding
3. Follow the direction of gradient and replace byte with other byte

[Kolosnjaji, Biggio et al., Adversarial malware binaries: Evading deep learning for malware detection in executables EUSIPICO 2018]

Gradient-free attacks

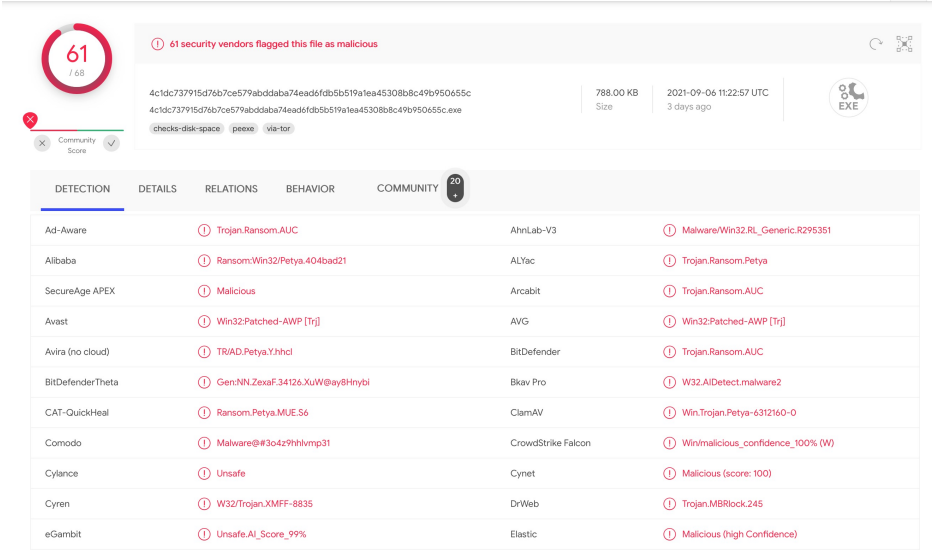
Reality check: robust models are not differentiable



State-of-the-art classifiers use decision trees

No gradients!

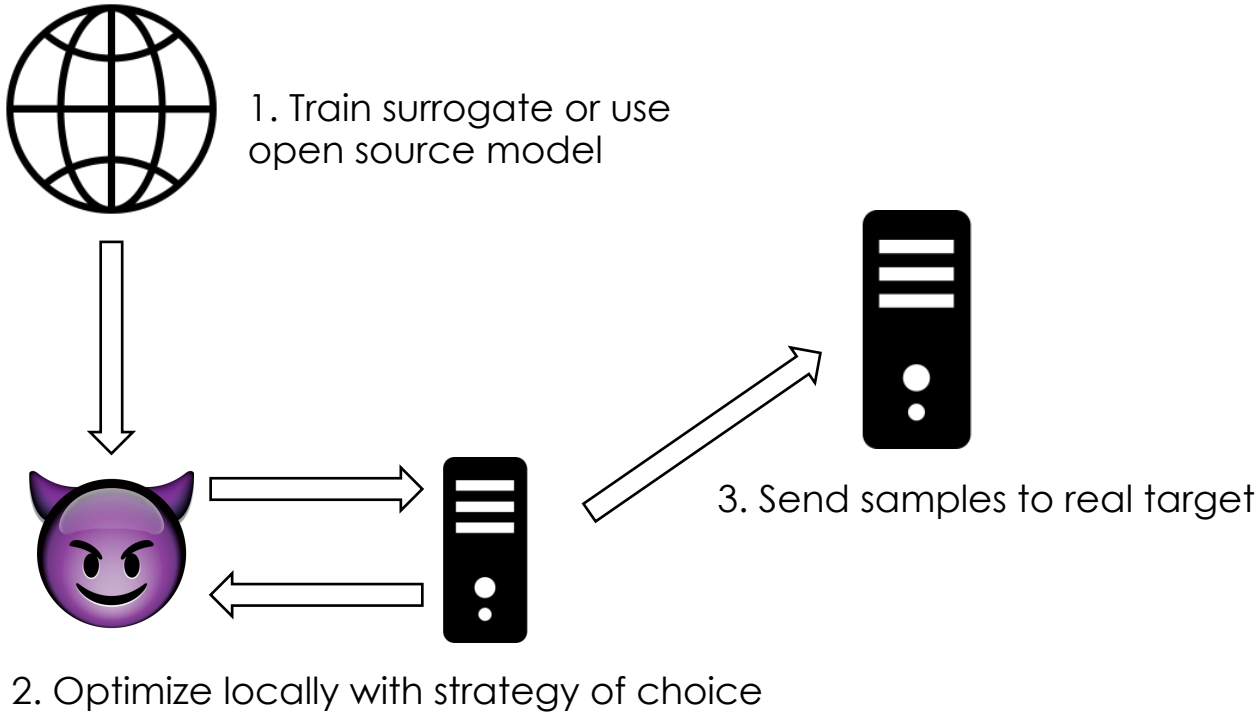
Reality check: most models are unavailable



Most models are hosted on private servers

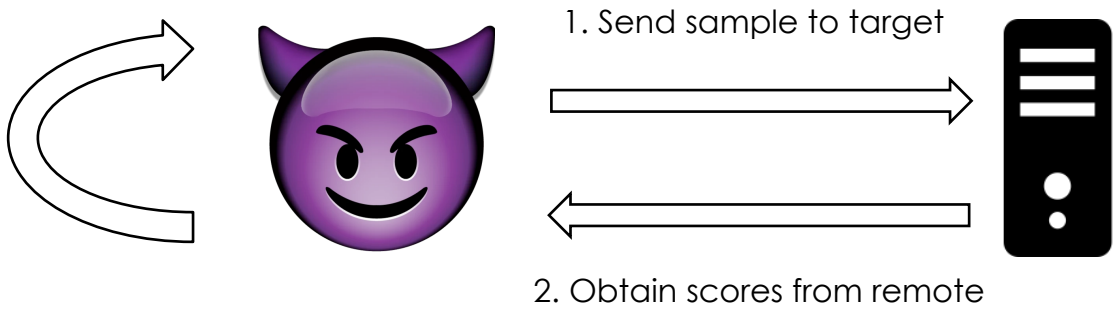
Detection performed in cloud

Transfer attacks



[Demetrio, Biggio et al., Functionality-preserving Black-box Optimization of Adversarial Windows Malware IEEE TIFS 2021]

Query attacks



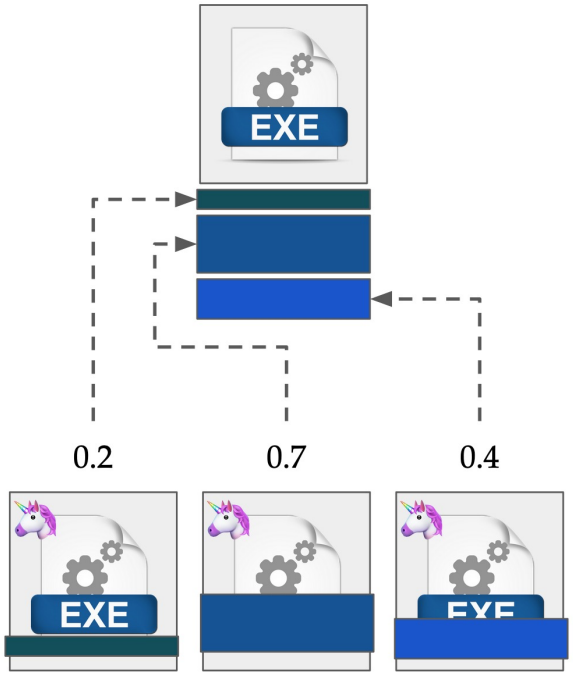
Very slow if optimizer works byte-per-byte

[Demetrio, Biggio et al., Functionality-preserving Black-box Optimization of Adversarial Windows Malware IEEE TIFS 2021]

Speeding up by injecting benign content

Inject portions of benign programs

Less to explore, by injecting already-known byte patterns



[Demetrio, Biggio et al., Functionality-preserving Black-box Optimization of Adversarial Windows Malware IEEE TIFS 2021]

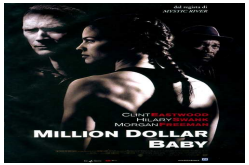
Hands-on lab on malware!

Luca Demetrio
luca.demetrio93@unica.it
@zangobot

Thanks!

If you know the enemy and know yourself, you need not fear the result of a hundred battles
Sun Tzu, The art of war, 500 BC

Countering Evasion Attacks



What is the rule? The rule is protect yourself at all times (from the movie "Million dollar baby", 2004)

Security Measures for Machine Learning

B. Biggio, F. Roli / Pattern Recognition 84 (2018) 317–331

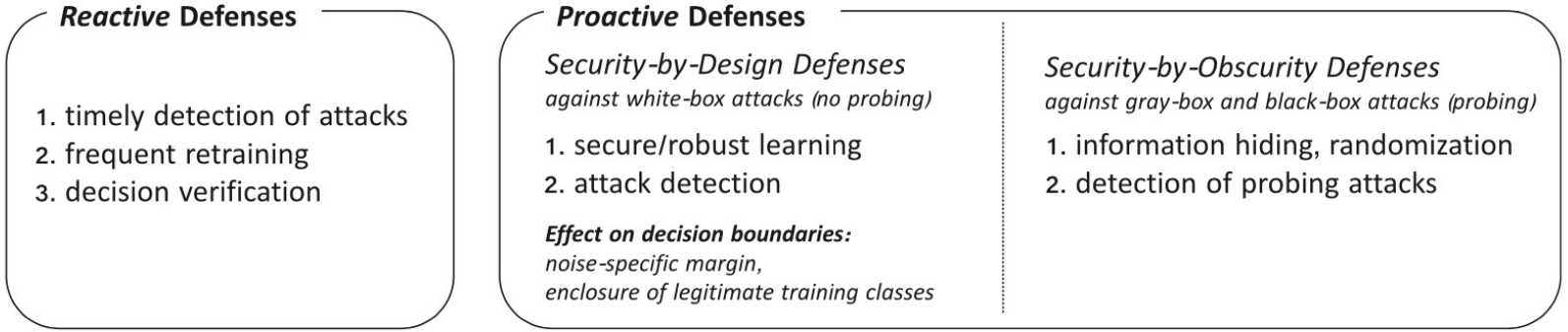


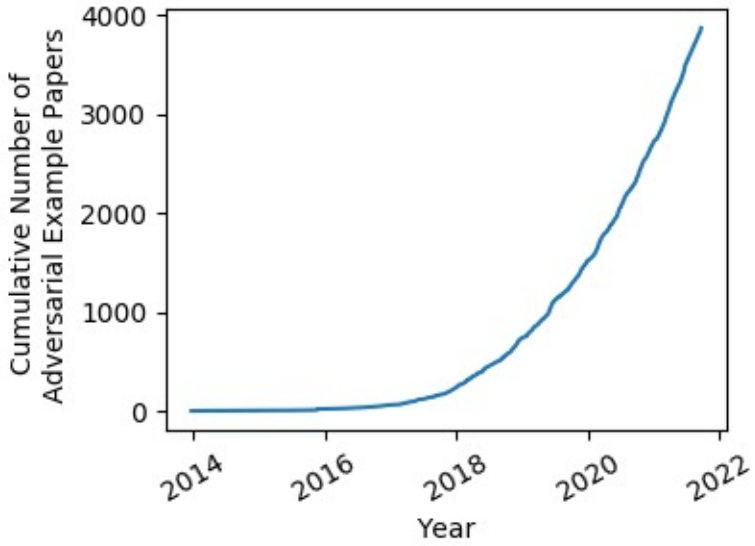
Fig. 11. Schematic categorization of the defense techniques discussed in Section 5.

Wild patterns: Ten years after the rise of adversarial machine learning

Battista Biggio[a,b,*], Fabio Roli[a,b]

A Challenging Problem!

- https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html



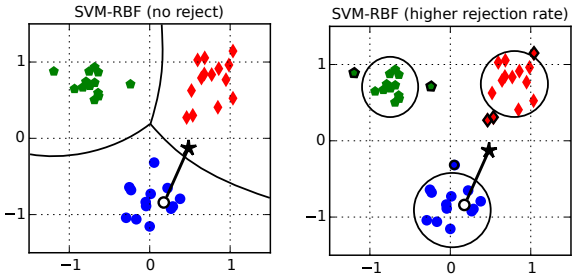
Security Measures against Evasion Attacks

1. Reduce sensitivity to input changes with **robust optimization**
   - Adversarial Training / Regularization

$$\min_{\boldsymbol{w}} \sum_i \max_{||\boldsymbol{\delta}_i|| \leq \epsilon} \ell(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i))$$

bounded perturbation!

2. Introduce ***rejection / detection*** of adversarial examples



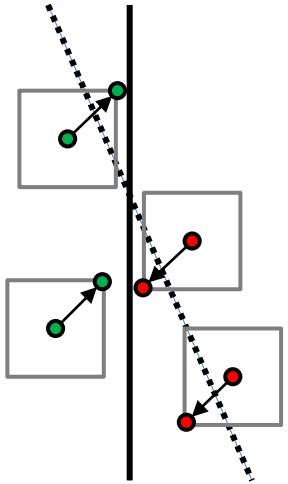
Countering Evasion:
Robust Optimization / Adversarial Training

Robust Optimization via Adversarial Training (AT)

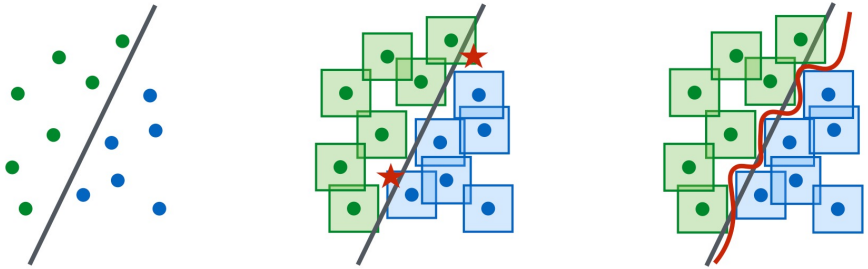
- Robust optimization (a.k.a. *adversarial training*)

$$\min_{\boldsymbol{w}} \max_{||\boldsymbol{\delta}_i||_\infty \leq \epsilon} \sum_i \ell\left(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i)\right)$$

bounded perturbation!



- Madry et al., ICLR 2018 (https://arxiv.org/pdf/1706.06083.pdf)
  - PGD-AT better than FGSM-AT but more computationally costly
  - Fast AT (NeurIPS 2020, https://arxiv.org/abs/2007.02617)

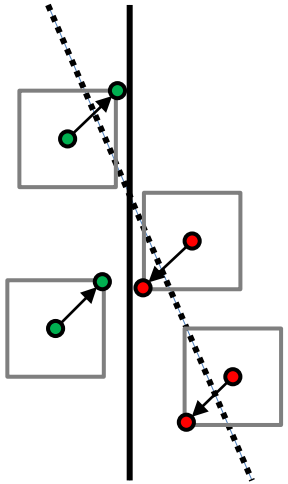


Robust Optimization via Regularization

- Robust optimization (a.k.a. *adversarial training*)

$$\min_{\boldsymbol{w}} \max_{||\boldsymbol{\delta}_i||_\infty \leq \epsilon} \sum_i \ell\left(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i)\right)$$

bounded perturbation!



- Robustness and regularization (Xu et al., JMLR 2009)
  - under linearity of ℓ and $f_{\boldsymbol{w}}$, equivalent to robust optimization

$$\min_{\boldsymbol{w}} \sum_{\boldsymbol{i}} \ell\left(y_i, f_{\boldsymbol{w}}(\boldsymbol{x}_i)\right) + \epsilon||\nabla_{\boldsymbol{x}} f||_1$$

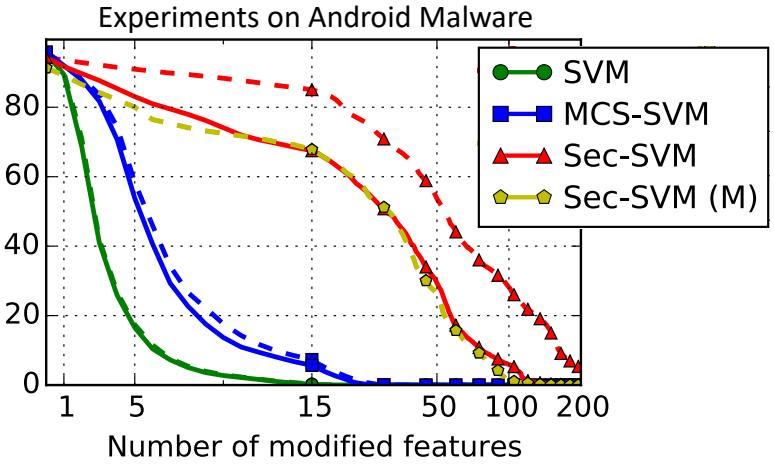
dual norm of the perturbation

$||\nabla_{\boldsymbol{x}} f||_1 = ||\boldsymbol{w}||_1$

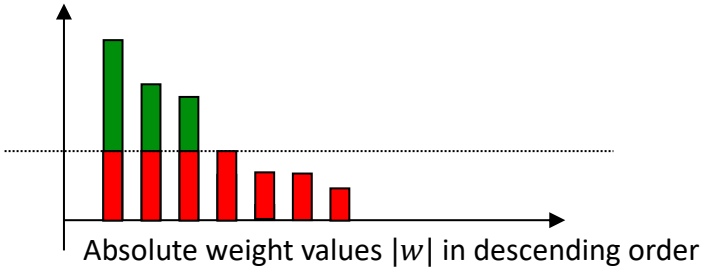
Results on *Adversarial* Android Malware

- **Infinity-norm regularization** is the optimal regularizer against **sparse evasion attacks**
  - Sparse evasion attacks penalize $||\boldsymbol{\delta}||_1$ promoting the manipulation of only few features

Sec-SVM $$\min_{w,b} \|w\|_\infty + C\sum_i \max\left(0, 1 - y_i f(x_i)\right), \quad \|w\|_\infty = \max_{i=1,\ldots,d} |w_i|$$



Why? It bounds the maximum weight absolute values!



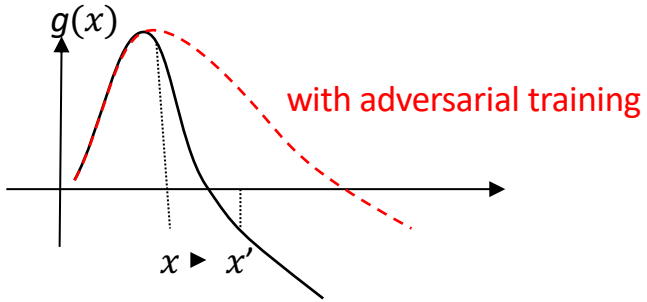
[Demontis, Biggio et al., Yes, ML Can Be More Secure!..., IEEE TDSC 2017]

Adversarial Training and Regularization

- Adversarial training can also be seen as a form of regularization, which penalizes the (dual) norm of the input gradients $\epsilon||\nabla_{\mathbf{x}}\ell||_{\mathbf{q}}$

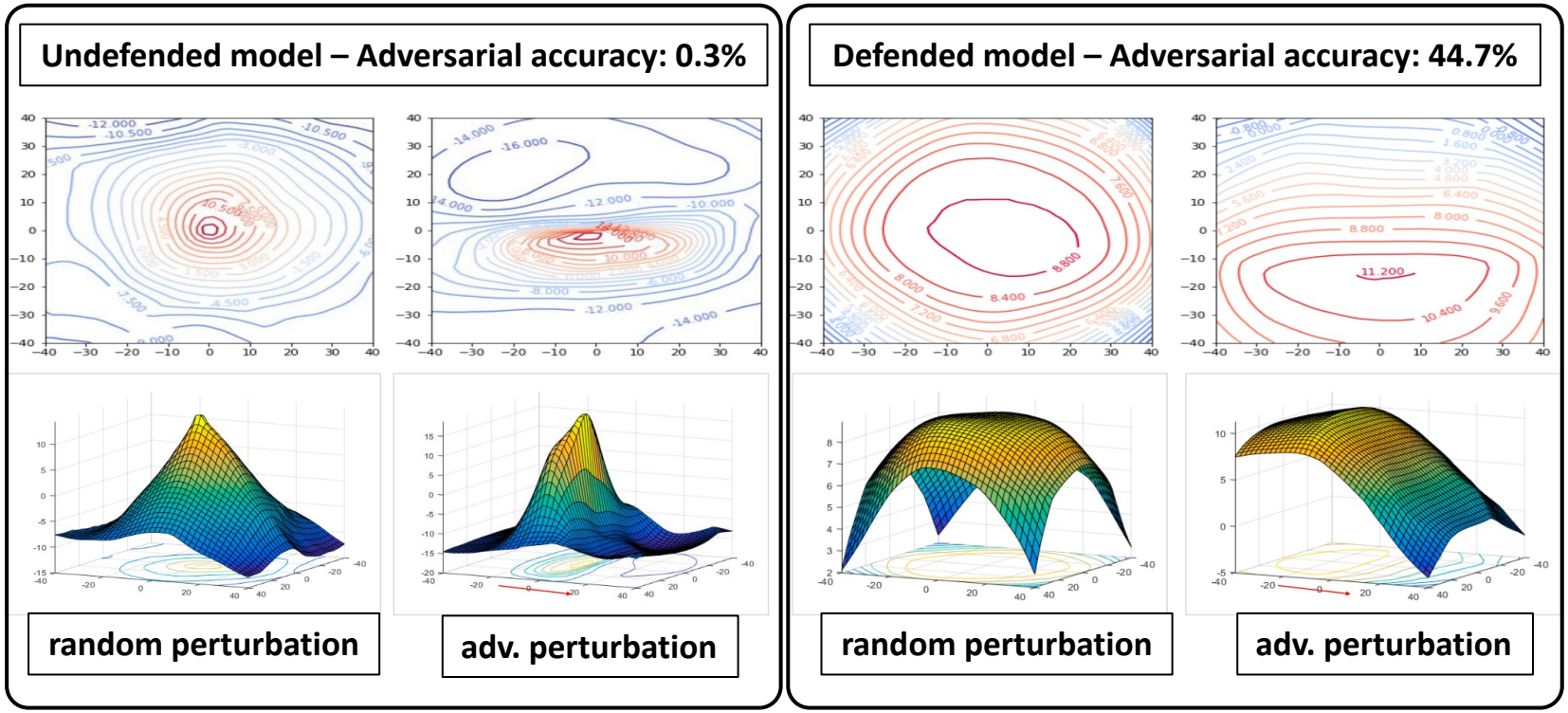
- Known as double backprop or gradient/Jacobian regularization
  - see, e.g., *Simon-Gabriel et al., Adversarial vulnerability of neural networks increases with input dimension, ArXiv 2018*; and *Lyu et al., A unified gradient regularization family for adversarial examples, ICDM 2015*.

Take-home message: the net effect of these techniques is to make the prediction function of the classifier smoother (increasing the input margin)



Why Does Robust Optimization Work?

CIFAR-10



Yu et al., Interpreting and Evaluating NN Robustness, IJCAI 2019

On Adversarial Training...

2004

Adversarial Classification

Nilesh Dalvi Pedro Domingos Mausam Sumit Sanghai Deepak Verma
Department of Computer Science and Engineering
University of Washington, Seattle
Seattle, WA 98195-2350, U.S.A.
{nilesh,pedrod,mausam,sanghai,deepak}@cs.washington.edu

2006

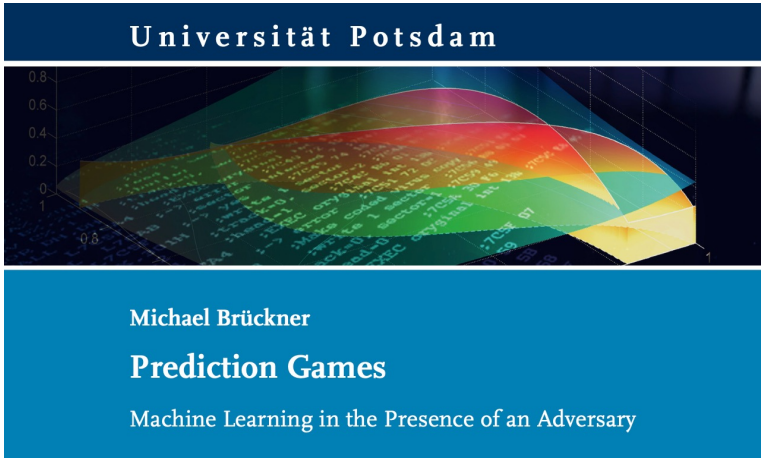
Nightmare at Test Time: Robust Learning by Feature Deletion

Amir Globerson GAMIR@CSAIL.MIT.EDU
Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA
Sam Roweis ROWEIS@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Canada

2012

Static Prediction Games for Adversarial Learning Problems

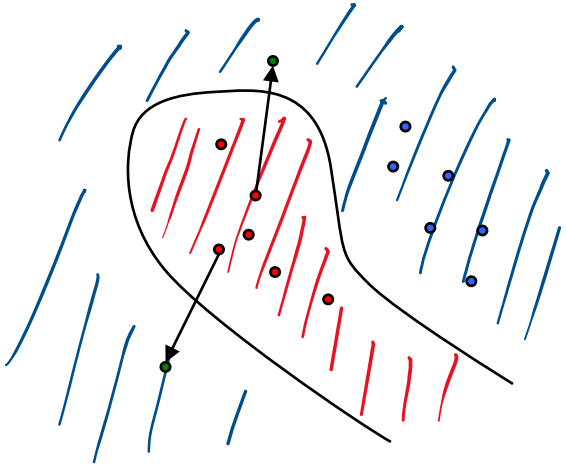
Michael Brückner MIBRUECK@CS.UNI-POTSDAM.DE
Department of Computer Science
University of Potsdam
August-Bebel-Str. 89
14482 Potsdam, Germany
Christian Kanzow KANZOW@MATHEMATIK.UNI-WUERZBURG.DE
Institute of Mathematics
University of Würzburg
Emil-Fischer-Str. 30
97074 Würzburg, Germany
Tobias Scheffer SCHEFFER@CS.UNI-POTSDAM.DE
Department of Computer Science
University of Potsdam
August-Bebel-Str. 89
14482 Potsdam, Germany



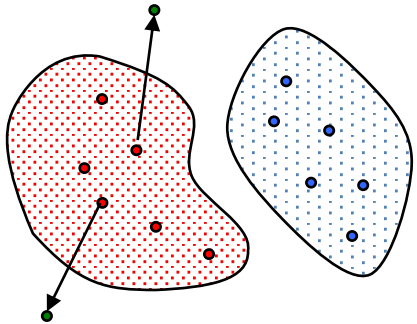
Countering Evasion:
Detecting & Rejecting Adversarial Examples

Detecting and Rejecting Adversarial Examples

- Adversarial examples tend to occur in *blind spots*
  - Regions far from training data that are anyway assigned to 'legitimate' classes

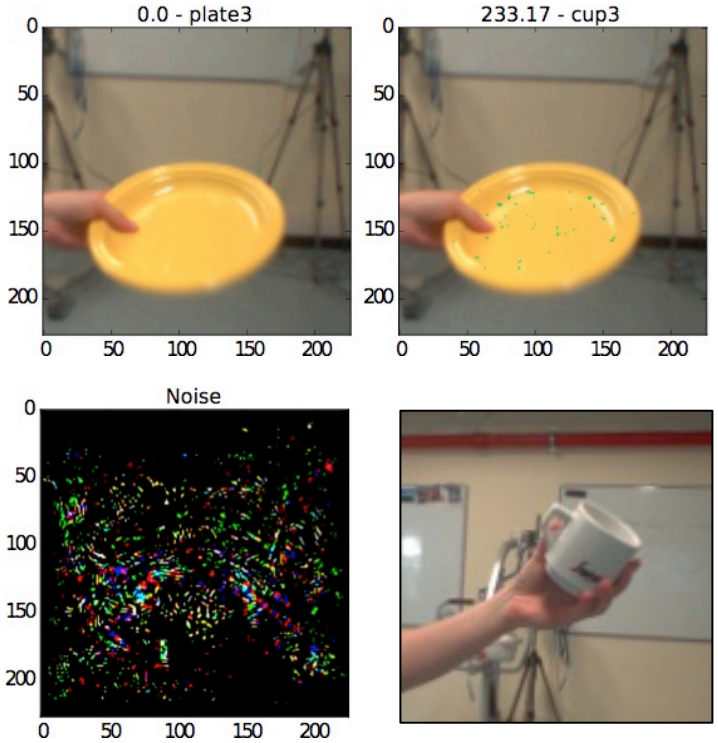


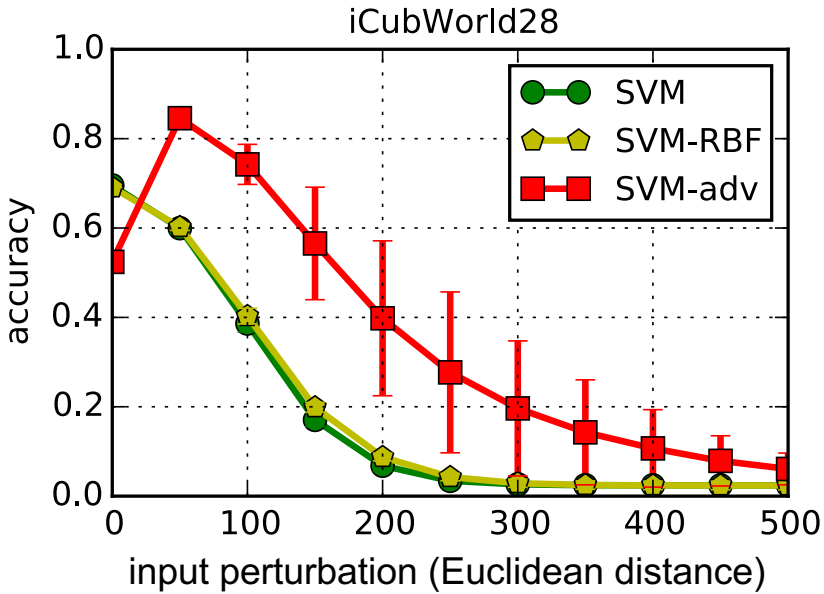
blind-spot evasion
(not even required to mimic the target class)



rejection of adversarial examples through enclosing of legitimate classes

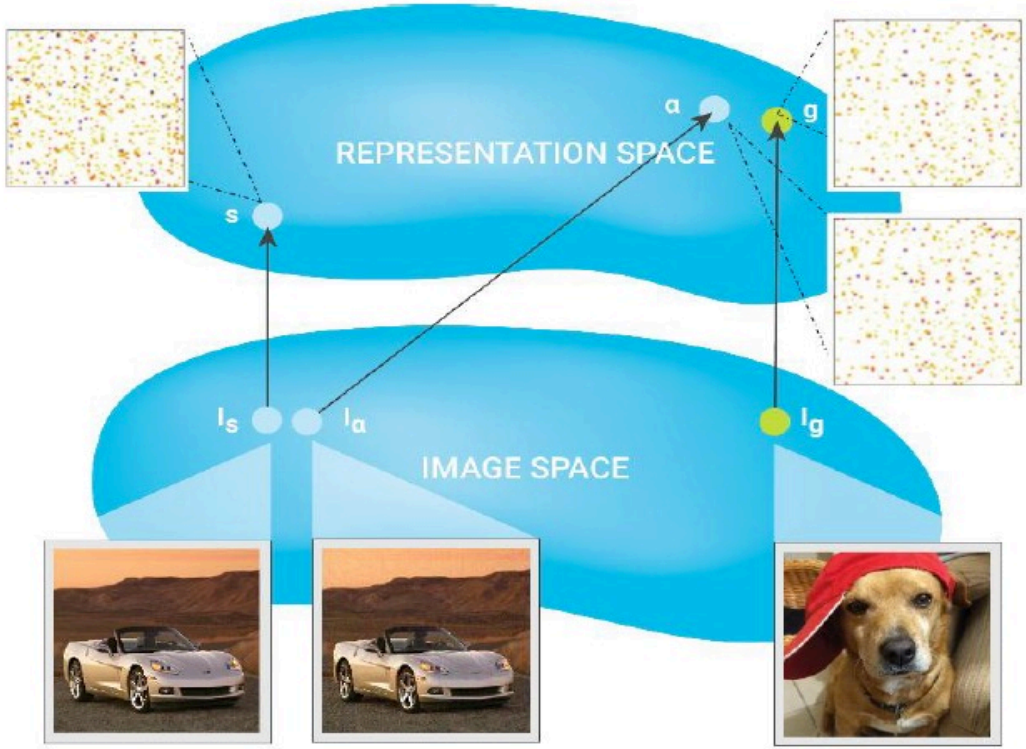
Detecting & Rejecting Adversarial Examples





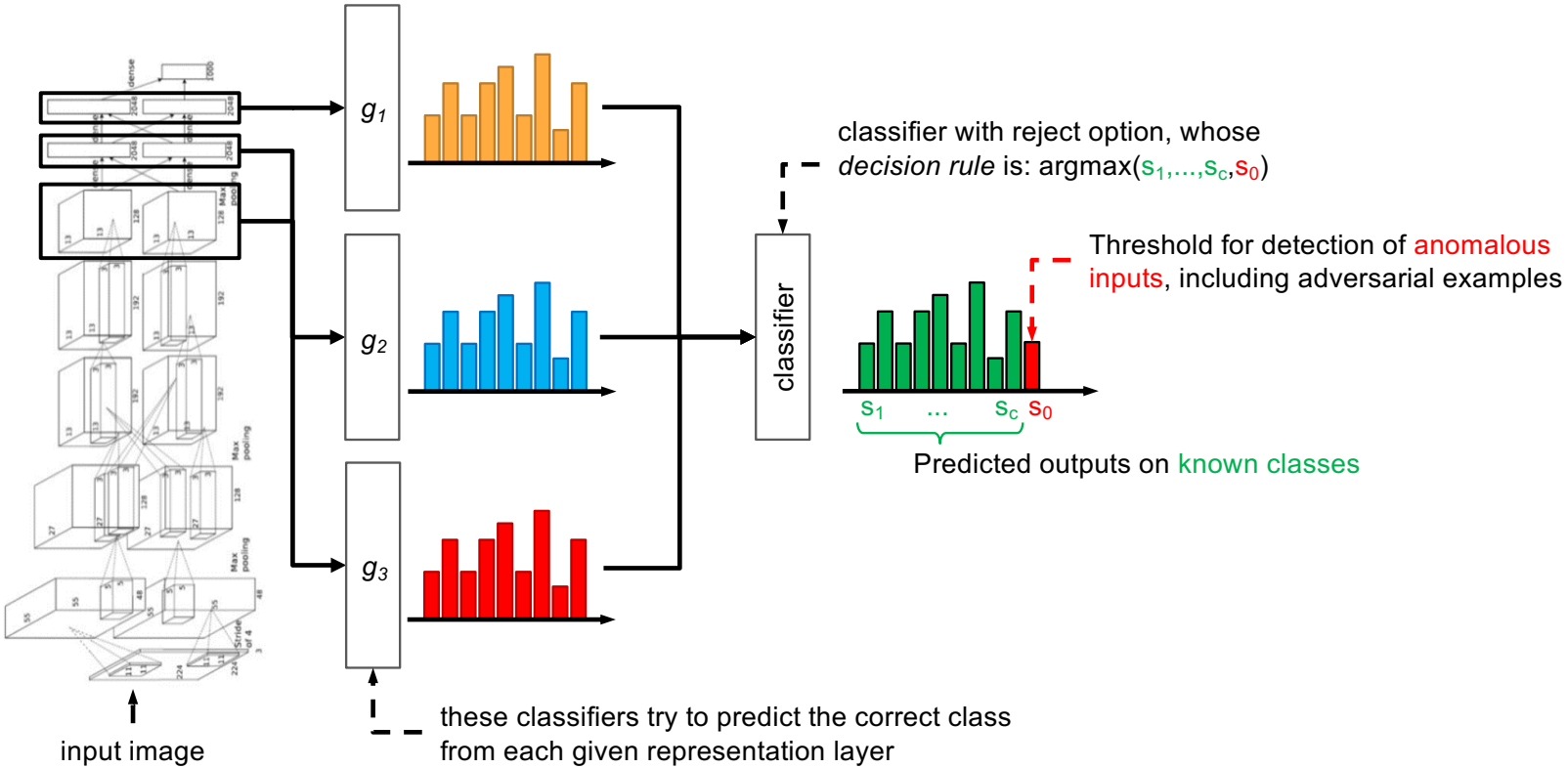
[Melis, Biggio et al., Is Deep Learning Safe for Robot Vision? ICCVW ViPAR 2017]

Why Rejection (in Representation Space) Is Not Enough?



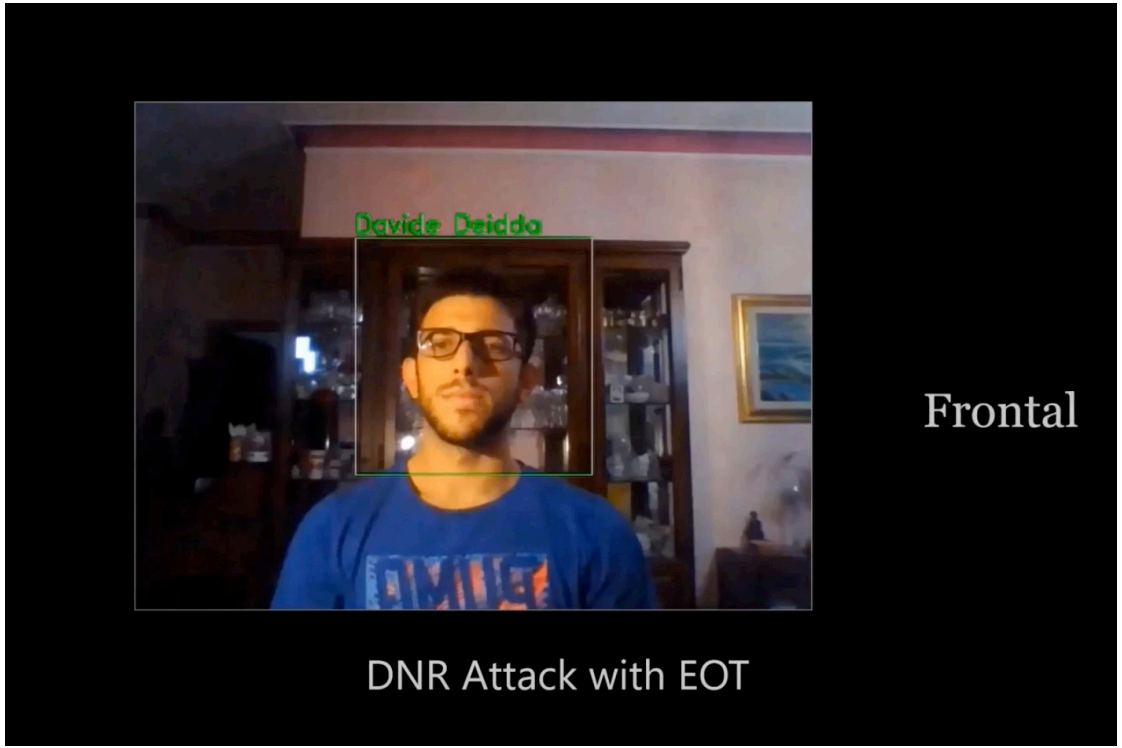
[S. Sabour at al., ICLR 2016]

Deep Neural Rejection against Adversarial Examples

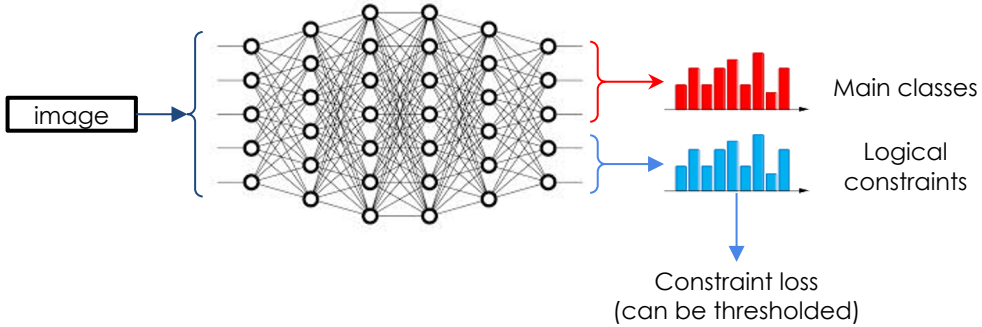


Sotgiu, Biggio et al., EURASIP JIS, 2020

Ongoing Work: DNR against Physical Attacks

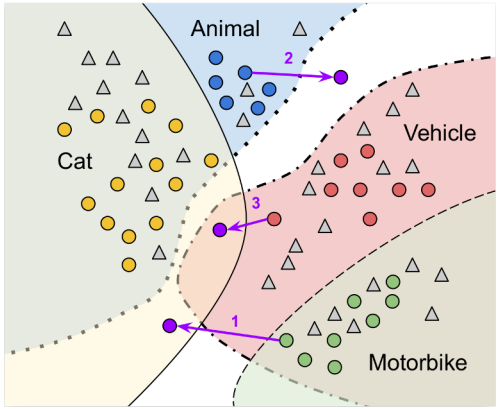


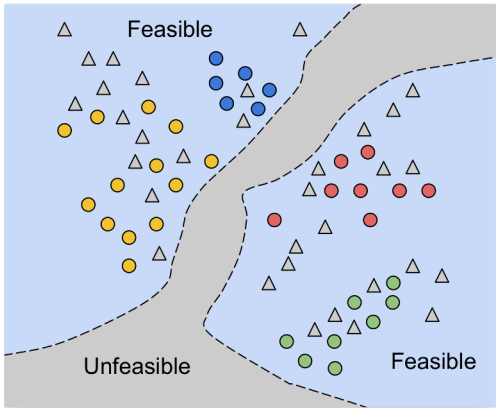
Robust Learning with Domain Knowledge



$$
\begin{aligned}
&\forall x, \quad \text{CAT}(x) \Rightarrow \text{ANIMAL}(x)\,,\\
&\forall x, \quad \text{MOTORBIKE}(x) \Rightarrow \text{VEHICLE}(x)\,,\\
&\forall x, \quad \text{VEHICLE}(x) \Rightarrow \neg\text{ANIMAL}(x)\,,\\
&\forall x, \quad \text{CAT}(x) \vee \text{ANIMAL}(x) \vee \text{MOTORBIKE}(x) \vee \text{VEHICLE}(x)
\end{aligned}
$$

$$
\min_{\mathbf{f}} = \frac{1}{n}\sum_{i=1}^{l} L_y(\mathbf{f}(\mathbf{x}_i), \mathbf{y}_i) + \sum_{j=1}^{l+u}\sum_{h=1}^{m} \lambda_m \cdot L_\phi(\phi_h(\mathbf{f}(\mathbf{x}_j))) + \lambda\|\mathbf{f}\|
$$

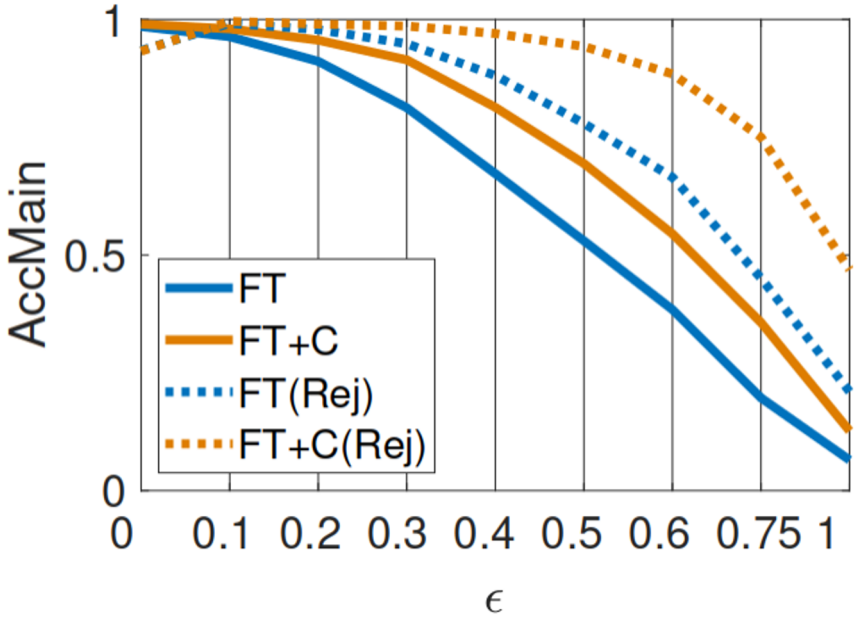




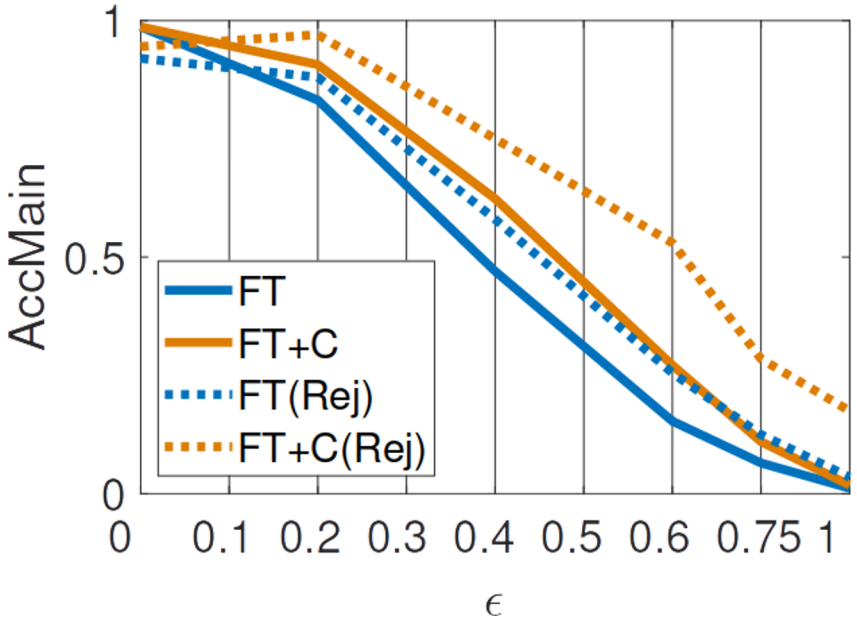
Melacci et al., https://arxiv.org/abs/2006.03833

Robust Learning with Domain Knowledge

Non adaptive attack
Imagenet Animals dataset



Adaptive attacks
Imagenet Animals dataset



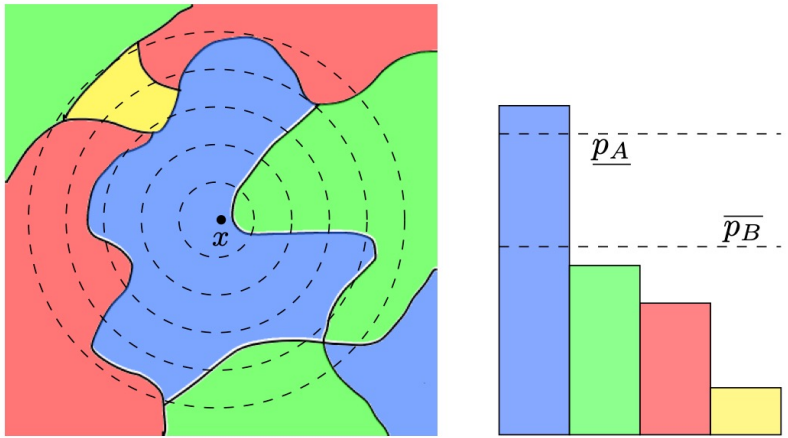
Melacci et al., https://arxiv.org/abs/2006.03833

Certified Defenses

Randomized Smoothing

Cohen et al., ICML 2019 https://arxiv.org/abs/1902.02918

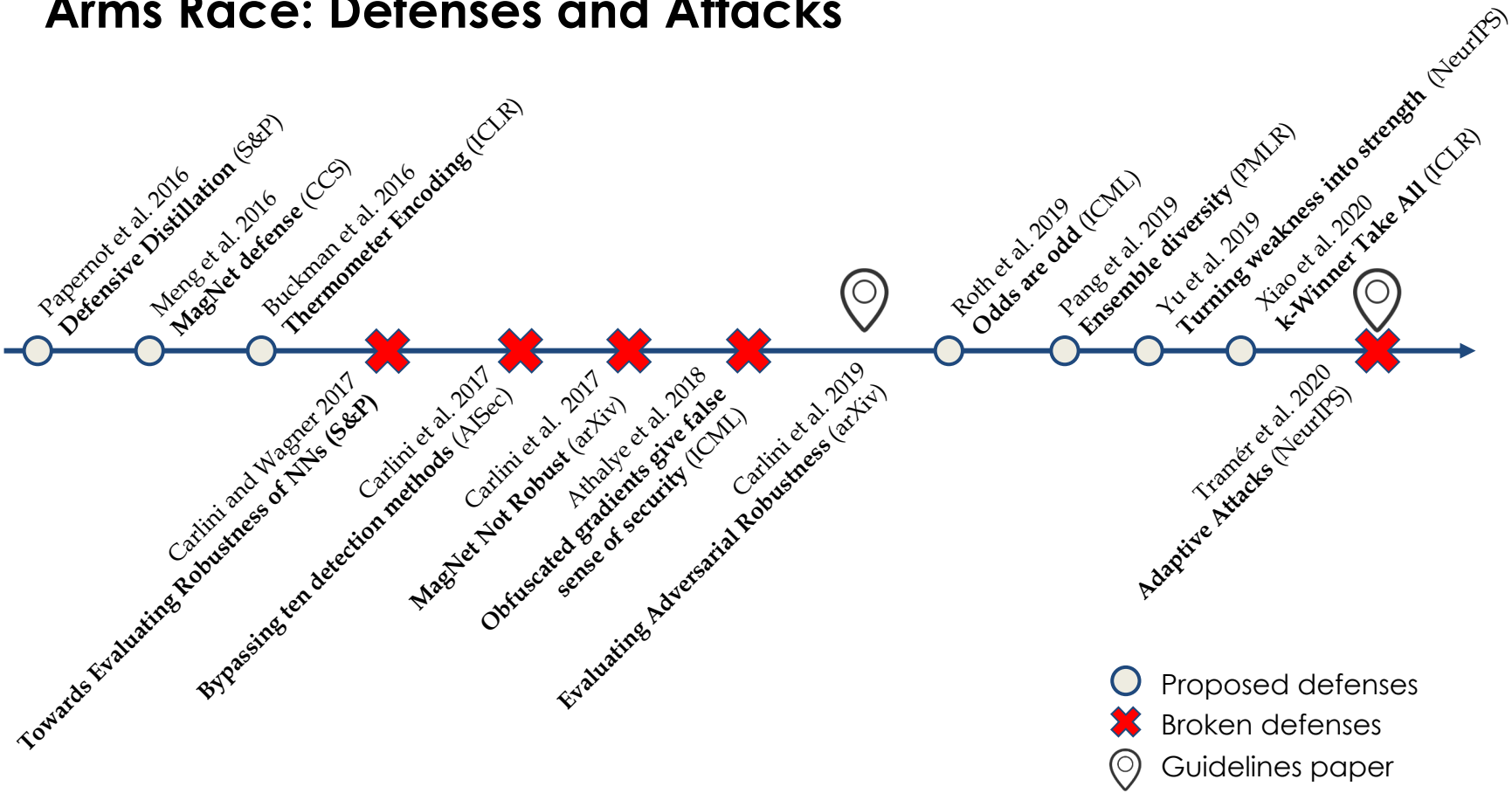
- Formal guarantee on adversarial robustness (a.k.a. *provable robustness*)
  - classification is consistent within l2 perturbations of size less than a given radius



Other relevant work on formal verification / certification of DNNs by: Martin Vechev, Marta Kwiatkowska, Zico Kolter

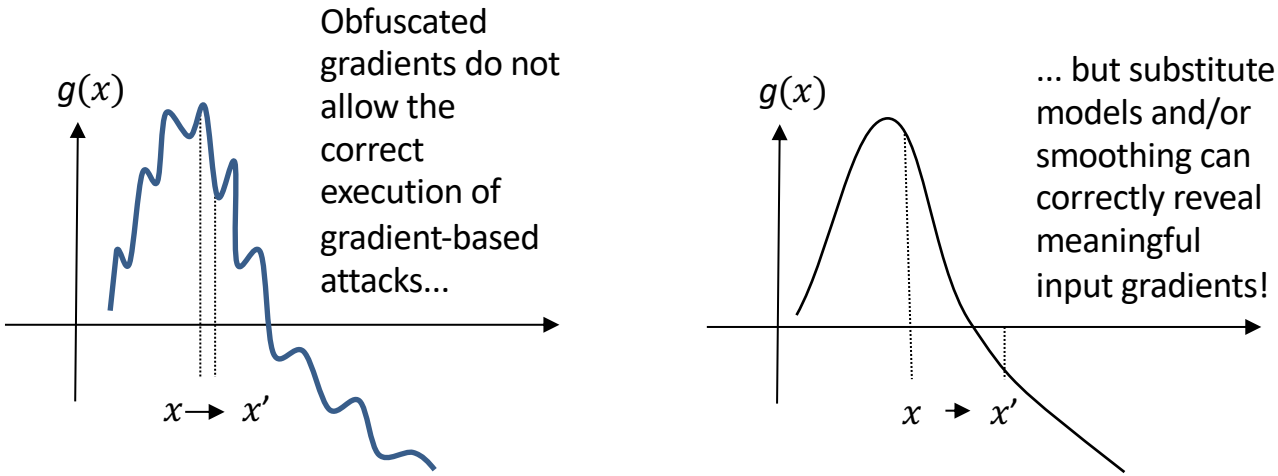
Ineffective Defenses

Arms Race: Defenses and Attacks



Ineffective Defenses: Obfuscated Gradients

- Carlini & Wagner (SP' 17), Athalye et al. (ICML '18), Tramer et al. (NeurIPS '20) have shown that
  - some recently-proposed defenses rely on obfuscated / masked gradients, and
  - they can be circumvented

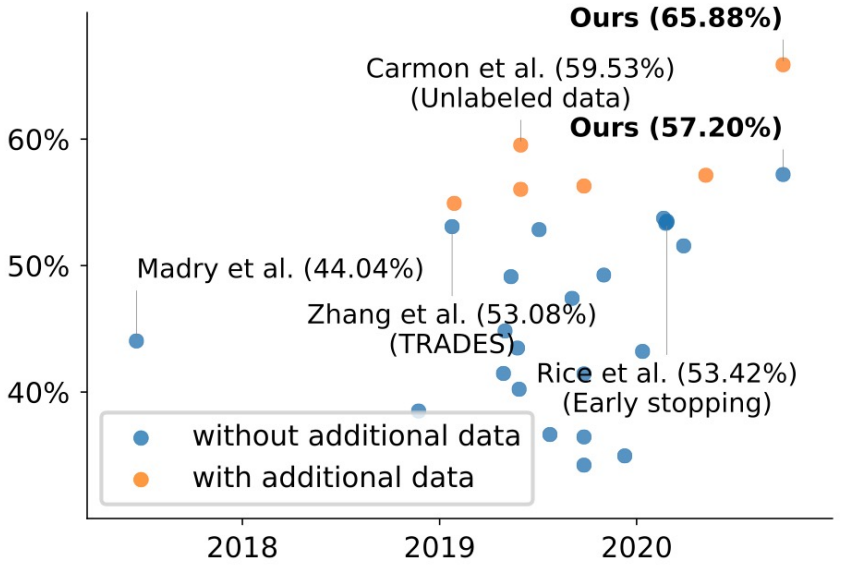


More on State-of-the-art Defenses

Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples

- Sven Gowal (Deepmind) et al., 2021
  https://arxiv.org/abs/2010.03593

- *«We discover that it is possible to train robust models that go well beyond state-of-the-art results by combining larger models, Swish/SiLU activations and model weight averaging.»*



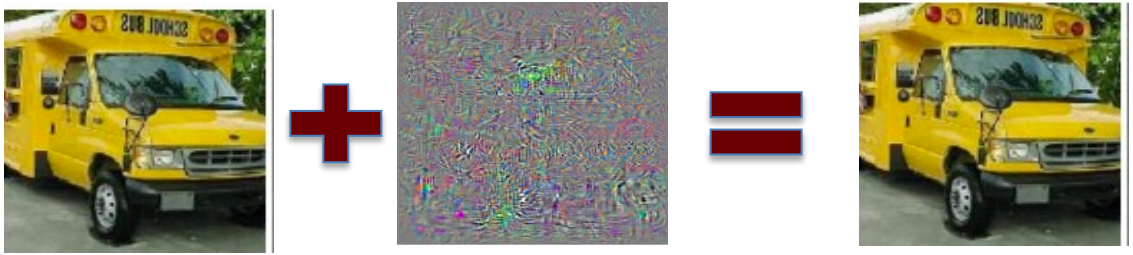
Extracted from RobustBench:
https://robustbench.github.io/

Are Adversarial Examples a Real Security Threat?

World Is Not Digital…

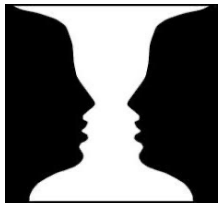
- *….**Previous** cases of **adversarial examples** have common characteristic: the **adversary** is able to **precisely control** the **digital representation** of the input to the machine learning tools…..*

[M. Sharif et al., ACM CCS 2016]



School Bus (*x*) **Adversarial Noise (*r*)** **Ostrich Struthio Camelus**

Do Adversarial Examples Exist in the Physical World?



Adversarial Images in the Physical World

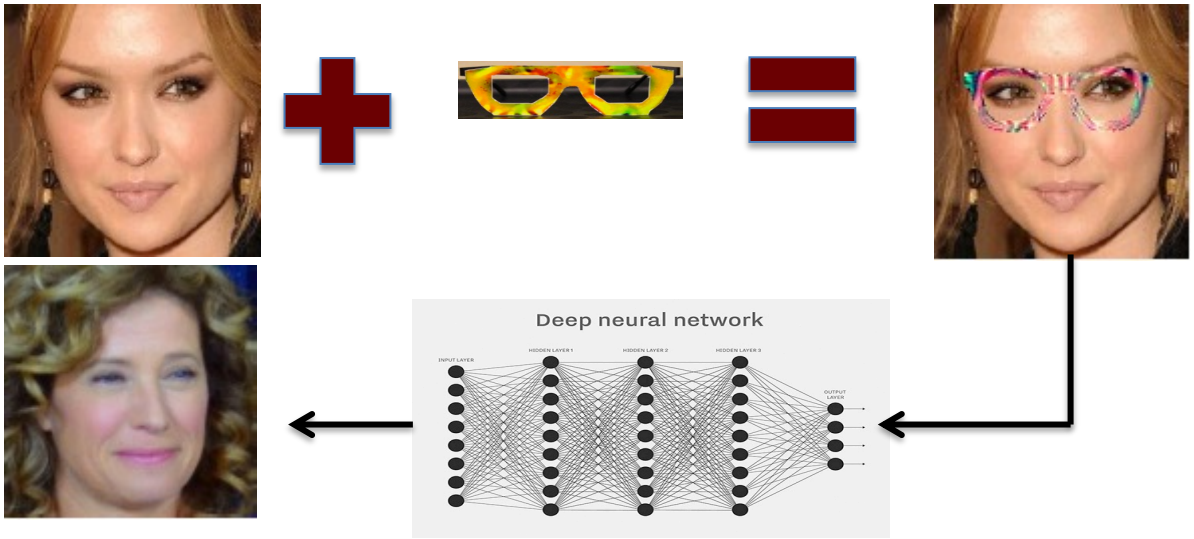
- Adversarial images fool deep networks **even when** they **operate** in the **physical world**, for example, **images** are **taken from** a **cell-phone camera**?
  - Alexey Kurakin et al. (2016, 2017) explored the possibility of creating adversarial images for machine learning systems which operate in the physical world. They used images taken from a cell-phone camera as an input to an Inception v3 image classification neural network
  - They showed that in such a set-up, a significant fraction of adversarial images crafted using the original network are misclassified even when fed to the classifier through the camera

[Alexey Kurakin et al., ICLR 2017]

Adversarial Glasses

$$\underset{r}{\operatorname{argmin}} \Big(\big(\sum_{x \in X} softmaxloss(x + r, c_t) \big) + \kappa_1 \cdot TV(r) + \kappa_2 \cdot NPS(r) \Big)$$

The adversarial perturbation is applied only to the eyeglasses image region



[M. Sharif et al., ACM CCS 2016]

Should We Be Worried ?

No, We Should Not…

[arXiv:1707.03501; CVPR 2017]

NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles

Jiajun Lu*, Hussein Sibai*, Evan Fabry, David Forsyth
University of Illinois at Urbana Champaign
{jlu23, sibai2, efabry2, daf}@illinois.edu

In this paper, we show experiments that suggest that a trained neural network classifies most of the pictures taken from different distances and angles of a perturbed image correctly. We believe this is because the adversarial property of the perturbation is **sensitive to the scale** at which the perturbed picture is viewed, so (for example) **an autonomous car** will **misclassify** a **stop** sign **only** from a **small range of distances**.

Yes, We Should...

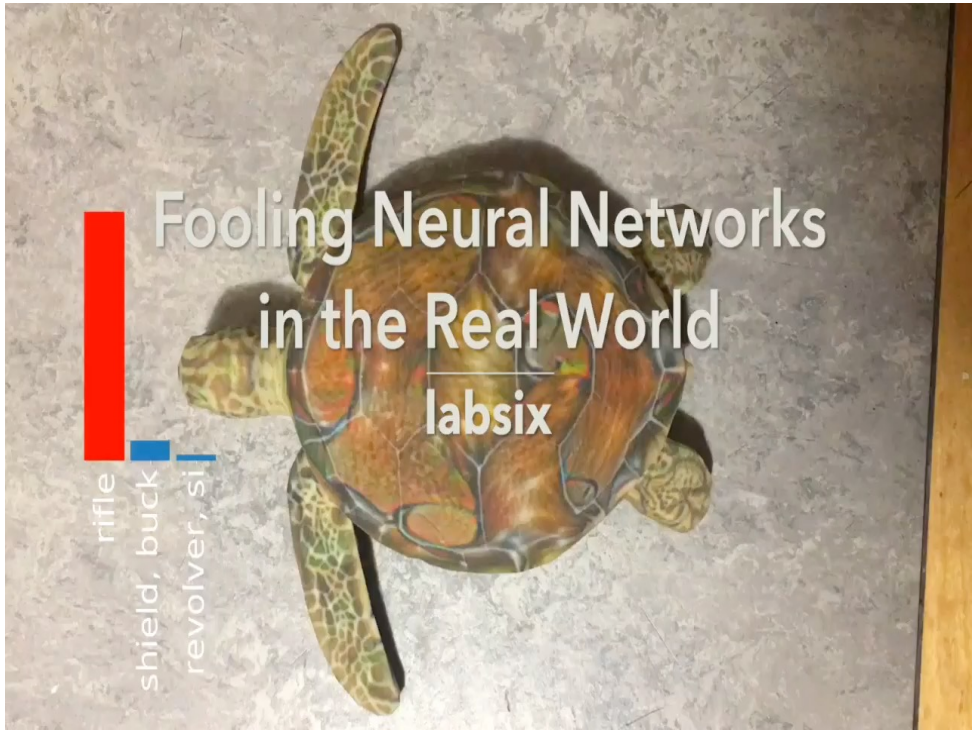
Synthesizing Robust Adversarial Examples

Anish Athalye
OpenAI, MIT

Ilya Sutskever
OpenAI

[https://blog.openai.com/robust-adversarial-inputs/]

Yes, We Should...



[Athalye et al., Synthesizing robust adversarial examples. ICLR, 2018]

Yes, We Should…

Robust Physical-World Attacks on Machine Learning Models

Visit https://iotsecurity.eecs.umich.edu/#roadsigns for an FAQ

Ivan Evtimov[1], Kevin Eykholt[2], Earlence Fernandes[1], Tadayoshi Kohno[1],
Bo Li[4], Atul Prakash[2], Amir Rahmati[3], and Dawn Song*[4]

[1]University of Washington
[2]University of Michigan Ann Arbor
[3]Stony Brook University
[4]University of California, Berkeley

Is This a Real Security Threat?

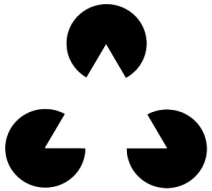
- **Adversarial examples** can exist in the *physical world*, we can fabricate concrete adversarial objects (glasses, road signs, etc.)

- But the effectiveness of attacks carried out by adversarial objects is still to be investigated with **large-scale experiments** in *realistic* **security scenarios**

- Gilmer et al. (2018) have recently discussed the realism of security threat caused by adversarial examples, pointing out that it should be carefully investigated
  - *Are indistinguishable adversarial examples a real security threat?*
  - For which real security scenarios adversarial examples are the best attack vector? Better than attacking components outside the machine learning component
  - …

[Justin Gilmer et al., Motivating the Rules of the Game for Adversarial Example Research, https://arxiv.org/abs/1807.06732]

Are Indistinguishable Perturbations a Real Security Threat?



Indistinguishable Adversarial Examples

- Minimize $\|r\|_2$ subject to:
  1. $f(x+r) = l \quad f(x) \neq l$
  2. $x + r \in [0,1]^m$

The adversarial image $x + r$ is visually hard to distinguish from x

*…There is a **torrent of work** that views increased robustness to **restricted perturbations** as making these models **more secure**. While not all of this work requires completely indistinguishable modications, many of the papers focus on specifically small modications, and the language in many suggests or implies that the **degree of perceptibility** of the **perturbations** is an important aspect of their **security risk**…*

[Justin Gilmer et al., Motivating the Rules of the Game for Adversarial Example Research, arXiv 2018]

Indistinguishable Adversarial Examples

- The attacker can benefit by minimal perturbation of a legitimate input; e.g., she could use the attack for a longer period of time before it is detected

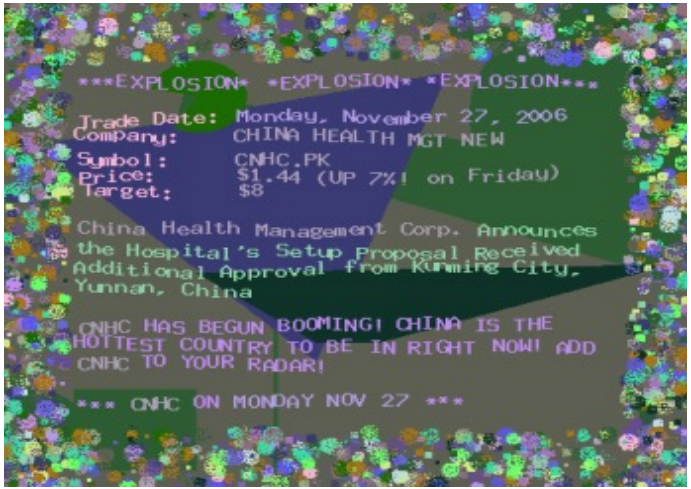
- But is *minimal perturbation* a necessary constraint for the attacker?

Indistinguishable Adversarial Examples

- Is *minimal perturbation* a necessary constraint for the attacker?

Attacks with Content Preservation

There are well known security applications where minimal perturbations and indistinguishability of adversarial inputs are not required at all…



Are Indistinguishable Perturbations a Real Security Threat?

*…At the time of writing, we were **unable** to find a **compelling example** that **required indistinguishability**…*

*To have the largest impact, we should both recast future adversarial example research as a **contribution** to **core machine learning** and develop new abstractions that capture **realistic threat models**.*

[Justin Gilmer et al., Motivating the Rules of the Game for Adversarial Example Research, arXiv 2018]

Battista Biggio
battista.biggio@unica.it
@biggiobattista

Thanks!

If you know the enemy and know yourself, you need not fear the result of a hundred battles
Sun Tzu, The art of war, 500 BC