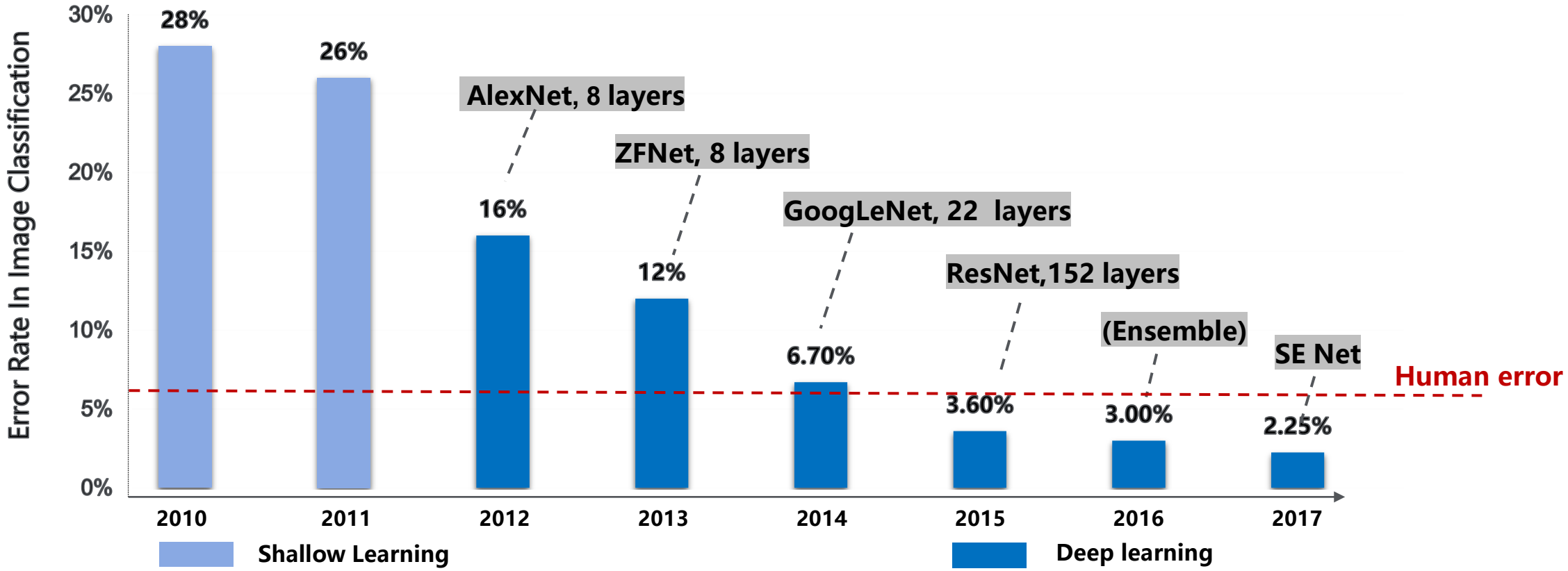
对抗机器学习：攻击、防御与模型鲁棒性评估

易津锋 博士

京东智联云机器学习部负责人
京东（上海）人工智能研究院负责人

JD AI Research

ImageNet Challenge



Accuracy is NOT the Sole Metric

- Suppose you have a well-trained ImageNet classifier that achieves >97% accuracy.
- Should you always trust its prediction?

JDAI Research

Accuracy is NOT the Sole Metric

- Suppose you have a **well-trained ImageNet classifier** that achieves **>97% accuracy**.
- Should you **always trust its prediction?**




Any differences?

Accuracy is NOT the Sole Metric

- Suppose you have a well-trained ImageNet classifier that achieves >97% accuracy.
- Should you always trust its prediction?

Toy poodle

Blow dryer

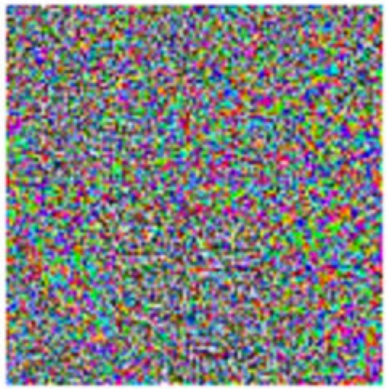
Rubber

Crawfish

Adversarial Examples

- A **small perturbation** could **dramatically change the prediction**

$+ .007 \times$

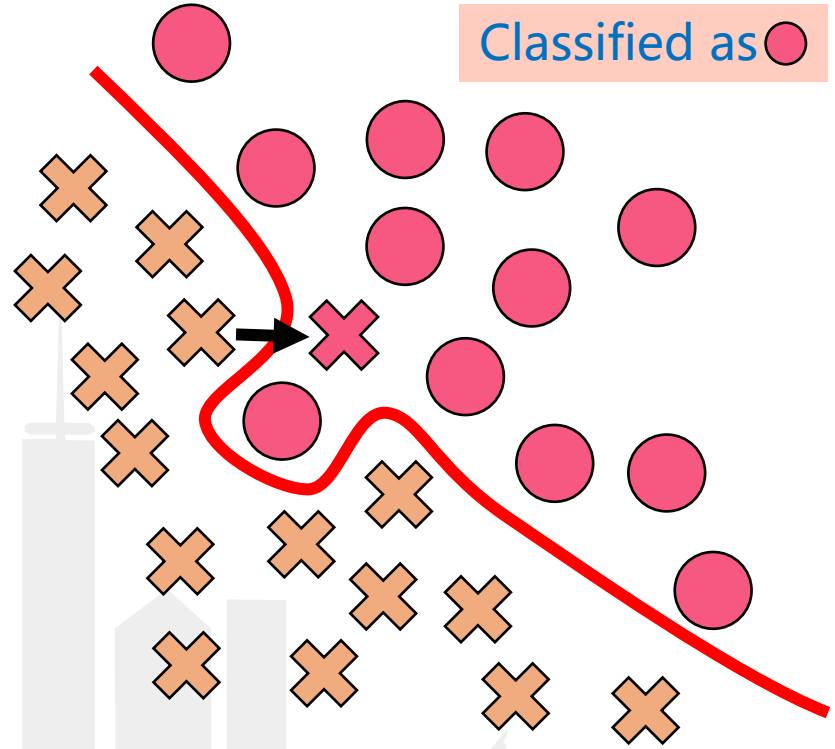


$=$

"panda"
57.7% confidence

"nematode"
8.2% confidence

"gibbon"
99.3 % confidence

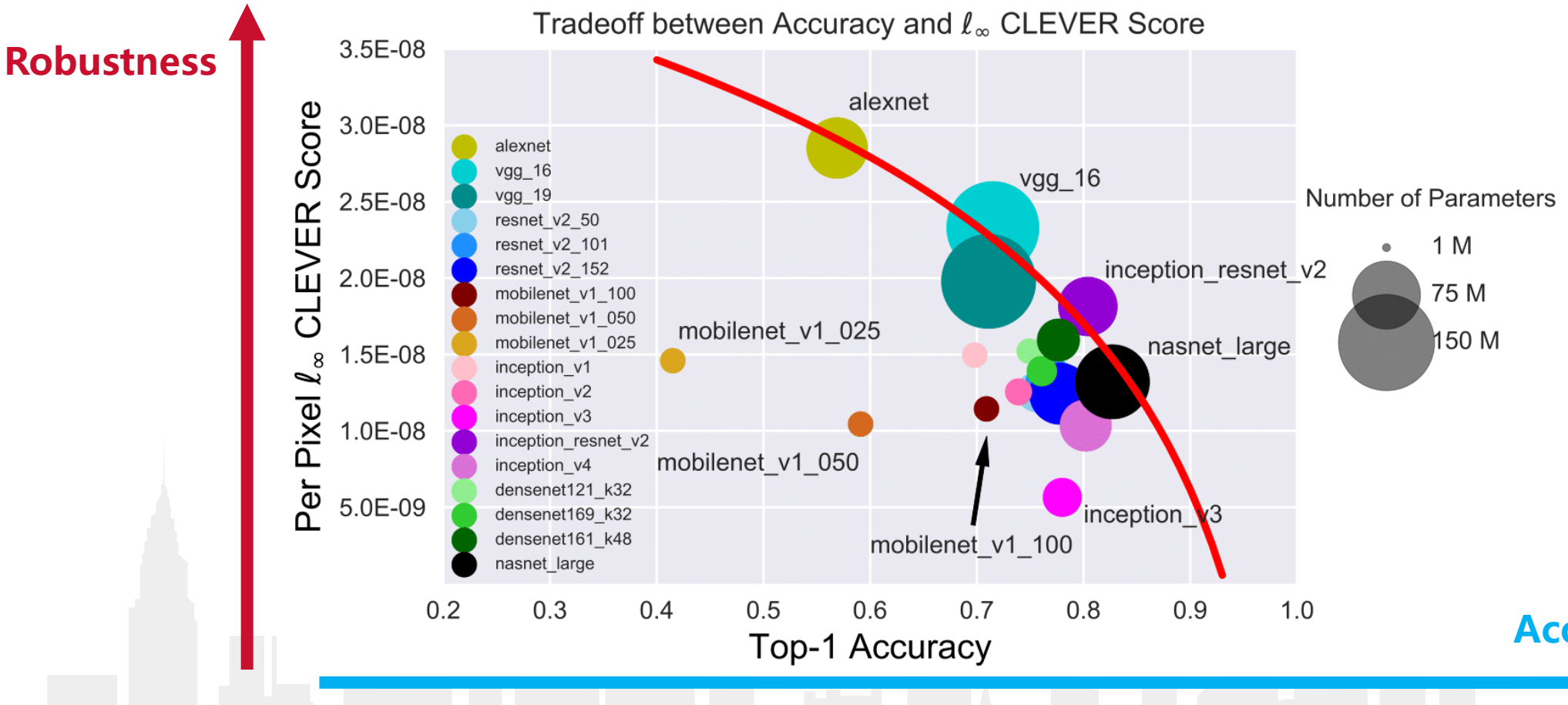


Goodfellow et al., Explaining and Harnessing Adversarial Examples, ICLR 2015

JDAI Research

Trade-off between Accuracy & Robustness

- **Solely pursuing for high-accuracy rate may be misleading**



Su et al., Is Robustness the Cost of Accuracy? A Comprehensive Study on the Robustness of 18 Deep Image Classification Models, ECCV 2018

Dilemma in Security-critical Tasks

- Security-critical tasks require **both accuracy and robustness**

Autonomous Driving

Healthcare

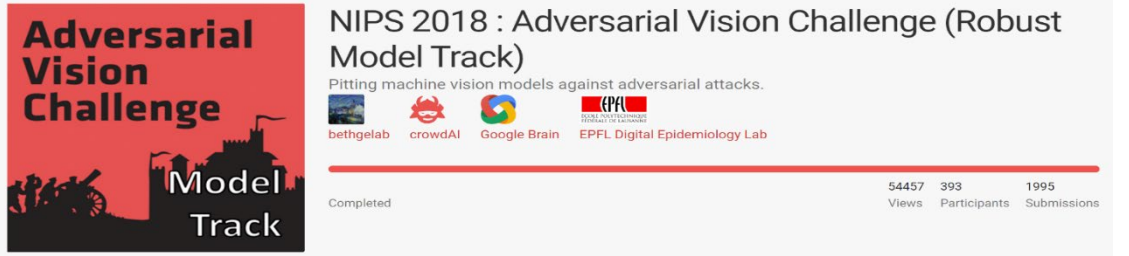
Finance

Surveillance

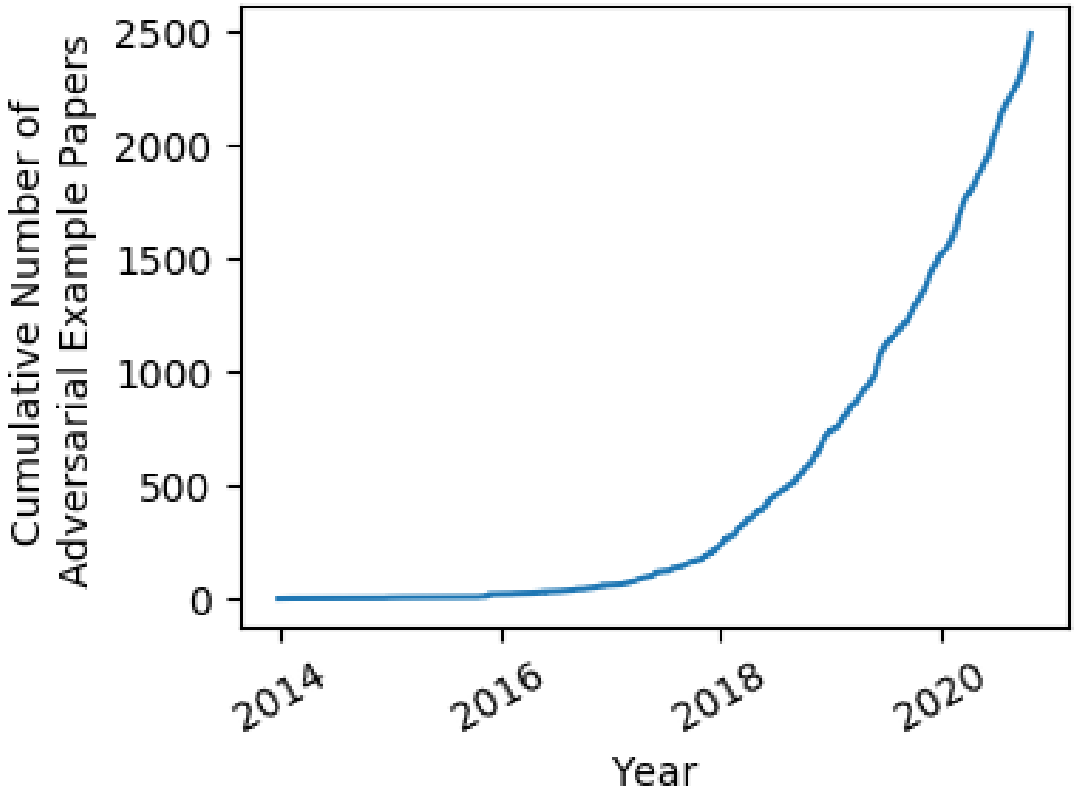
Military

Law

Adversarial Machine Learning: a Fast Growing Area



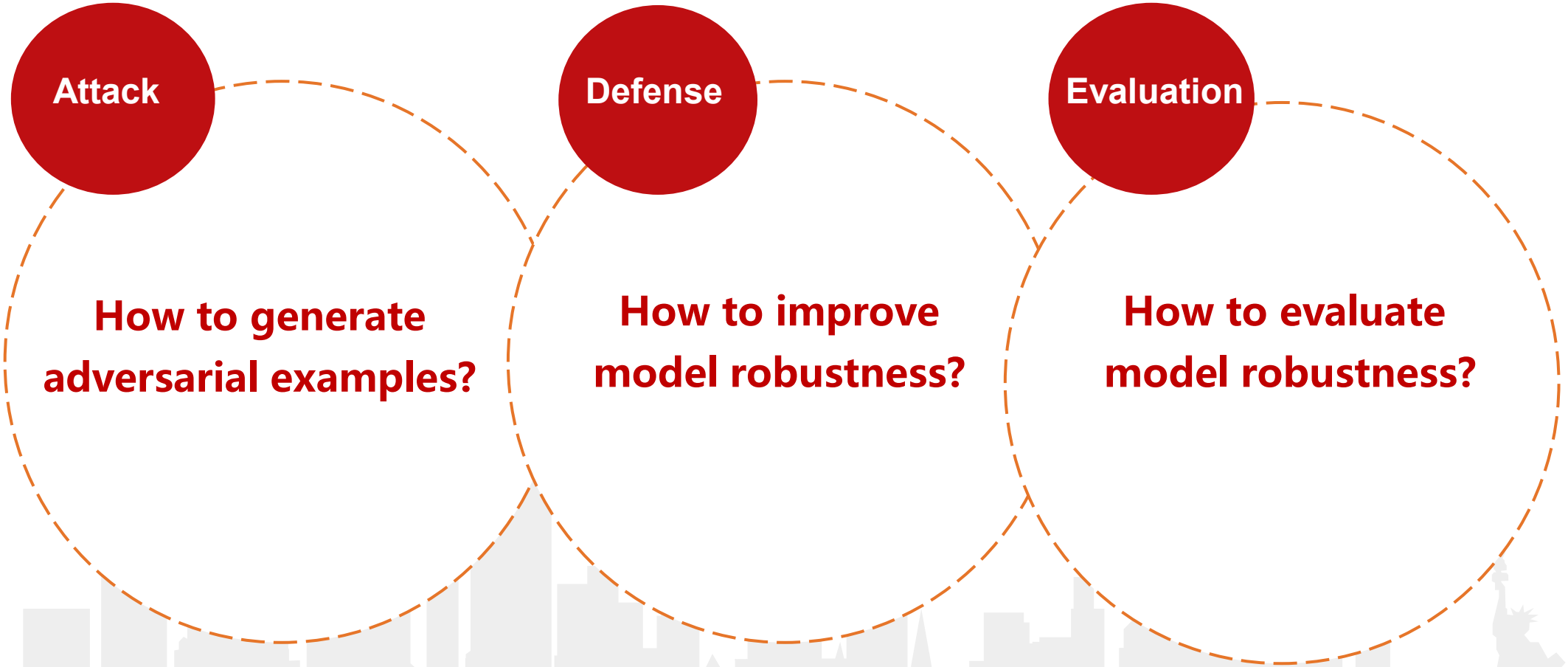
Number of papers about adversarial examples **increases almost exponentially**.



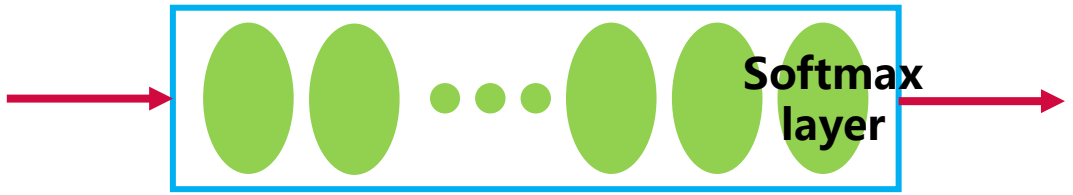
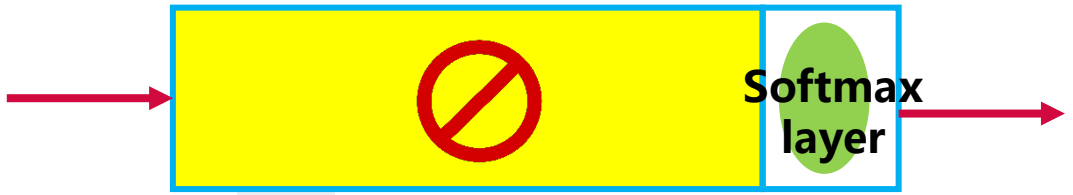
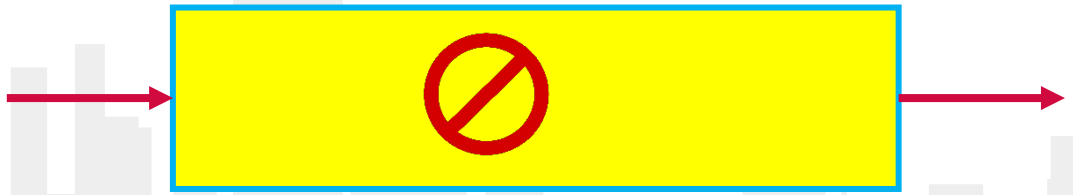
https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html

JDAI Research

Key Problems in Adversarial Machine Learning



Attack Settings: White and Black-Box Attacks

Attack Settings	Deep Neural Networks (DNNs)	Output	Back Propagation
White-box		Confidence scores	
Black-box with soft labels		Confidence scores	
Black-box with hard labels		Hard labels	

Increasingly harder

White-box Attack by Optimization-based Approaches

General attack formulation

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \ \mathrm{Dist}(\mathbf{x}, \mathbf{x}_0) + c \cdot \mathcal{L}(\mathbf{x}, t)$$

- $\mathrm{Dist}(\mathbf{x}, \mathbf{x}_0)$: the distortion between the input image and the adversarial example
  - ℓ_∞ distance: $\|\delta\|_\infty = \max_i |\delta_i|$ maximal pixel-wise distortion
  - ℓ_2 distance: $\|\delta\|_2 = \sqrt{\sum_i \delta_i^2}$ Euclidean distance
  - ℓ_1 distance: $\|\delta\|_1 = \sum_i |\delta_i|$ total variation
  - Elastic Net regularization: $\|\mathbf{x} - \mathbf{x}_0\|_2^2 + \beta\|\mathbf{x} - \mathbf{x}_0\|_1$ [Chen et al., AAAI 2018]
    - Craft adversarial examples with better attack transferability
    - Bypass many defending techniques
- $\mathcal{L}(\mathbf{x}, t)$: loss function, usually use cross entropy or contrastive loss

N. Carlini, D. Wagner. Towards Evaluating the Robustness of Neural Networks. IEEE Symposium on Security and Privacy, 2017
Chen et al. EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples, AAAI 2018

Black-box Attacks with Soft Labels

- The adversary has no access to the structure and parameters of deep neural networks
- He can only query the model and get the probability outputs

Black-box Attacks by Zeroth Order Optimization (ZOO)

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \ \mathrm{Dist}(\mathbf{x}, \mathbf{x}_0) + c \cdot \mathcal{L}(\mathbf{x})$$

- Cannot compute the gradient in the black-box setting

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

Black-box Attacks by Zeroth Order Optimization (ZOO)

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \ \mathrm{Dist}(\mathbf{x}, \mathbf{x}_0) + c \cdot \mathcal{L}(\mathbf{x})$$

- Cannot compute the gradient in the black-box setting
- Symmetric difference quotient to estimate the gradient:

$$\hat{g}_i \approx \frac{\partial \mathcal{L}(\boldsymbol{x})}{\partial x_i} \approx \frac{\mathcal{L}(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - \mathcal{L}(\boldsymbol{x} - \epsilon \boldsymbol{e}_i)}{2\epsilon}$$

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta \begin{bmatrix} \hat{g}_1 \\ \vdots \\ \hat{g}_d \end{bmatrix}$$

- However, need O(d) queries to estimate a gradient
  - ImageNet: d = 299*299*3 > 268K
  - 100 iterations => 26.8 million queries

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

Black-box Attacks by Zeroth Order Optimization (ZOO)

- Reduce number of queries:
  - Stochastic Coordinate descent: update a small set of coordinates at each time
  - Greedy approach: select important coordinates first
  - Attack-space dimension reduction + hierarchical attack
- **The first black-box attack algorithm that achieves close to 100% attack success rate**.

	MNIST		
	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box	100 %	2.00661	0.53 min
Black-box (Substitute Model)	26.74 %	5.272	0.80 min (+ 6.16 min)
Proposed black-box (ZOO)	**98.9** %	1.987068	1.62 min
	CIFAR10		
	Success Rate	Avg. L_2	Avg. Time (per attack)
White-box	100 %	0.37974	0.16 min
Black-box (Substitute Model)	5.3 %	5.7439	0.49 min (+ 7.81 min)
Proposed Black-box (ZOO)	**96.8** %	0.39879	3.95 min

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

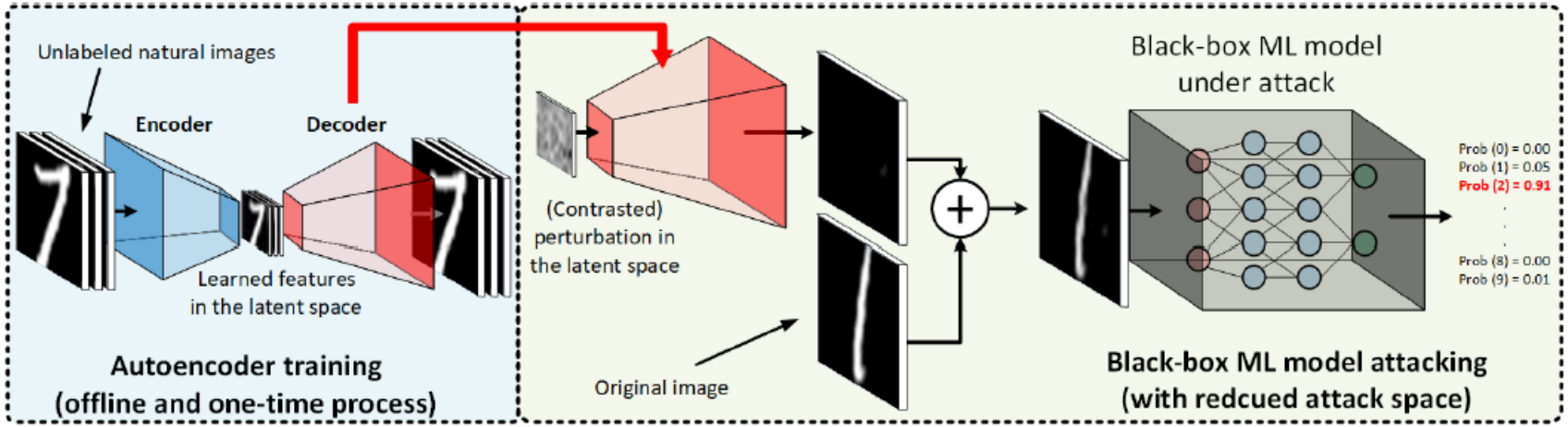
Further Improves Query Efficiency

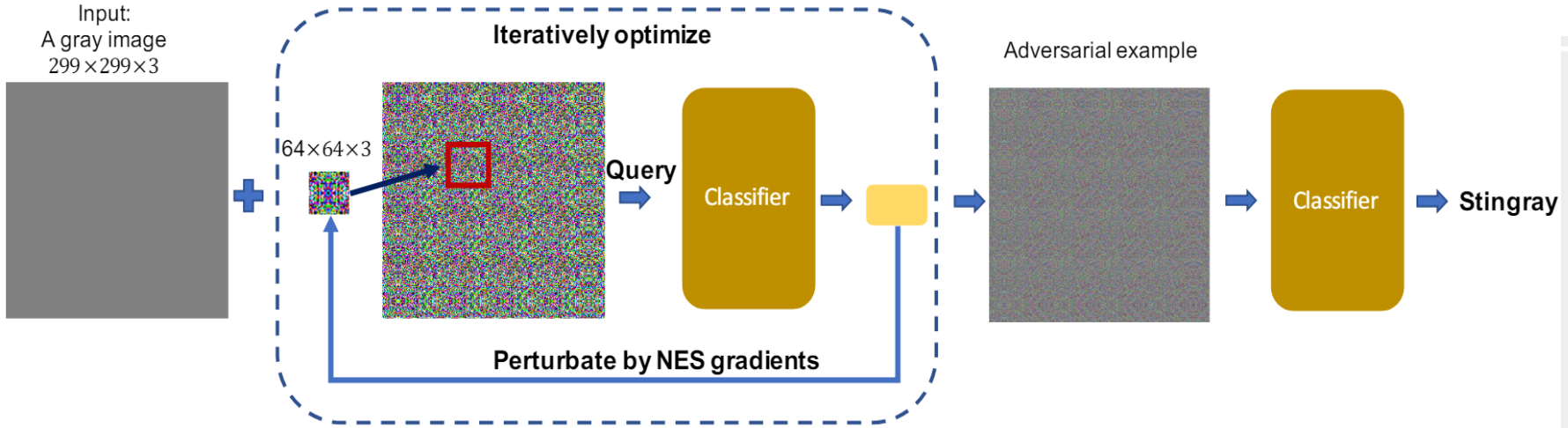
AutoZOOM:

1. Dispense with coordinate-wise estimation and instead propose a scaled random full gradient estimator.

2. An autoencoder trained offline with unlabeled data or a bilinear resizing operation for attack acceleration.

Reduced at least 93.2% query count
99.4% for ImageNet





Input-free Attack:

1. Start with a gray color image;
2. Shrink the dimension, then perturb a small region and tile it to cover the input image.

With only 1.7K queries on average, can perturb a gray image to any target class of ImageNet with a 100% success rate on InceptionV3.

Tu et al., AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method for Attacking Black-Box Neural Networks. AAAI 2019
Du et al., Towards Query Efficient Black-box Attacks: An Input-free Perspective. AISec@CCS 2018

JDAI Research

Further Improves Query Efficiency

Using an **active learning strategy** to significantly reduce the number of queries

Algorithm 2 Substitute DNN training with active learning

INPUT: target oracle $\tilde{O}$, a maximum number ρ_{max} of training epochs, and an initial training set S_0.
OUTPUT: a trained substitute model F.

Define architecture F;
for $\rho = 0; \rho < \rho_{max}; \rho++$ **do**
  if $\rho = 0$ **then**
    $D \leftarrow \{(\mathbf{x}, \tilde{O}(\mathbf{x}))|\mathbf{x} \in S_\rho\}$;
  else
    $D_{add} \leftarrow \{(\mathbf{x}, \tilde{O}(\mathbf{x}))|\mathbf{x} \in S_{add}\}$;
    $D \leftarrow [D, D_{add}]$;
  end if
  train F with D;
  craft S_{add};
  Use Active Learning strategy to generate a new S_{add};
  $S_{\rho+1} \leftarrow S_\rho \cup S_{add}$;
end for

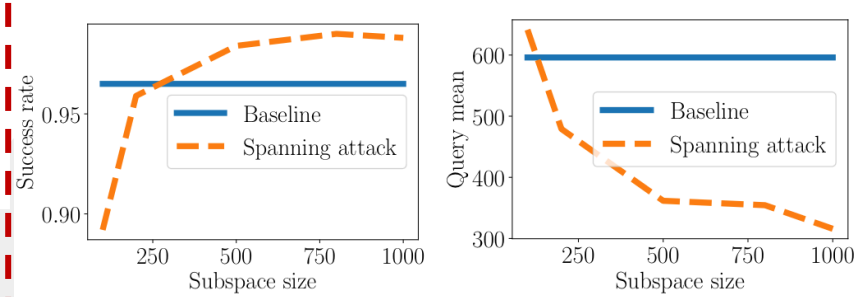
Reduce more than 90% of queries
Obtain an accurate substitute model 85% similar with the target oracle.

Spanning Attack
Constrain a subspace spanned by an auxiliary small unlabeled dataset

$$\boldsymbol{a} = \boldsymbol{b}^\top \boldsymbol{S} = (\boldsymbol{b}^\top \boldsymbol{U}_N \boldsymbol{\Sigma}_N)\boldsymbol{V}_N^\top.$$

$$\boldsymbol{\Sigma}_N^{-1}\boldsymbol{U}_N^\top \boldsymbol{S} = \boldsymbol{V}_N^\top.$$

$$\boldsymbol{a} = \boldsymbol{b}^\top \boldsymbol{V}_N^\top = (\boldsymbol{b}^\top \boldsymbol{\Sigma}_N^{-1}\boldsymbol{U}_N^\top)\boldsymbol{S}.$$



(a) Success rate (b) Query mean

The reinforced attack typically requires less than 50% queries while improves success rates in the meantime.

Propose a novel **Frank-Wolfe based projection-free attack framework** for both white-box and black-box settings

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}_t) + \langle \mathbf{x} - \mathbf{x}_t, \nabla f(\mathbf{x}_t)\rangle$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \epsilon \cdot \mathrm{sign}(\mathbf{m}_t) - \gamma_t(\mathbf{x}_t - \mathbf{x}_{\mathrm{ori}}).$$

Algorithm 1 Frank-Wolfe White-box Attack Algorithm

input: number of iterations T, step sizes $\{\gamma_t\}$;
$\mathbf{x}_0 = \mathbf{x}_{\mathrm{ori}}, \mathbf{m}_{-1} = \nabla f(\mathbf{x}_0)$
for $t = 0, \ldots, T-1$ **do**
  $\mathbf{m}_t = \beta \cdot \mathbf{m}_{t-1} + (1-\beta) \cdot \nabla f(\mathbf{x}_t)$
  $\mathbf{v}_t = \mathrm{argmin}_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \mathbf{m}_t \rangle$ // LMO
  $\mathbf{d}_t = \mathbf{v}_t - \mathbf{x}_t$
  $\mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_t \mathbf{d}_t$
end for
output: $\mathbf{x}_T$

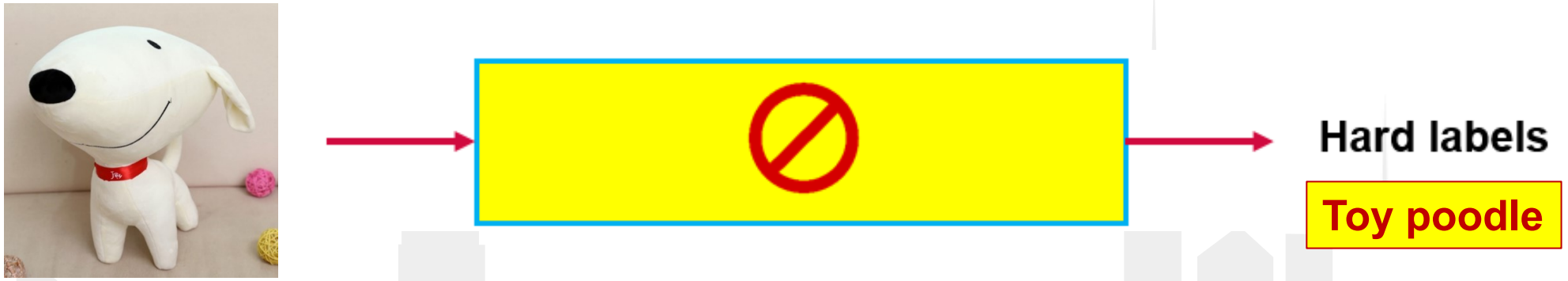
The proposed attack algorithms with momentum mechanism enjoy an $O(1/\sqrt{T})$ convergence rate in the nonconvex setting.

Li, et al., Query-Efficient Black-Box Attack by Active Learning, ICDM 2018
Wang et al., Spanning Attack: Reinforce Black-box Attacks with Unlabeled Data, Machine Learning 2020
Chen et al., A Frank-Wolfe Framework for Efficient and Effective Adversarial Attacks, AAAI 2020

JDAI Research

Black-box Attacks with Hard Labels

- The adversary has no access to the structure and parameters of deep neural networks
- He can only query the model and get the hard-label multi-class output



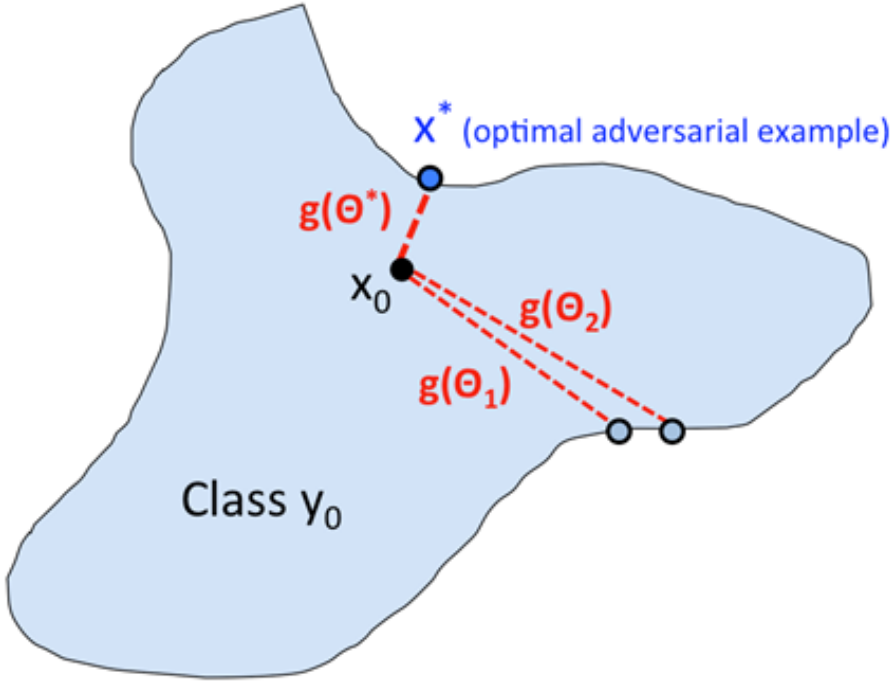
Optimization-based Hard-label Black-box Attack

- Reformulate the attack optimization problem

$$\theta^* = \arg\min_{\boldsymbol{\theta}} \; g(\boldsymbol{\theta})$$

$$\text{Untargeted attack: } g(\boldsymbol{\theta}) = \texttt{argmin}_{\lambda>0}\left(f(\mathbf{x}_0 + \lambda\frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|}) \neq y_0\right)$$

$$\text{Targeted attack: } g(\boldsymbol{\theta}) = \texttt{argmin}_{\lambda>0}\left(f(\mathbf{x}_0 + \lambda\frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|}) = t\right)$$



- Cannot compute the gradient of g
- However, can compute the function value of g via querying
- Binary search + fine-grained search

θ: the direction of adversarial example

Cheng et al., Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach. ICLR 2019

Optimization-based Hard-label Black-box Attack

Algorithm for computing $g(\theta)$

Algorithm 1 Compute $g(\boldsymbol{\theta})$ locally

Input: Hard-label model f, original image x_0, query direction $\boldsymbol{\theta}$, previous value v, increase/decrease ratio $\alpha = 0.01$, stopping tolerance ϵ (maximum tolerance of computed error)
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}/\|\boldsymbol{\theta}\|$
if $f(\boldsymbol{x}_0 + v\boldsymbol{\theta}) = y_0$ **then**
    $v_{left} \leftarrow v, v_{right} \leftarrow (1+\alpha)v$
    while $f(\boldsymbol{x}_0 + v_{right}\boldsymbol{\theta}) = y_0$ **do**
        $v_{right} \leftarrow (1+\alpha)v_{right}$
else
    $v_{right} \leftarrow v, v_{left} \leftarrow (1-\alpha)v$
    while $f(\boldsymbol{x}_0 + v_{left}\boldsymbol{\theta}) \neq y_0$ **do**
        $v_{left} \leftarrow (1-\alpha)v_{left}$
Binary Search within $[v_{left}, v_{right}]$
while $v_{right} - v_{left} > \epsilon$ **do**
    $v_{mid} \leftarrow (v_{right} + v_{left})/2$
    if $f(\boldsymbol{x}_0 + v_{mid}\boldsymbol{\theta}) = y_0$ **then**
        $v_{left} \leftarrow v_{mid}$
    else
        $v_{right} \leftarrow v_{mid}$
return v_{right}

Zeroth-order optimization for minimizing $g(\theta)$

Algorithm 2 RGF for hard-label black-box attack

Input: Hard-label model f, original image x_0, initial $\boldsymbol{\theta}_0$.
for $t = 0, 1, 2, \ldots, T$ **do**
    Randomly choose $\boldsymbol{u}_t$ from a zero-mean Gaussian distribution
    Evaluate $g(\boldsymbol{\theta}_t)$ and $g(\boldsymbol{\theta}_t + \beta\boldsymbol{u})$ using Algorithm 1
    Compute $\hat{\boldsymbol{g}} = \dfrac{g(\boldsymbol{\theta}_t + \beta\boldsymbol{u}) - g(\boldsymbol{\theta}_t)}{\beta} \cdot \boldsymbol{u}$
    Update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t\hat{\boldsymbol{g}}$
return $\boldsymbol{x}_0 + g(\boldsymbol{\theta}_T)\boldsymbol{\theta}_T$

More than 4 times faster than Decision-attack

	MNIST		CIFAR10	
	Avg L_2	# queries	Avg L_2	# queries
Decision-attack (black-box)	2.3158	30,103	0.2850	55,552
	2.0052	58,508	0.2213	140,572
	1.8668	192,018	0.2122	316,791
Opt-attack (black-box)	1.8522	46,248	0.2758	61,869
	1.7744	57,741	0.2369	141,437
	1.7114	73,293	0.2300	186,753
C&W (white-box)	1.4178	-	0.1901	-

Cheng etal., Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach. ICLR 2019

Adversarial Examples in Image Captioning

CNN → RNN → A clock on a pole in front of a building.

A typical neural image captioning system with a CNN+RNN architecture.

Experiments	Success Rate	Avg. $\|\delta\|_2$
targeted caption	95.8%	2.213
1-keyword	97.1%	1.589
2-keyword	97.5%	2.363
3-keyword	96.0%	2.626
C&W on CNN	22.4%	2.870
I-FGSM on CNN	34.5%	15.596

Chen et al., Attacking Visual Language Grounding with Adversarial Examples: A Case Study on Neural Image Captioning. ACL 2018

Targeted keywords attack

Original Top-3 inferred captions:
1. A bus is parked on the side of the street.
2. A bus is parked on the side of the road.
3. A bus is parked on the side of a street.

Adversarial Keywords:
"tub", "bathroom" and "sink"

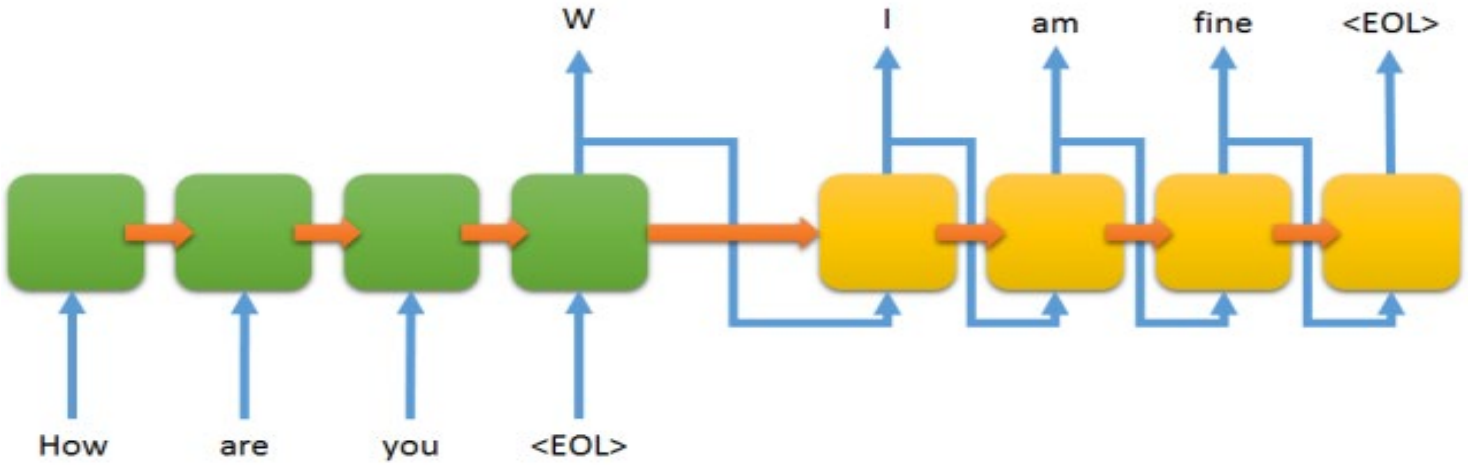
Adversarial Top-3 captions:
(targeted keyword method)
1. A bathroom with a sink, toilet and tub.
2. A bathroom with a sink, toilet, and bathtub.
3. A bathroom with a tub, sink, and toilet.

Targeted caption attack

Original Top-3 inferred captions:
1. A red stop sign sitting on the side of a road.
2. A stop sign on the corner of a street.
3. A red stop sign sitting on the side of a street.

Adversarial Top-3 captions:
1. A brown teddy bear laying on top of a bed.
2. A brown teddy bear sitting on top of a bed.
3. A large brown teddy bear laying on top of a bed.

Adversarial Examples in Sequence-to-Sequence Models



Sequence-to-Sequence Models
(https://github.com/farizrahman4u/seq2seq)

DATASET	SUCCESS RATE	BLEU	# CHANGED
GIGAWORD	86.0%	0.828	2.17
DUC2003	85.2%	0.774	2.90
DUC2004	84.2%	0.816	2.50

Attacking Text Summarization

METHOD	SUCCESS%	BLEU	# CHANGED
NON-OVERLAP	89.4%	0.349	3.5
1-KEYWORD	100.0%	0.705	1.8
2-KEYWORD	91.0 %	0.303	4.0
3-KEYWORD	69.6%	0.205	5.3

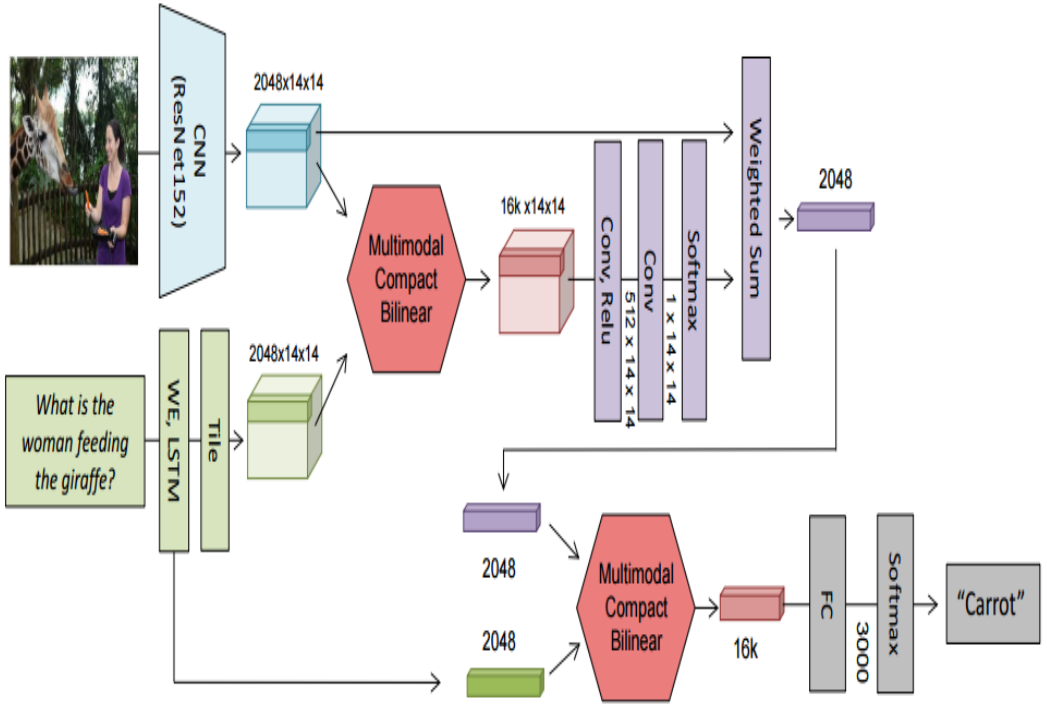
Attacking Machine Translation

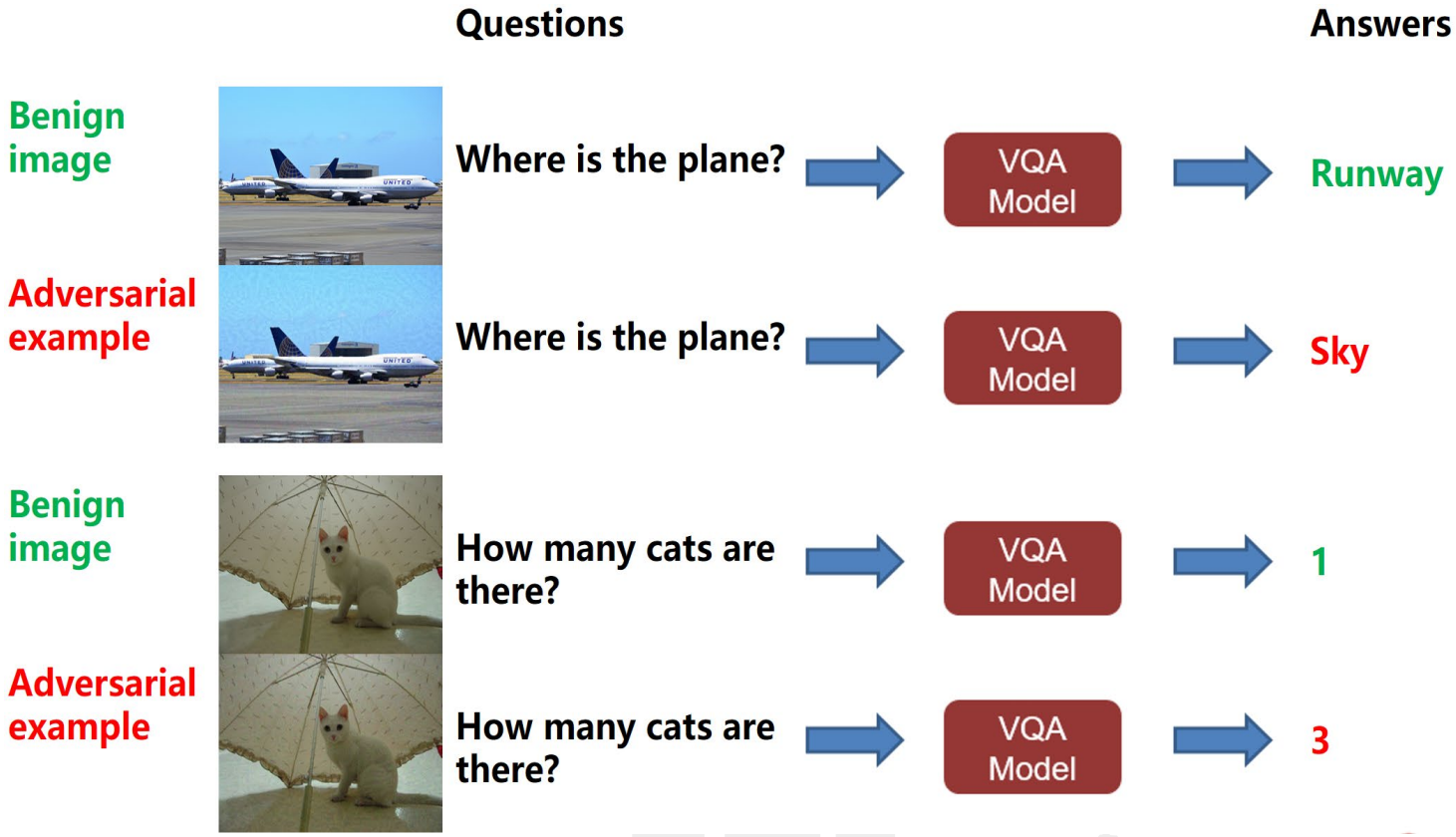
SOURCE INPUT SEQ	UNDER NATO THREAT TO END HIS PUNISHING OFFENSIVE AGAINST ETHNIC ALBANIAN SEPARATISTS IN KOSOVO , PRESIDENT SLOBODAN MILOSEVIC OF YUGOSLAVIA HAS ORDERED MOST UNITS OF HIS ARMY BACK TO THEIR BARRACKS AND MAY WELL AVOID AN ATTACK BY THE ALLIANCE , MILITARY OBSERVERS AND DIPLOMATS SAY
ADV INPUT SEQ	UNDER NATO THREAT TO END HIS PUNISHING OFFENSIVE AGAINST ETHNIC ALBANIAN SEPARATISTS IN KOSOVO , PRESIDENT SLOBODAN MILOSEVIC OF YUGOSLAVIA HAS **jean-sebastien** MOST UNITS OF HIS ARMY BACK TO THEIR BARRACKS AND MAY WELL AVOID AN ATTACK BY THE ALLIANCE , MILITARY OBSERVERS AND DIPLOMATS SAY.
SOURCE OUTPUT SEQ	MILOSEVIC ORDERS ARMY BACK TO BARRACKS
ADV OUTPUT SEQ	**nato may not attack kosovo**

Cheng et al., Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples. AAAI 2020

JDAI Research

Adversarial Examples in Visual Question Answering



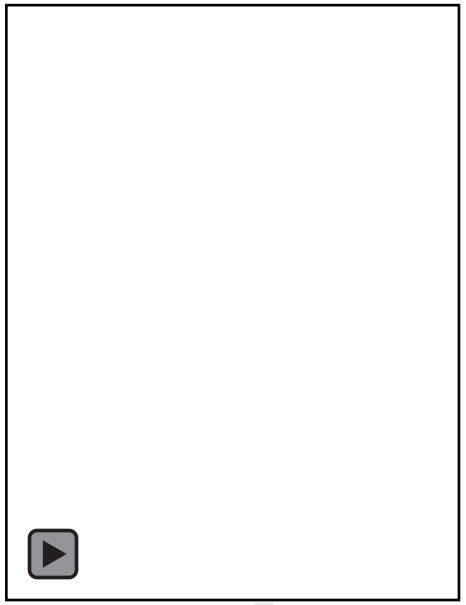


Fukui et al., Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. EMNLP 2016
Zeng et al., Adversarial Attacks Beyond the Image Space. CVPR 2019

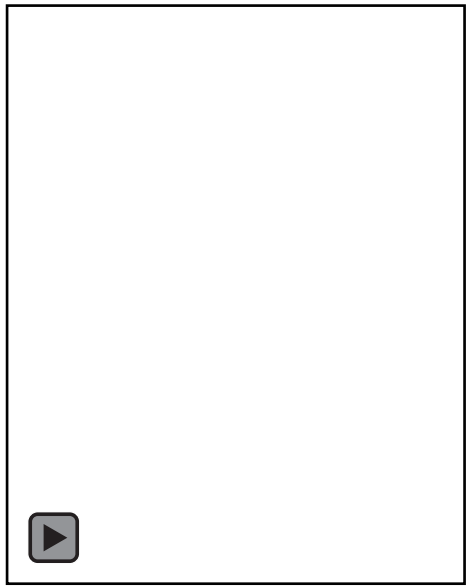
JDAI Research

Adversarial Examples in Reinforcement Learning

Original Frames

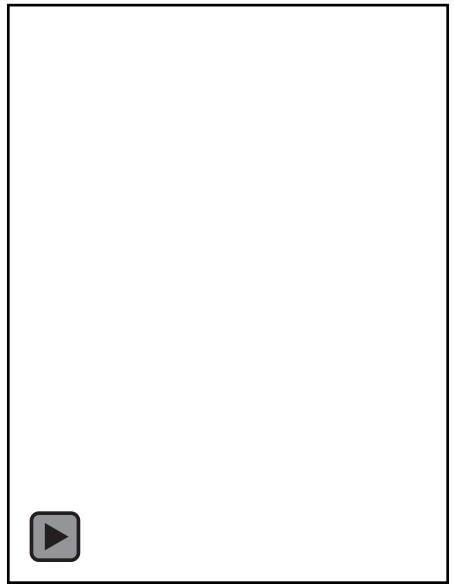


Attack the policy network



Adversarial perturbation injected into **every frame**

Attack the value function

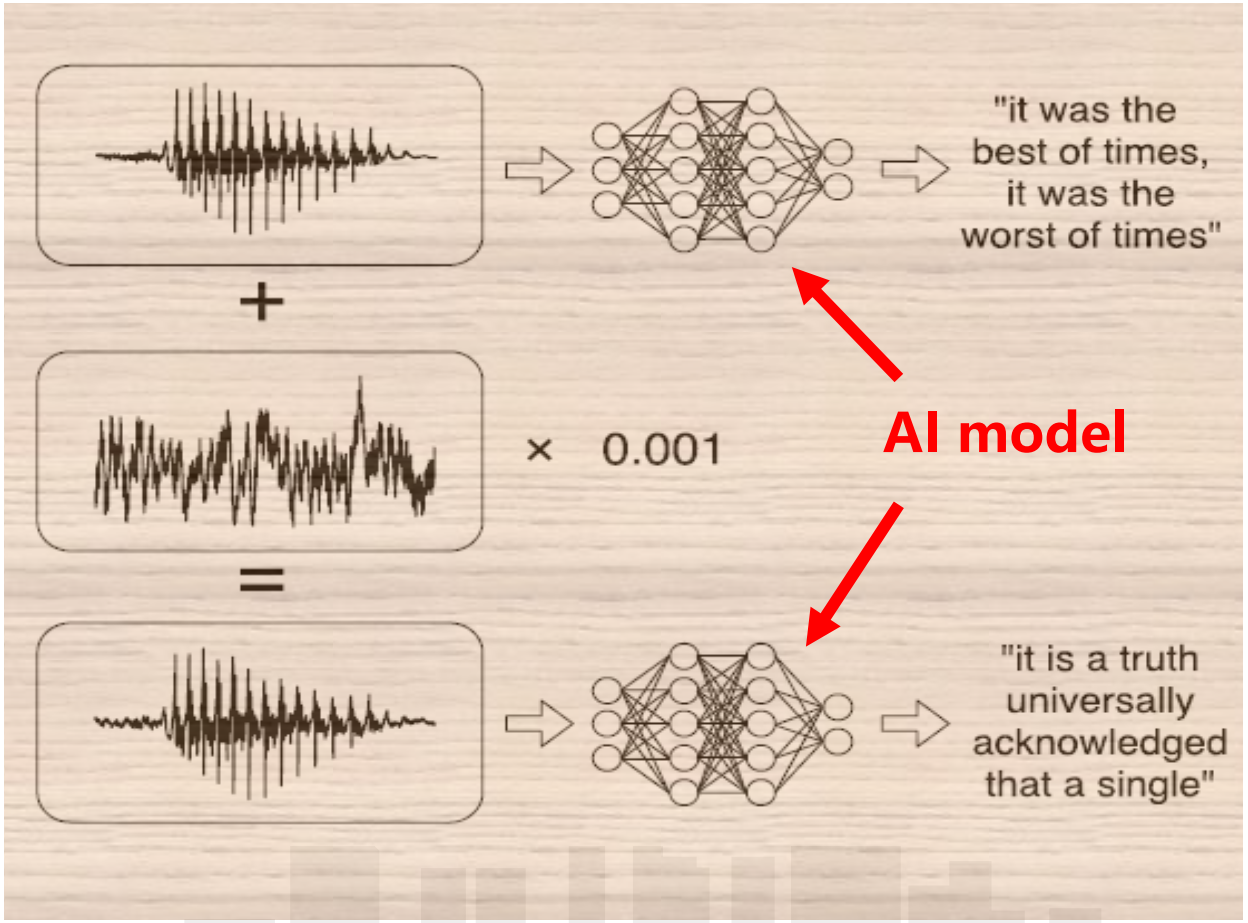


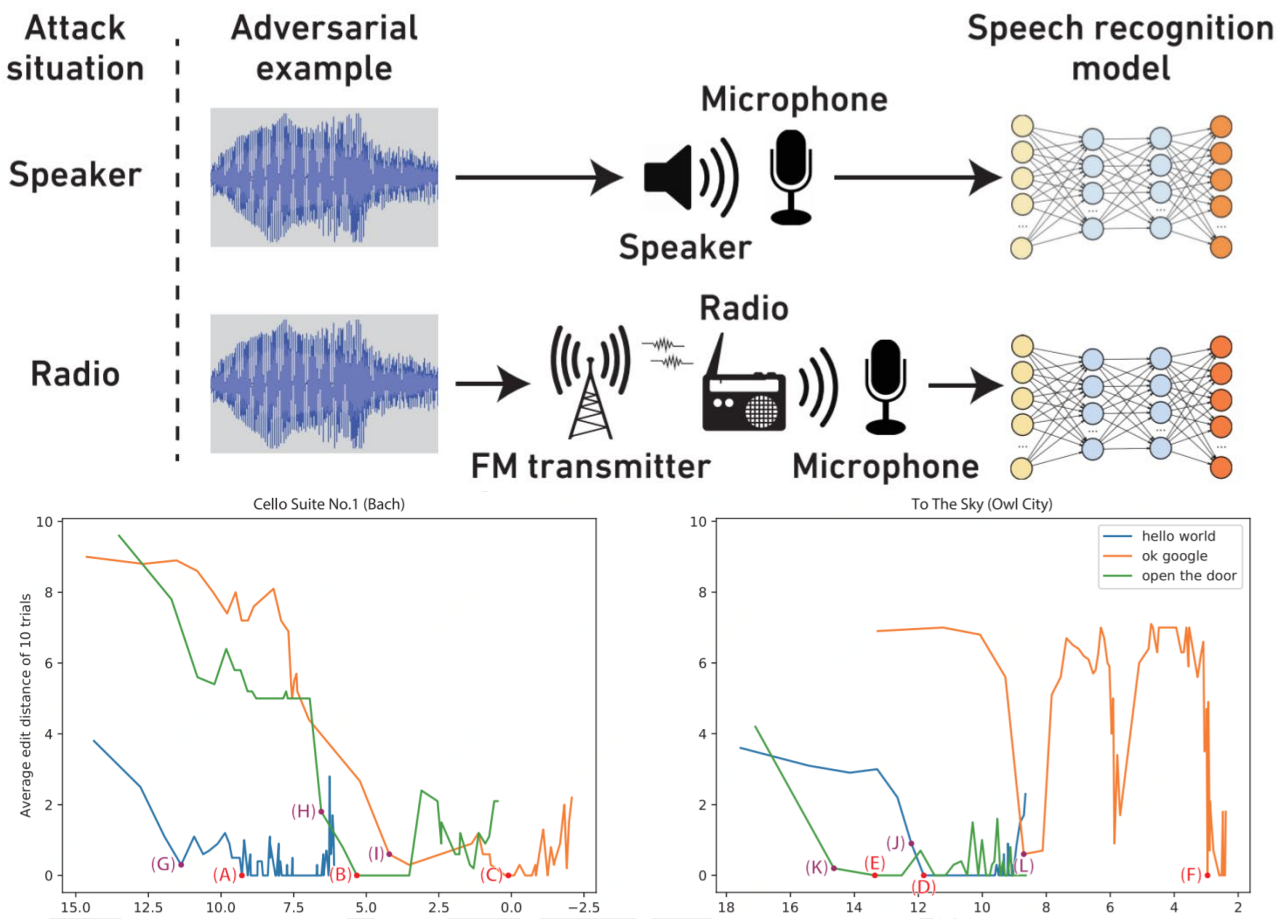
Adversarial perturbation injected into **every other 10 frames**

Kos & Song, Delving into adversarial attacks on deep policies. ICLR 2017

Sun et al., Stealthy and Efficient Adversarial Attacks against Deep Reinforcement Learning. AAAI 2020

Adversarial Examples in Speech Recognition





Carlini et al., Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. IEEE Symposium on Security and Privacy Workshops 2018
Yakura & Sakuma, Robust Audio Adversarial Example for a Physical Attack. IJCAI 2019
Taori et al., Targeted adversarial examples for black box audio systems. IEEE Security and Privacy Workshops 2019

JDAI Research

Adversarial Examples in Shallow Learning

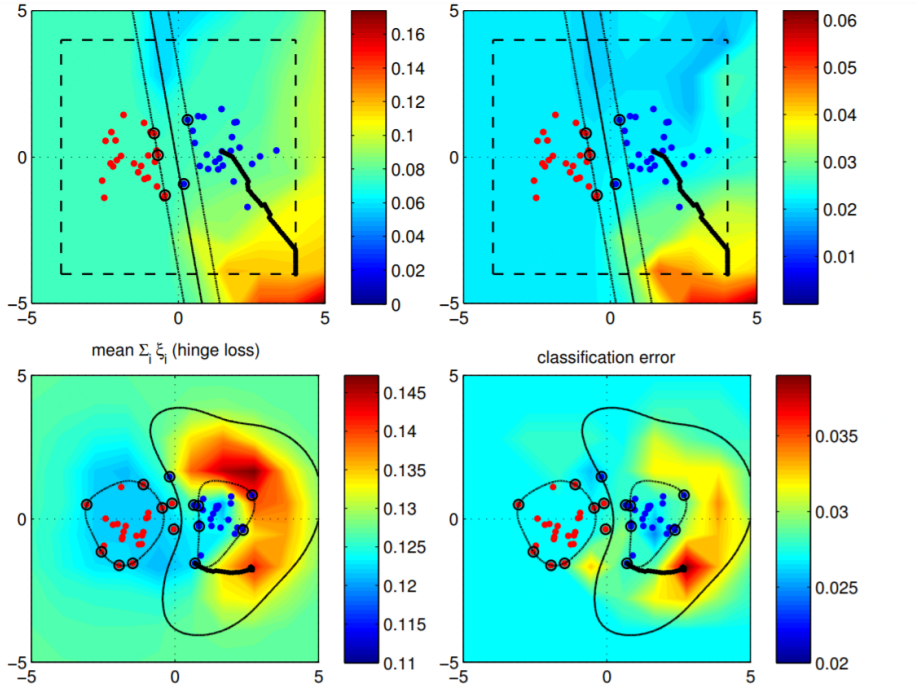
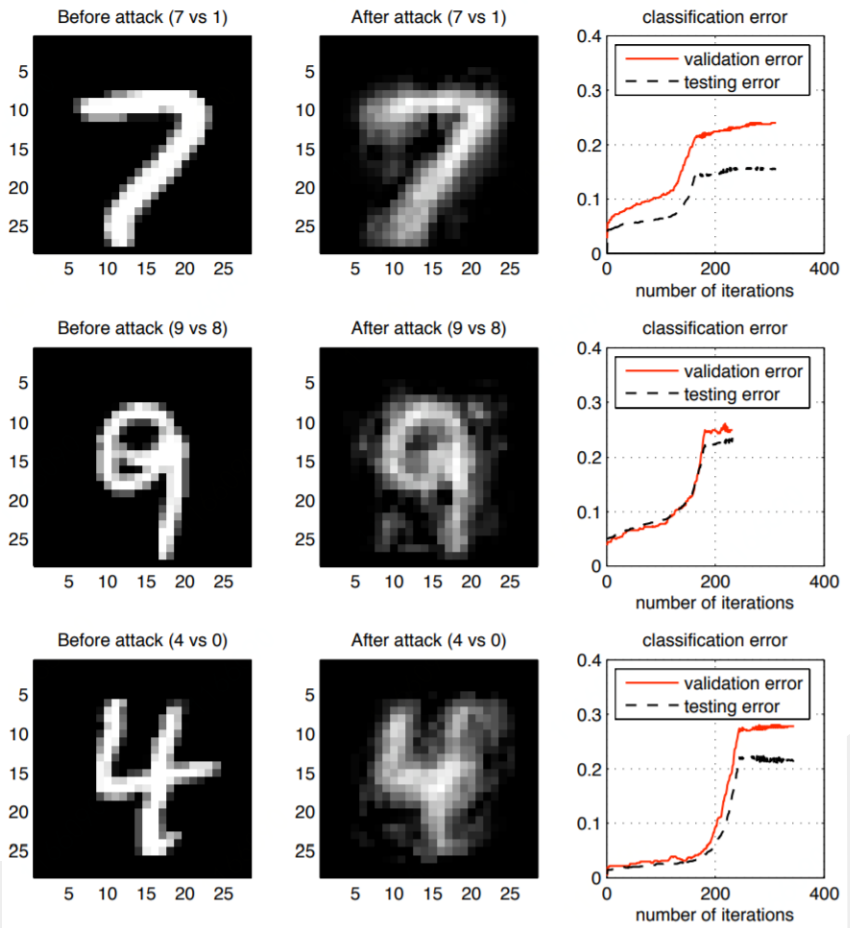
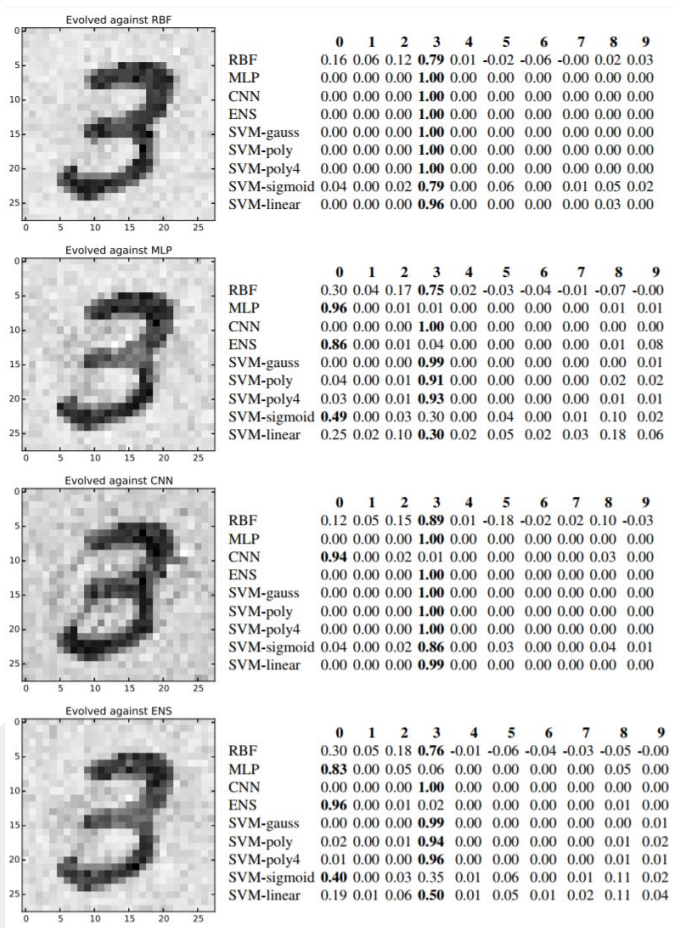


Fig. 7 Behavior of the gradient-based attack strategy on the Gaussian datasets, for the linear (top row) and the RBF kernel (bottom row) with $\gamma = 0.5$. The regularization parameter C was set to 1 in both cases. The solid black line represents the gradual shift of the attack point $\mathbf{x}_c^p$ toward a local maximum. The hinge loss and the classification error are shown in colors, to appreciate that the hinge loss provides a good approximation of the classification error. The value of such functions for each point $\mathbf{x} \in [-5, 5]^2$ is computed by learning an SVM on $\mathcal{D}_{\text{tr}} \cup \{\mathbf{x}, y = -1\}$ and evaluating its performance on $\mathcal{D}_{\text{val}}$. The SVM solution on the clean data $\mathcal{D}_{\text{tr}}$, and the training data itself, are reported for completeness, highlighting the support vectors (with black circles), the decision hyperplane and the margin bounds (with black lines).





	0	1	2	3	4	5	6	7	8	9
RBF	0.16	0.06	0.12	**0.79**	0.01	-0.02	-0.06	-0.00	0.02	0.03
MLP	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
CNN	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
ENS	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
SVM-gauss	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
SVM-poly	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
SVM-poly4	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
SVM-sigmoid	0.04	0.00	0.02	**0.79**	0.00	0.06	0.00	0.01	0.05	0.02
SVM-linear	0.00	0.00	0.00	**0.96**	0.00	0.00	0.00	0.00	0.03	0.00

	0	1	2	3	4	5	6	7	8	9
RBF	0.30	0.04	0.17	**0.75**	0.02	-0.03	-0.04	-0.01	-0.07	-0.00
MLP	**0.96**	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.01
CNN	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
ENS	**0.86**	0.00	0.01	0.04	0.00	0.00	0.00	0.00	0.01	0.08
SVM-gauss	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.01
SVM-poly	0.04	0.00	0.01	**0.91**	0.00	0.00	0.00	0.00	0.02	0.02
SVM-poly4	0.03	0.00	0.01	**0.93**	0.00	0.00	0.00	0.00	0.01	0.01
SVM-sigmoid	**0.49**	0.00	0.03	0.30	0.00	0.04	0.00	0.01	0.10	0.02
SVM-linear	0.25	0.02	0.10	**0.30**	0.02	0.05	0.02	0.03	0.18	0.06

	0	1	2	3	4	5	6	7	8	9
RBF	0.12	0.05	0.15	**0.89**	0.01	-0.18	-0.02	0.02	0.10	-0.03
MLP	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
CNN	**0.94**	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.03	0.00
ENS	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
SVM-gauss	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
SVM-poly	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
SVM-poly4	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
SVM-sigmoid	0.04	0.00	0.02	**0.86**	0.00	0.03	0.00	0.00	0.04	0.01
SVM-linear	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00

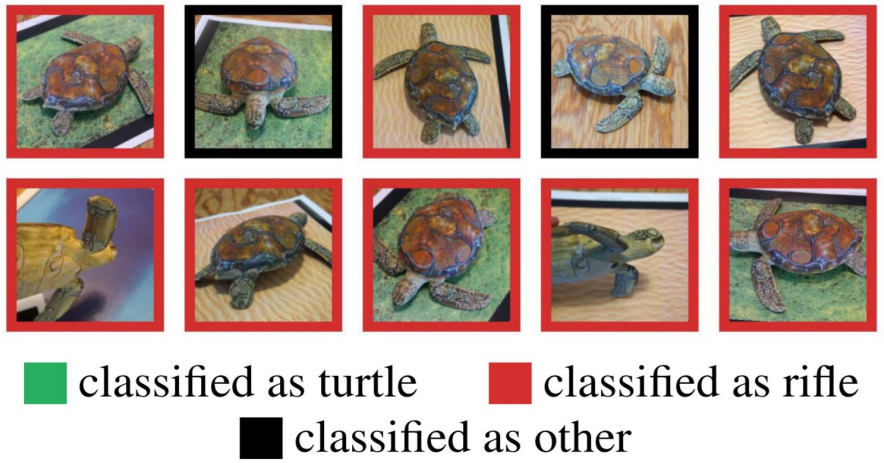
	0	1	2	3	4	5	6	7	8	9
RBF	0.30	0.05	0.18	**0.76**	-0.01	-0.06	-0.04	-0.03	-0.05	-0.00
MLP	**0.83**	0.00	0.05	0.06	0.00	0.00	0.00	0.00	0.05	0.00
CNN	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00
ENS	**0.96**	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.01	0.00
SVM-gauss	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.01
SVM-poly	0.02	0.00	0.01	**0.94**	0.00	0.00	0.00	0.00	0.01	0.02
SVM-poly4	0.01	0.00	0.00	**0.96**	0.00	0.00	0.00	0.00	0.01	0.01
SVM-sigmoid	**0.40**	0.00	0.03	0.35	0.01	0.06	0.00	0.01	0.11	0.02
SVM-linear	0.19	0.01	0.06	**0.50**	0.01	0.05	0.01	0.02	0.11	0.04

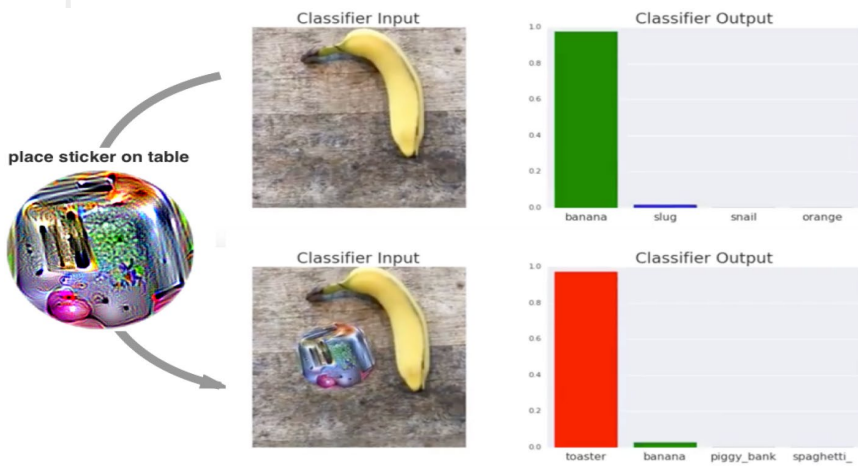
Figure 3: Model outputs for individual adversarial examples.

Vidnerová & Nerud., Vulnerability of Machine Learning Models to Adversarial Examples. ITAT, 2016

Biggio et al., Security evaluation of support vector machines in adversarial environments. Support Vector Machines Applications, 2014

Adversarial Examples in Physical World

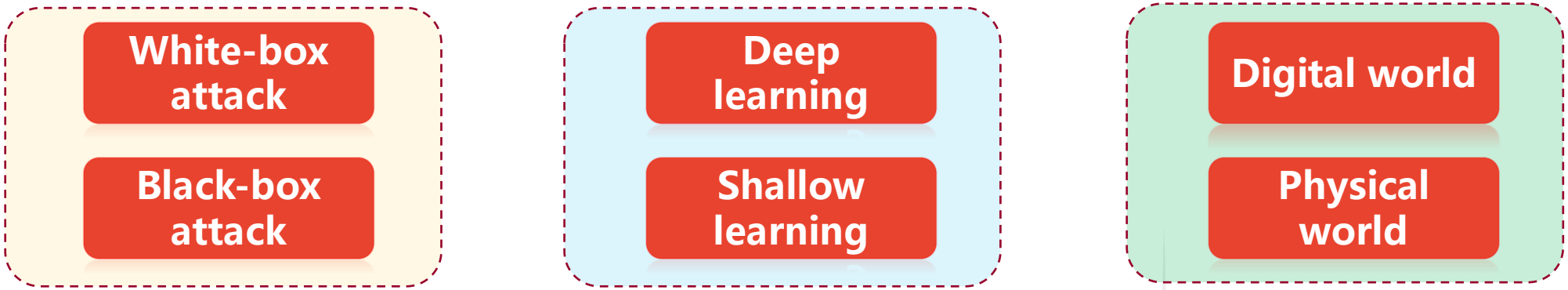




Brown et al., Adversarial Patch. 2017
Athalye et al., Synthesizing Robust Adversarial Examples. ICML 2018
Eykholt et al., Physical Adversarial Examples for Object Detectors. WOOT 2018
Thys, Ranst, Goedemé., Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. 2019
Xu et al., Evading Real-Time Person Detectors by Adversarial T-shirt. ECCV 2020
Sharif, et al., A General Framework for Adversarial Examples with Objectives. ACM Transactions on Privacy and Security 2019

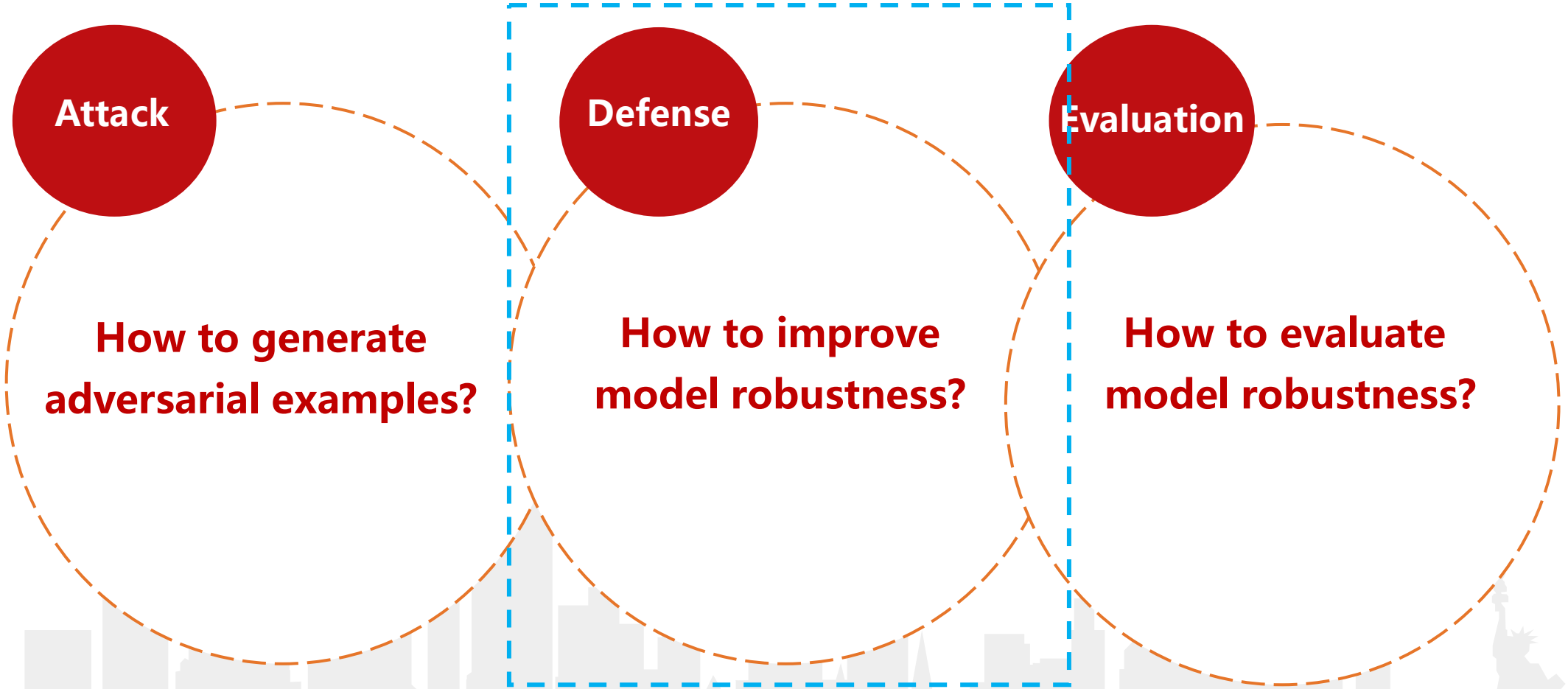
Why Do Adversarial Examples Matter?

- Whenever there is an AI model, there is (almost) a way to generate adversarial examples



- Adversarial examples exist in various domains

Key Problems in Adversarial Machine Learning



(Incomplete) List of Defense Strategies

Defensive Distillation

Secondary Classification

Normalization

Gradient Regularization

Feature Squeezing

Adversarial Training

Activation Pruning

Distributional Detection

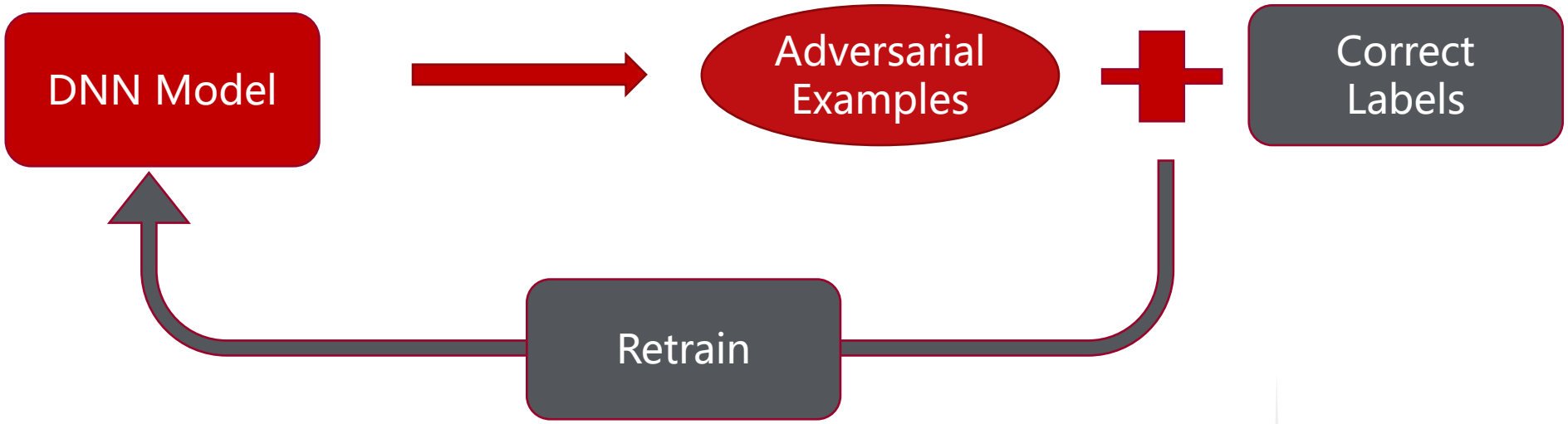
Model Compression

PCA Detection

Gradient Shattering

Stochastic Gradients

Adversarial Training



- Add adversarial examples to the training set, with their correct labels
- Robust optimization problem:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{x \in \mathcal{X}} \left[\max_{\delta \in [-\epsilon, \epsilon]^N} \ell(x + \delta; F_\theta) \right]$$

- (One of the) **strongest defense mechanism so far**
- **Not scalable enough/ vulnerable to blind-spot attack/ high sample complexity。。。**

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

Further Improves Adversarial Training

Adversarial Distributional Training for Robust Deep Learning

Zhijie Deng [*1] Yinpeng Dong [*1] Tianyu Pang [1] Hang Su [1] Jun Zhu [1]

Boosting Adversarial Training with Hypersphere Embedding

Tianyu Pang*, **Xiao Yang***, **Yinpeng Dong**, **Kun Xu**, **Hang Su**, **Jun Zhu**
Department of Computer Science and Technology
Tsinghua University, Beijing, China
{pty17, yangxiao19, dyp17}@mails.tsinghua.edu.cn
kunxu.thu@gmail.com, {suhangss, dcszj}@mail.tsinghua.edu.cn

Convergence of Adversarial Training in Overparametrized Neural Networks

Ruiqi Gao[1,*] **Tianle Cai**[1,*] **Haochuan Li**[2] **Liwei Wang**[3] **Cho-Jui Hsieh**[4] **Jason D. Lee**[5]
[1]School of Mathematical Sciences, Peking University
[2]Department of EECS, Massachusetts Institute of Technology
[3]Key Laboratory of Machine Perception, MOE, School of EECS, Peking University
[4]Department of Computer Science, University of California, Los Angeles
[5]Department of Electrical Engineering, Princeton University

Theoretically Principled Trade-off between Robustness and Accuracy

Hongyang Zhang [1 2] Yaodong Yu [3] Jiantao Jiao [4] Eric P. Xing [1 5] Laurent El Ghaoui [4] Michael I. Jordan [4]

Unlabeled Data Improves Adversarial Robustness

Yair Carmon*
Stanford University
yairc@stanford.edu

Aditi Raghunathan*
Stanford University
aditir@stanford.edu

Ludwig Schmidt
UC Berkeley
ludwig@berkeley.edu

Percy Liang
Stanford University
pliang@cs.stanford.edu

John C. Duchi
Stanford University
jduchi@stanford.edu

You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle

Dinghuai Zhang*, **Tianyuan Zhang***
Peking University
{zhangdinghuai, 1600012888}@pku.edu.cn

Yiping Lu*
Stanford University
yplu@stanford.edu

Zhanxing Zhu†
School of Mathematical Sciences, Peking University
Center for Data Science, Peking University
Beijing Institute of Big Data Research
zhanxing.zhu@pku.edu.cn

Bin Dong†

Dynamic Adversarial Training

First-Order Stationary Condition for constrained optimization (FOSC)

$$c(\mathbf{x}^k) = \max_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x} - \mathbf{x}^k, \nabla_{\mathbf{x}} f(\boldsymbol{\theta}, \mathbf{x}^k) \rangle$$

- The lower the score, the better convergence quality of the adversarial example $\mathbf{x}^k$
- Inner maximization: early (later) stages prefer low (high) convergence quality AEs

Algorithm 1 Dynamic Adversarial Training

Input: Network $h_{\boldsymbol{\theta}}$, training data S, initial model parameters $\boldsymbol{\theta}^0$, step size η_t, mini-batch $\mathcal{B}$, maximum FOSC value c_{max}, training epochs T, FOSC control epoch T', PGD step K, PGD step size α, maximum perturbation ϵ.

for $t = 0$ **to** $T-1$ **do**
  $c_t = \max(c_{\max} - t \cdot c_{\max}/T', 0)$
  for each batch $\mathbf{x}_{\mathcal{B}}^0$ **do**
    $V = \mathbb{1}_{\mathcal{B}}$ # *control vector of all elements is 1*
    while $\sum V > 0$ & $k < K$ **do**
      $\mathbf{x}_{\mathcal{B}}^{k+1} = \mathbf{x}_{\mathcal{B}}^k + V \cdot \alpha \cdot \text{sign}(\nabla_{\mathbf{x}} \ell(h_{\boldsymbol{\theta}}(\mathbf{x}_{\mathcal{B}}^k), y))$
      $\mathbf{x}_{\mathcal{B}}^k = clip(\mathbf{x}_{\mathcal{B}}^k, \mathbf{x}_{\mathcal{B}}^0 - \epsilon, \mathbf{x}_{\mathcal{B}}^0 + \epsilon)$
      $V = \mathbb{1}_{\mathcal{B}}(c(\mathbf{x}_{1\ldots\mathcal{B}}^k) \le c_t)$ # *The element of V becomes 0 at which FOSC is smaller than c_t*
    end while
    $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_t \mathbf{g}(\boldsymbol{\theta}^t)$ # $\mathbf{g}(\boldsymbol{\theta}^t)$: *stochastic gradient*
  end for
end for

Theorem 1. *Suppose Assumptions 1, 2 and 3 hold. Let $\Delta = L_S(\boldsymbol{\theta}^0) - \min_{\boldsymbol{\theta}} L_S(\boldsymbol{\theta})$. If the step size of the outer minimization is set to $\eta_t = \eta = \min(1/L, \sqrt{\Delta/L\sigma^2 T})$. Then the output of Algorithm 1 satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla L_S(\boldsymbol{\theta}^t)\|_2^2\right] \le 4\sigma\sqrt{\frac{L\Delta}{T}} + \frac{5L_{\theta x}^2\delta}{\mu},$$

where $L = (L_{\theta x}L_{x\theta}/\mu + L_{\theta\theta})$.

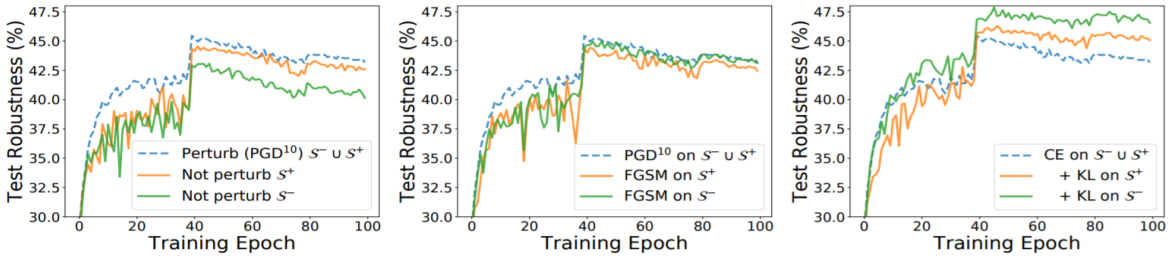
Defense	MNIST				CIFAR-10			
	FGSM	PGD-10	PGD-20	C&W$_\infty$	FGSM	PGD-10	PGD-20	C&W$_\infty$
Standard	96.12	95.73	95.73	97.20	65.65	65.80	65.60	66.12
Curriculum	96.59	95.87	96.09	97.52	71.25	71.44	71.13	71.94
Dynamic	**97.60**	**97.01**	**96.97**	**98.36**	**71.95**	**72.15**	**72.02**	**72.85**

Wang et al., On the Convergence and Robustness of Adversarial Training. ICML 2019

34

Adversarial Training with Misclassified Examples

Answer the following question: are the adversarial examples generated from **misclassified** and **correctly classified examples**, equally important for adversarial robustness?



(a) Not perturb (b) Inner maximization (c) Outer minimization

$$\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{misc}}(h_{\boldsymbol{\theta}}) := \frac{1}{n}\Big(\sum_{i\in\mathcal{S}^+_{h_{\boldsymbol{\theta}}}} \mathcal{R}^+(h_{\boldsymbol{\theta}},\mathbf{x}_i) + \sum_{i\in\mathcal{S}^-_{h_{\boldsymbol{\theta}}}} \mathcal{R}^-(h_{\boldsymbol{\theta}},\mathbf{x}_i)\Big)$$
$$= \frac{1}{n}\sum_{i=1}^{n}\big\{\mathbb{1}(h_{\boldsymbol{\theta}}(\hat{\mathbf{x}}'_i)\neq y_i) + \mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}_i)\neq h_{\boldsymbol{\theta}}(\hat{\mathbf{x}}'_i))\cdot\mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}_i)\neq y_i)\big\}$$

Treat misclassified examples separately

Algorithm 1 Misclassification Aware adveRsarial Training (MART)

Input: Training data $\{\mathbf{x}_i, y_i\}_{i=1,\dots,n}$, outer iteration number T_O, inner iteration number T_I, maximum perturbation ϵ, step size for inner optimization η_I, step size for outer optimization η_O
Initialization: Standard random initialization of $h_{\boldsymbol{\theta}}$
for $t = 1, \dots, T_O$ **do**
  Uniformly sample a minibatch of training data $B^{(t)}$
  for $\mathbf{x}_i \in B^{(t)}$ **do**
    $\mathbf{x}'_i = \mathbf{x}_i + \epsilon\cdot\xi$, with $\xi \sim \mathcal{U}(-1,1)$ # $\mathcal{U}$ is a uniform distribution
    for $s = 1, \dots, T_I$ **do**
      $\mathbf{x}'_i \leftarrow \Pi_{\mathcal{B}_\epsilon(\mathbf{x}_i)}\big(\mathbf{x}'_i + \eta_I\cdot\text{sign}(\nabla_{\mathbf{x}'_i}\text{CE}(\mathbf{p}(\mathbf{x}'_i,\boldsymbol{\theta}), y_i))\big)$ # $\Pi(\cdot)$ is the projection operator
    end for
    $\hat{\mathbf{x}}'_i \leftarrow \mathbf{x}'_i$
  end for
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_O\sum_{\mathbf{x}_i\in B^{(t)}}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i, y_i, \hat{\mathbf{x}}'_i; \boldsymbol{\theta})$
end for
Output: Robust classifier $h_{\boldsymbol{\theta}}$

Table 1: Loss function comparison with existing work. The adversarial example $\hat{\mathbf{x}}'$ is generated by (10) for all defense methods except TRADES and MMA. The adversarial example in TRADES is generated by maximizing its regularization term (KL-divergence), and the adversarial example in MMA is generated by solving (10) with different perturbation limit (*i.e.*, ϵ).

Defense Method	Loss Function
Standard	$\text{CE}(\mathbf{p}(\hat{\mathbf{x}}',\boldsymbol{\theta}), y)$
ALP	$\text{CE}(\mathbf{p}(\hat{\mathbf{x}}',\boldsymbol{\theta}), y) + \lambda\cdot\|\mathbf{p}(\hat{\mathbf{x}}',\boldsymbol{\theta}) - \mathbf{p}(\mathbf{x},\boldsymbol{\theta})\|_2^2$
CLP	$\text{CE}(\mathbf{p}(\mathbf{x},\boldsymbol{\theta}), y) + \lambda\cdot\|\mathbf{p}(\hat{\mathbf{x}}',\boldsymbol{\theta}) - \mathbf{p}(\mathbf{x},\boldsymbol{\theta})\|_2^2$
TRADES	$\text{CE}(\mathbf{p}(\mathbf{x},\boldsymbol{\theta}), y) + \lambda\cdot\text{KL}(\mathbf{p}(\mathbf{x},\boldsymbol{\theta})\|\|\mathbf{p}(\hat{\mathbf{x}}',\boldsymbol{\theta}))$
MMA	$\text{CE}(\mathbf{p}(\hat{\mathbf{x}}',\boldsymbol{\theta}), y)\cdot\mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x}) = y) + \text{CE}(\mathbf{p}(\mathbf{x},\boldsymbol{\theta}), y)\cdot\mathbb{1}(h_{\boldsymbol{\theta}}(\mathbf{x})\neq y)$
MART	$\text{BCE}(\mathbf{p}(\hat{\mathbf{x}}',\boldsymbol{\theta}), y) + \lambda\cdot\text{KL}(\mathbf{p}(\mathbf{x},\boldsymbol{\theta})\|\|\mathbf{p}(\hat{\mathbf{x}}',\boldsymbol{\theta}))\cdot(1-\mathbf{p}_y(\mathbf{x},\boldsymbol{\theta}))$

Table 3: Black-box robustness (accuracy (%) on black-box test attacks) on MNIST and CIFAR-10.

Defense	MNIST				CIFAR-10			
	FGSM	PGD^{10}	PGD^{20}	CW_∞	FGSM	PGD^{10}	PGD^{20}	CW_∞
Standard	96.12	95.73	95.47	96.34	79.98	80.27	80.01	80.85
MMA	96.11	95.94	95.81	96.87	80.28	80.52	80.48	81.32
Dynamic	97.60	96.25	95.82	97.03	81.37	81.71	81.38	82.05
TRADES	97.49	96.03	95.73	97.20	81.52	81.73	81.53	82.11
MART	**97.77**	**96.96**	**96.97**	**98.36**	**82.75**	**82.93**	**82.70**	**82.95**

Wang et al., Improving Adversarial Robustness Requires Revisiting Misclassified Examples, ICLR 2020

Adversarial Robustness Leaderboard

RobustBench Leaderboard FAQ Contribute Model Zoo

RobustBench

A standardized benchmark for adversarial robustness

The goal of **RobustBench** is to systematically track the *real* progress in adversarial robustness. There are already more than 2'000 papers on this topic, but it is still unclear which approaches really work and which only lead to overestimated robustness. We start from benchmarking the ℓ_∞- and ℓ_2-robustness since these are the most studied settings in the literature. We use AutoAttack, an ensemble of white-box and black-box attacks, to standardize the evaluation (for details see our paper). Additionally, we open source the RobustBench library that contains models used for the leaderboard to facilitate their usage for downstream applications.

Up-to-date leaderboard based on 30+ recent papers

Unified access to 20+ state-of-the-art robust models via Model Zoo

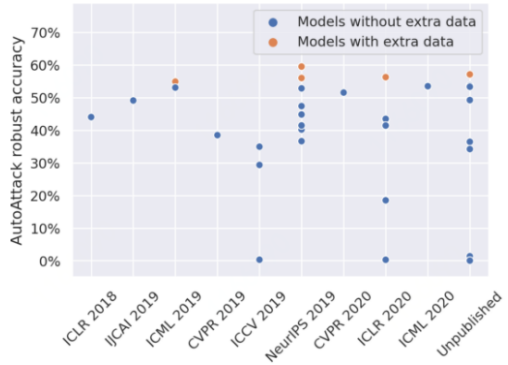
Model Zoo

Check out the available models and our Colab tutorials.

```
!pip install git+https://github.com/RobustBench/robustbench

from robustbench.utils import load_model
model = load_model(model_name='Carmon2019Unlabeled')

from robustbench.data import load_cifar10
x_test, y_test = load_cifar10(n_examples=100)

!pip install git+https://github.com/fra31/auto-attack
from autoattack import AutoAttack
adversary = AutoAttack(model, norm='Linf', eps=8/255)
x_adv = adversary.run_standard_evaluation(x_test, y_test)
```

Analysis

Check out our paper with a detailed analysis.



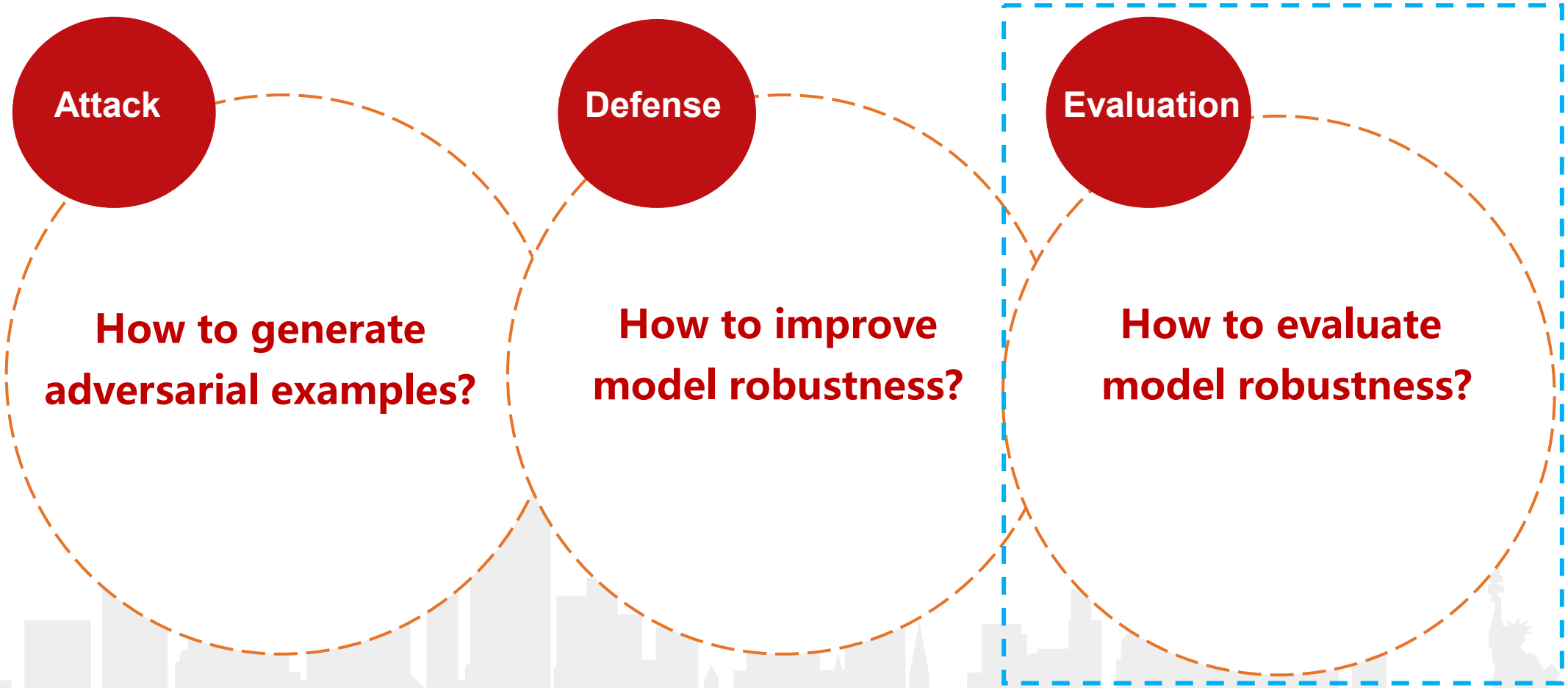
Leaderboard: CIFAR-10, $\ell_\infty = 8/255$, Untargeted, AutoAttack

Rank	Method	Standard accuracy	Robust accuracy	Extra data	Architecture	Venue
1	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples *We show the robust accuracy reported in the paper since AutoAttack performs slightly worse (65.88%).*	91.10%	65.87%	☑	WideResNet-70-16	arXiv, Oct 2020
2	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples *We show the robust accuracy reported in the paper since AutoAttack performs slightly worse (62.80%).*	89.48%	62.76%	☑	WideResNet-28-10	arXiv, Oct 2020
3	Adversarial Weight Perturbation Helps Robust Generalization	88.25%	60.04%	☑	WideResNet-28-10	NeurIPS 2020
4	Does Network Width Really Help Adversarial Robustness?	85.60%	59.78%	☑	WideResNet-34-15	arXiv, Oct 2020
5	Unlabeled Data Improves Adversarial Robustness	89.69%	59.53%	☑	WideResNet-28-10	NeurIPS 2019
6	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples *We show the robust accuracy reported in the paper since AutoAttack performs slightly worse (57.20%).*	85.29%	57.14%	×	WideResNet-70-16	arXiv, Oct 2020
7	HYDRA: Pruning Adversarially Robust Neural Networks	88.98%	57.14%	☑	WideResNet-28-10	NeurIPS 2020
8	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples *We show the robust accuracy reported in the paper since AutoAttack performs slightly worse (56.86%).*	85.64%	56.82%	×	WideResNet-34-20	arXiv, Oct 2020
9	Improving Adversarial Robustness Requires Revisiting Misclassified Examples	87.50%	56.29%	☑	WideResNet-28-10	ICLR 2020
10	Adversarial Weight Perturbation Helps Robust Generalization	85.36%	56.17%	×	WideResNet-34-10	NeurIPS 2020

https://robustbench.github.io/

JDAI Research

Key Problems in Adversarial Machine Learning



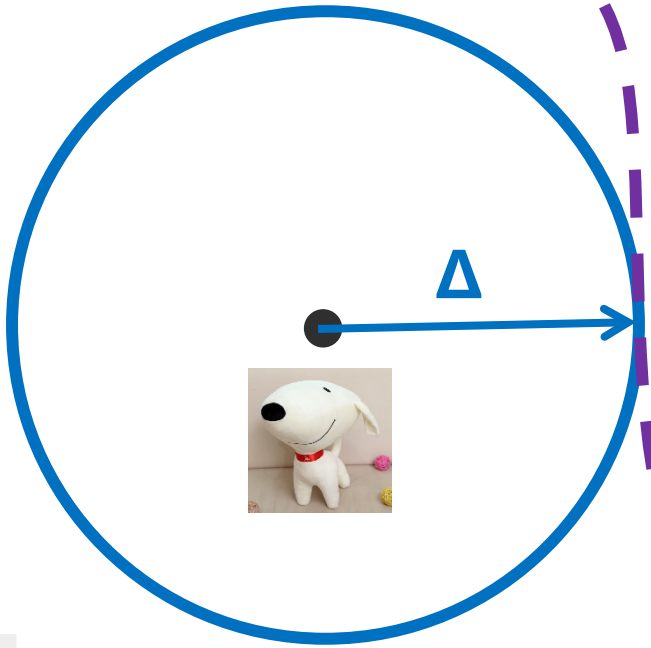
JDAI Research

Robustness Evaluation

Amount of Perturbation

Δ

Maximum Safe Perturbation
=
Minimal Adversarial Perturbation



Unfortunately, finding an exact Δ for DNNs is **NP-Complete**
Can only approximately estimate it

Decision boundary

Katz et al., Reluplex: An efficient SMT solver for verifying deep neural networks. ICCAV 2017

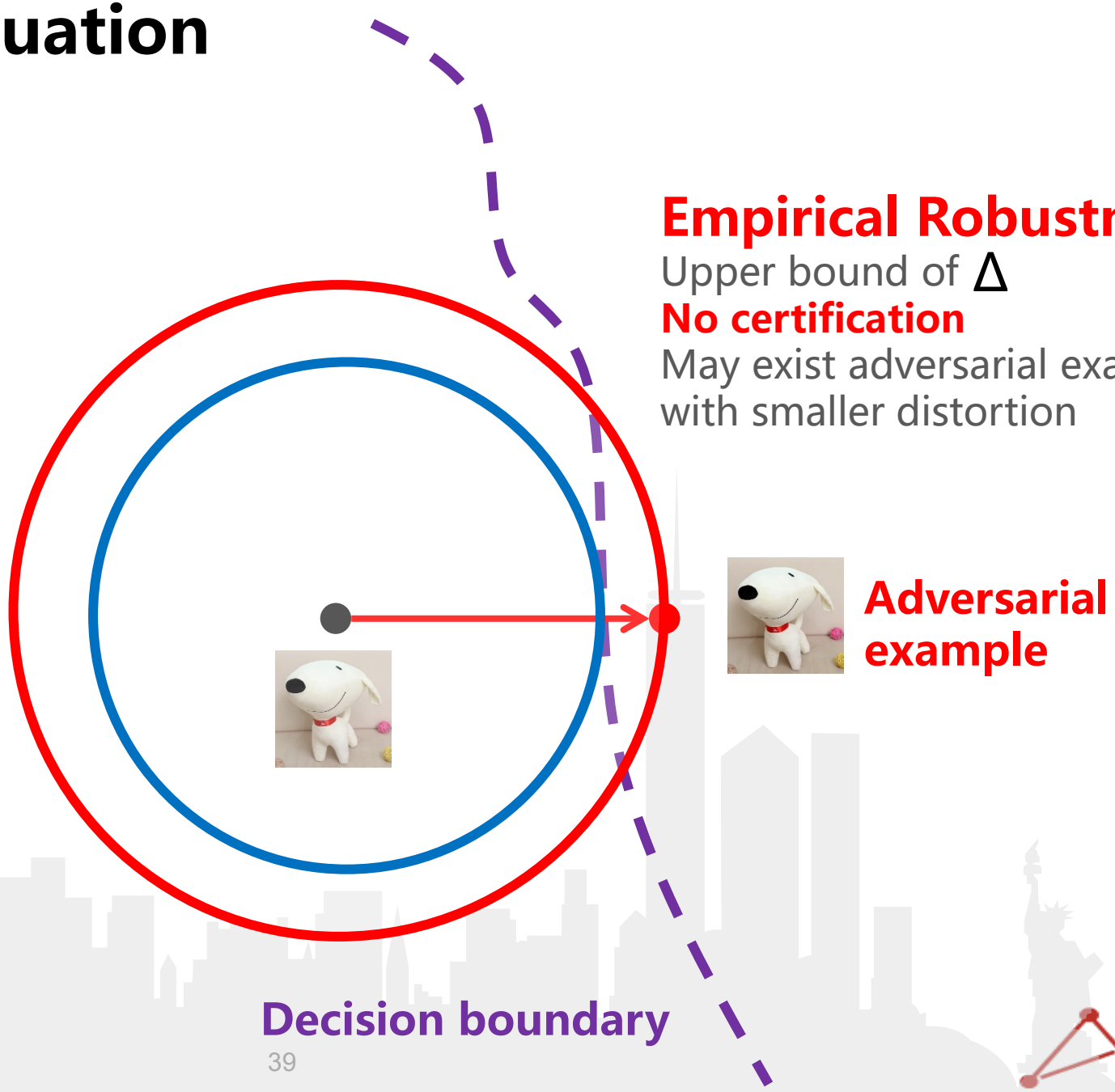
JDAI Research

Robustness Evaluation

Amount of Perturbation

Upper Bound

Δ Maximum Safe Perturbation



Empirical Robustness

Upper bound of Δ

No certification

May exist adversarial examples with smaller distortion

Adversarial example

Decision boundary

JDAI Research

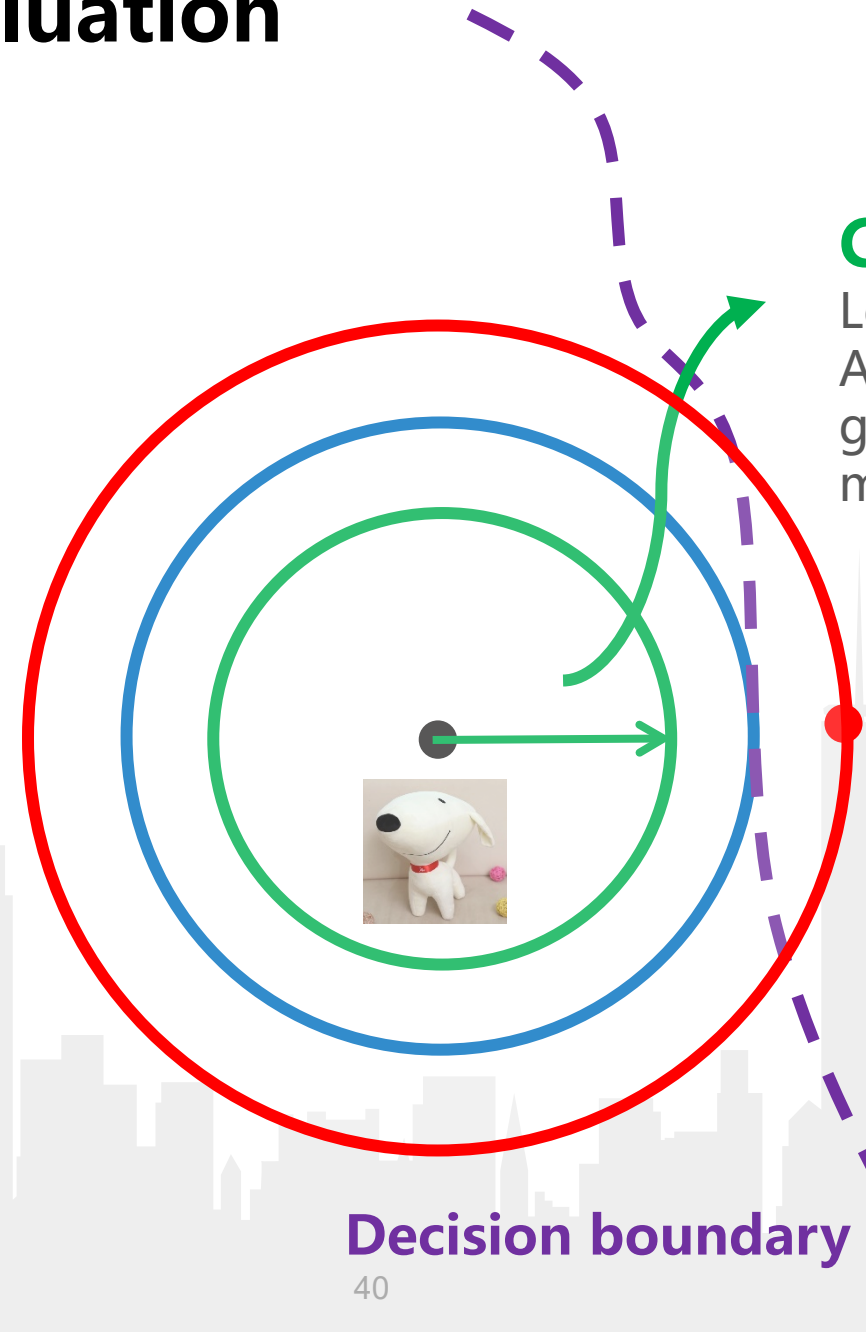
Robustness Evaluation

Amount of Perturbation

Upper Bound

Δ Maximum Safe Perturbation

Lower Bound



Certified Robustness

Lower bound of Δ
Any perturbations within green region cannot cause misclassification

Adversarial example

Decision boundary

JDAI Research

Robustness Estimation: CLEVER Score

- **CLEVER: C**ross **L**ipschitz **E**xtreme **V**alue for n**E**twork **R**obustness
- **First attack-independent** robustness metric that can be applied to **any neural network classifier**

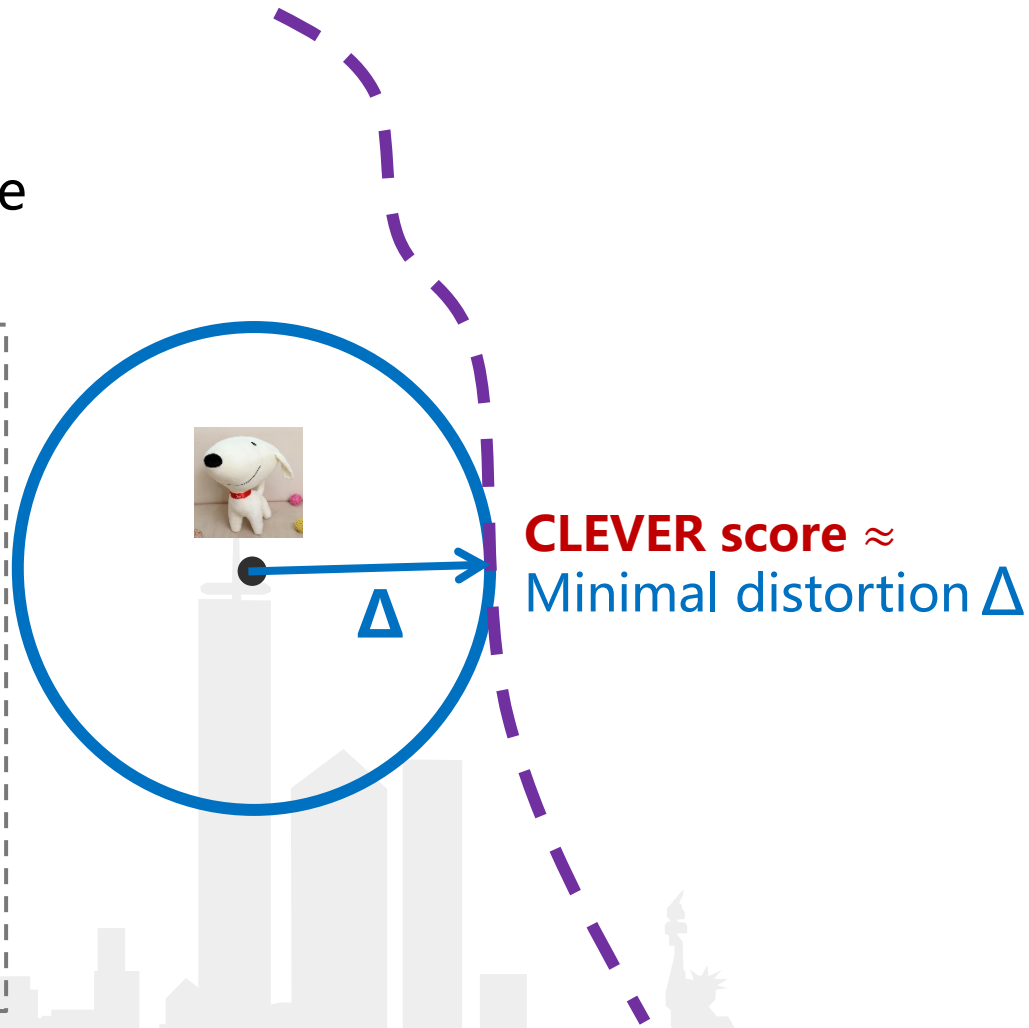
Theorem 3.1 (Formal guarantee on lower bound β_L). *Let $\boldsymbol{x_0} \in \mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}^K$ be a multi-class classifier with continuously differentiable components f_i and let $c = \text{argmax}_{1\le i\le K} f_i(\boldsymbol{x_0})$ be the class which f predicts for $\boldsymbol{x_0}$. For all $\boldsymbol{\delta} \in \mathbb{R}^d$ with*

$$\|\boldsymbol{\delta}\|_p \le \min_{j\neq c} \frac{f_c(\boldsymbol{x_0}) - f_j(\boldsymbol{x_0})}{L_q^j}, \tag{1}$$

$\text{argmax}_{1\le i\le K} f_i(\boldsymbol{x_0}+\boldsymbol{\delta}) = c$ holds with $\frac{1}{p}+\frac{1}{q} = 1, 1 \le p, q \le \infty$ and L_q^j is the Lipschitz constant for the function $f_c(\boldsymbol{x}) - f_j(\boldsymbol{x})$ in ℓ_p norm.

Remark: $\beta_L = \min_{j\neq c} \frac{f_c(\boldsymbol{x_0})-f_j(\boldsymbol{x_0})}{L_q^j}$ is a lower bound of minimum distortion.

$$L_q^j = \max \|\nabla g(\boldsymbol{x})\|_q, \text{ where } g(\boldsymbol{x_0}) = f_c(\boldsymbol{x_0}) - f_j(\boldsymbol{x_0})$$



Weng, et al. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach. ICLR 2018

JDAI Research

Robustness Estimation: CLEVER Score

- Efficiently estimate the Lipschitz constant by sampling around input + **extreme value theory**

Algorithm 1: CLEVER-t, compute **CLEVER** score for **t**argeted attack

Input: a K-class classifier $f(\boldsymbol{x})$, data example $\boldsymbol{x_0}$ with predicted class c, target class j, batch size N_b, number of samples per batch N_s, perturbation norm p, maximum perturbation R
Result: CLEVER Score $\mu \in \mathbb{R}_+$ for target class j

$S \leftarrow \{\emptyset\}$, $g(\boldsymbol{x}) \leftarrow f_c(\boldsymbol{x}) - f_j(\boldsymbol{x})$, $q \leftarrow \frac{p}{p-1}$.
for $i \leftarrow 1$ **to** N_b **do**
  for $k \leftarrow 1$ **to** N_s **do**
    randomly select a point $\boldsymbol{x}^{(i,k)} \in B_p(\boldsymbol{x_0}, R)$
    compute $b_{ik} \leftarrow \|\nabla g(\boldsymbol{x}^{(i,k)})\|_q$ via back propagation
  end
  $S \leftarrow S \cup \{\max_k\{b_{ik}\}\}$
end
$\hat{a}_W \leftarrow$ MLE of location parameter of reverse Weibull distribution on S
$\mu \leftarrow \min(\frac{g(\boldsymbol{x_0})}{\hat{a}}, R)$

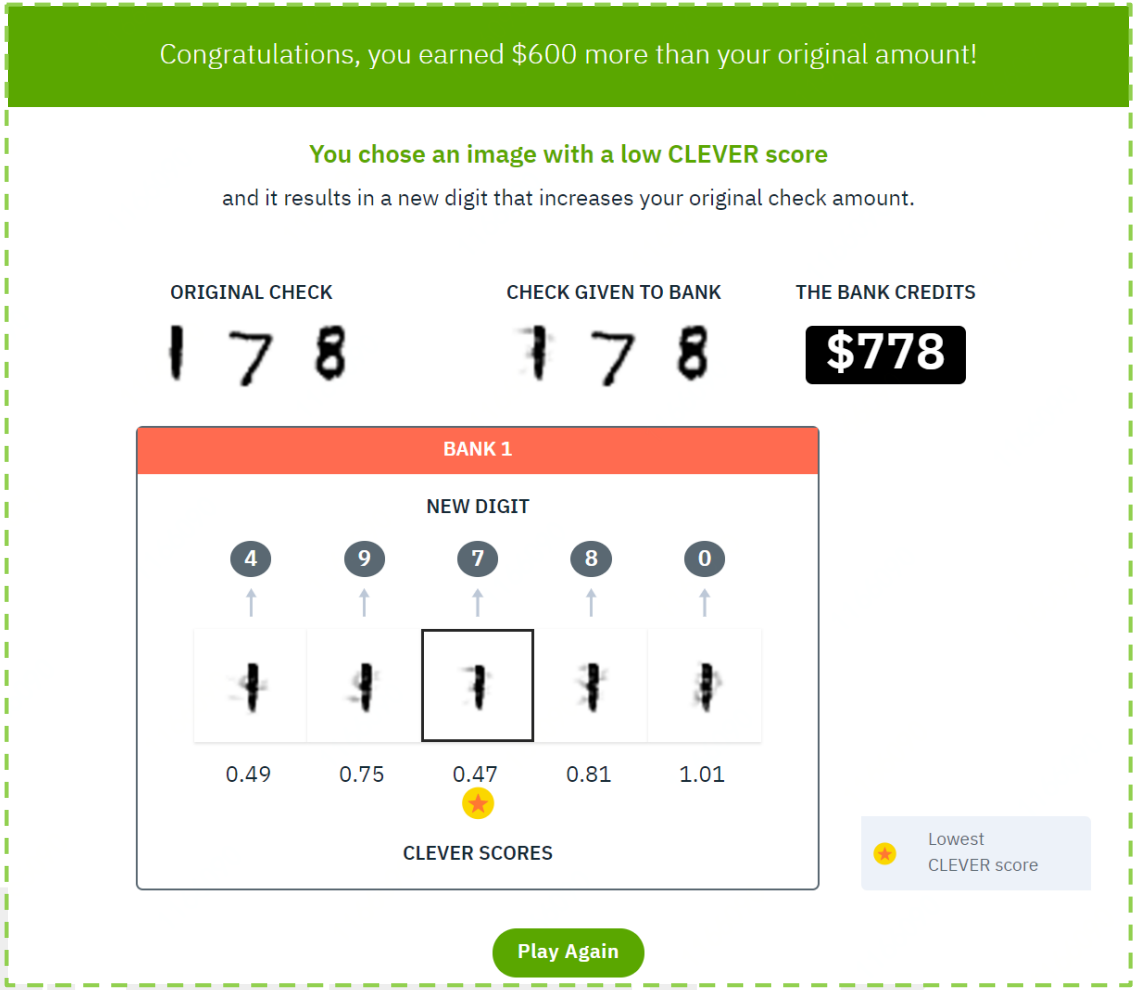
Algorithm 2: CLEVER-u, compute **CLEVER** score for **u**n-targeted attack

Input: Same as Algorithm 1, but without a target class j
Result: CLEVER score $\nu \in \mathbb{R}_+$ for un-targeted attack

for $j \leftarrow 1$ **to** K, $j \neq c$ **do**
  $\mu_j \leftarrow$ CLEVER-t$(f, \boldsymbol{x_0}, c, j, N_b, N_s, p, R)$
end
$\nu \leftarrow \min_j\{\mu_j\}$

- CLEVER score enables robustness comparison between
  - **different models**
  - **different datasets**
  - **different neural network architectures**
  - **different defense mechanisms**

Demo: http://bigcheck.mybluemix.net/



Weng, et al. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach. ICLR 2018

JDAI Research

Robustness Evaluation Metrics

CLEVER
Weng et al., ICLR 2018

Fast-Lin
Weng, et al., ICML 2018

CROWN
Zhang et al., NeurIPS 2018

Neurify
Wang et al., NeurIPS 2018

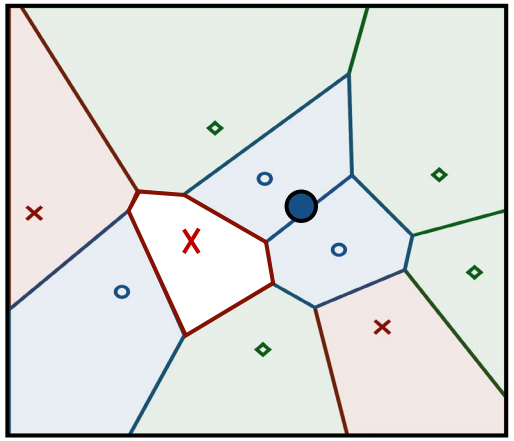
DeepZ
Singh et al., NeurIPS 2018

DeepPoly
Singh et al., POPL 2019

CNN-Cert
Boopathy et al., AAAI 2019

Robustness Verification: Nearest Neighbor Classifiers

Robustness verification for ReLU network (Katz et al., 2017) and tree ensemble (Kantchelian et al., 2016) are NP-complete. How about nearest neighbor classifiers?



Finding the minimum perturbation to make it closest to x_j

$$\epsilon^{(j)} = \min_{\delta} \frac{1}{2}\|\delta\|^2 \quad \text{s.t.} \quad \|z+\delta-x_j\|^2 \le \|z+\delta-x_i\|^2, \forall i \neq j$$

A quadratic programming problem with linear constraint, which is polynomial time solvable.

$$\epsilon^{(j)} = \min_{\delta} \frac{1}{2}\|\delta\|^2 \quad \text{s.t.} \quad A^{(j)}\delta + b^{(j)} \ge 0$$

where $A^{(j)}_{i,:} = x_j - x_i$, $b^{(j)}_i = \frac{\|z-x_i\|^2 - \|z-x_j\|^2}{2}$.

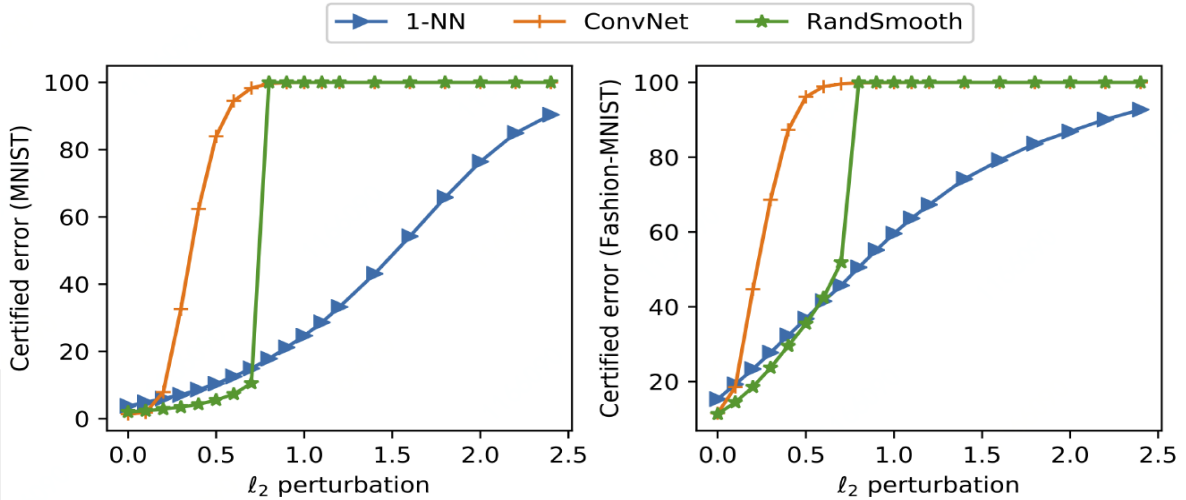
Algorithm 1: Computing minimum adversarial perturbation

Input: Target instance z, database $\{(x_j, y_j)\}_{j=1}^n$.

Initial $\epsilon = \infty$;
Sort subproblems $\{D^{(j)}\}_{j:y_j \neq 1}$ by ascending distances of $\|z - x_j\|$;
for *each j (according to the sorted order)* **do**
  if *not screenable via* (9) **then**
    Solve the subproblem via greedy coordinate ascent with screening rule (8);
    Update ϵ if we get a smaller value;
  end
end

O(n) quadratic problems, each QP has O(n) variables to solve, total time complexity >$O(n^3)$
Speed up the algorithm by

- greedy coordinate ascent algorithm
- a screening rule to remove variables in each dual QP problem
- removing unimportant subproblems without solving them



The first robustness verification for nearest neighbor models

Wang et al., Evaluating the Robustness of Nearest Neighbor Classifiers: A Primal-Dual Perspective, 2019

Robustness Verification: Distance Metric Learning

The first adversarial verification method and the first certified defense for distance metric learning.

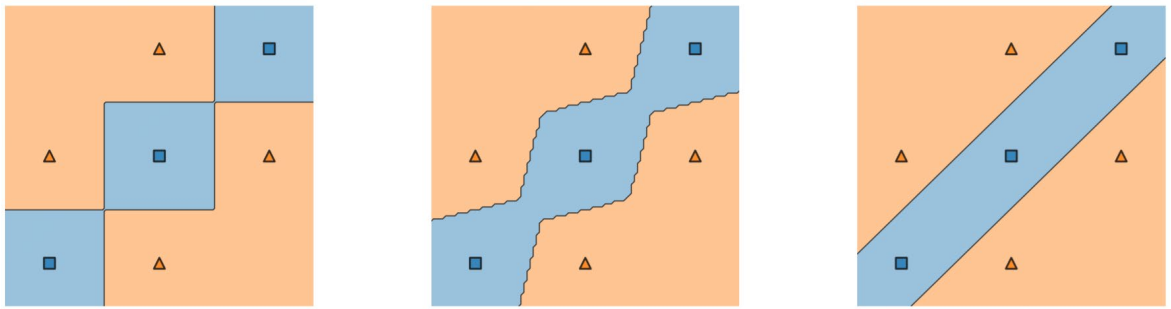
Compute a **lower bound of the minimal adversarial perturbation** of **Mahalanobis K-NN**

Theorem 1 (Robustness verification for Mahalanobis K-NN). *Given a Mahalanobis K-NN classifier parameterized by a neighbor parameter K, a training dataset $\mathbb{S}$ and a positive semi-definite matrix $\boldsymbol{M}$, for any instance $(\boldsymbol{x}_{test}, y_{test})$ we have*

$$\epsilon^*(\boldsymbol{x}_{test}, y_{test}; \boldsymbol{M}) \geq \underset{j:y_j \neq y_{test}}{k\text{th min}} \ \underset{i:y_i = y_{test}}{k\text{th max}} \ \tilde{\epsilon}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_{test}; \boldsymbol{M}), \tag{10}$$

where kth max *and* kth min *select the k-th maximum and k-th minimum respectively with* $k = (K+1)/2$.

where $$\tilde{\epsilon}(\boldsymbol{x}^+, \boldsymbol{x}^-, \boldsymbol{x}; \boldsymbol{M}) = \frac{d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}^-) - d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}^+)}{2\sqrt{(\boldsymbol{x}^+ - \boldsymbol{x}^-)^\top \boldsymbol{M}^\top \boldsymbol{M} (\boldsymbol{x}^+ - \boldsymbol{x}^-)}}.$$



(a) Euclidean (b) NCA [15] (c) ARML (Ours)

Figure 1: Decision boundaries of 1-NN with different Mahalanobis distances.

Train a **robust Mahalanobis distance with small certified and empirical robust errors**

Algorithm 1: Adversarially robust metric learning (ARML)

Input: Training data $\mathbb{S}$, number of epochs T.
Output: Positive semi-definite matrix $\boldsymbol{M}$.

Initialize $\boldsymbol{G}$ and $\boldsymbol{M}$ as identity matrices ;
for $t = 0 \ldots T-1$ **do**
  Update $\boldsymbol{G}$ with the gradient
  $\mathbb{E}_{(\boldsymbol{x},y)\in\mathbb{S}} \nabla_{\boldsymbol{G}} \ell\left(\tilde{\epsilon}\left(\text{randnear}^+_{\boldsymbol{M}}(\boldsymbol{x}), \text{randnear}^-_{\boldsymbol{M}}(\boldsymbol{x}), \boldsymbol{x}; \boldsymbol{G}^\top \boldsymbol{G}\right)\right)$;
  Update $\boldsymbol{M}$ with the constraint $\boldsymbol{M} = \boldsymbol{G}^\top \boldsymbol{G}$;
end

Table 2: Certified robust errors (left) and empirical robust errors (right) of Mahalanobis K-NN. The best (minimum) robust errors among all methods are in bold.

		Certified robust errors						Empirical robust errors					
	ℓ_2-radius	0.000	0.500	1.000	1.500	2.000	2.500	0.000	0.500	1.000	1.500	2.000	2.500
MNIST	Euclidean	0.038	0.134	0.360	0.618	0.814	0.975	0.031	0.063	0.104	0.155	0.204	0.262
	NCA	**0.030**	0.175	0.528	0.870	0.986	1.000	**0.027**	0.063	0.120	0.216	0.330	0.535
	LMNN	0.040	0.669	1.000	1.000	1.000	1.000	0.036	0.121	0.336	0.775	0.972	1.000
	ITML	0.106	0.731	0.943	1.000	1.000	1.000	0.084	0.218	0.355	0.510	0.669	0.844
	LFDA	0.237	1.000	1.000	1.000	1.000	1.000	0.215	1.000	1.000	1.000	1.000	1.000
	ARML (Ours)	0.034	**0.101**	**0.276**	**0.537**	**0.760**	**0.951**	0.032	**0.055**	**0.077**	**0.109**	**0.160**	**0.213**

ARML is more robust both provably (in terms of the certified robust error) **and empirically** (in terms of the empirical robust error).

Wang et al., Provably Robust Metric Learning. NeurIPS 2020

Adversarial ML Problems from Industry's Point of View

- Accuracy is **not the sole metric** to grade an AI model, **neither is robustness**
  - The industry needs **accurate, effective, robust,** and sometimes **fair and interpretable AI models.**
- **Scalability** is one of the key problems

- Defense **in the real-world**
  - **Robustness of robust models**

- **Plug and Play robust module** is necessary for protecting AI models that have been deployed

Thanks!

We Strive to Build Trustworthy AI
Welcome to join us!

yijinfeng@jd.com

JD AI Research