[70240413 Statistical Machine Learning, Spring, 2019]

Adversarial Robustness

Hang Su
suhangss@mail.tsinghua.edu.cn
http://www.suhangss.me
Department of Computer Science and Technology
Tsinghua University
May 8th, 2019

Machine Learning: The Success Story

- Artificial intelligence (AI) is a transformative technology that holds promise for tremendous societal and economic benefit , which has made dramatic success in a torrent of applications
- AI has the potential to revolutionize how we live, work, learn, discover, and communicate.

Machine Learning: The Success Story

IS "DEEP LEARNING" A REVOLUTION IN ARTIFICIAL INTELLIGENCE?

Andrew Ng
@AndrewYNg
Follow

"AI is the new electricity!" Electricity transformed countless industries; AI will now do the same.

2016: The Year That Deep Learning Took Over t

WHY DEEP LEARNING IS SUDDENLY CHANGING YOUR LIFE

We Are Living in the Best of the Worlds...

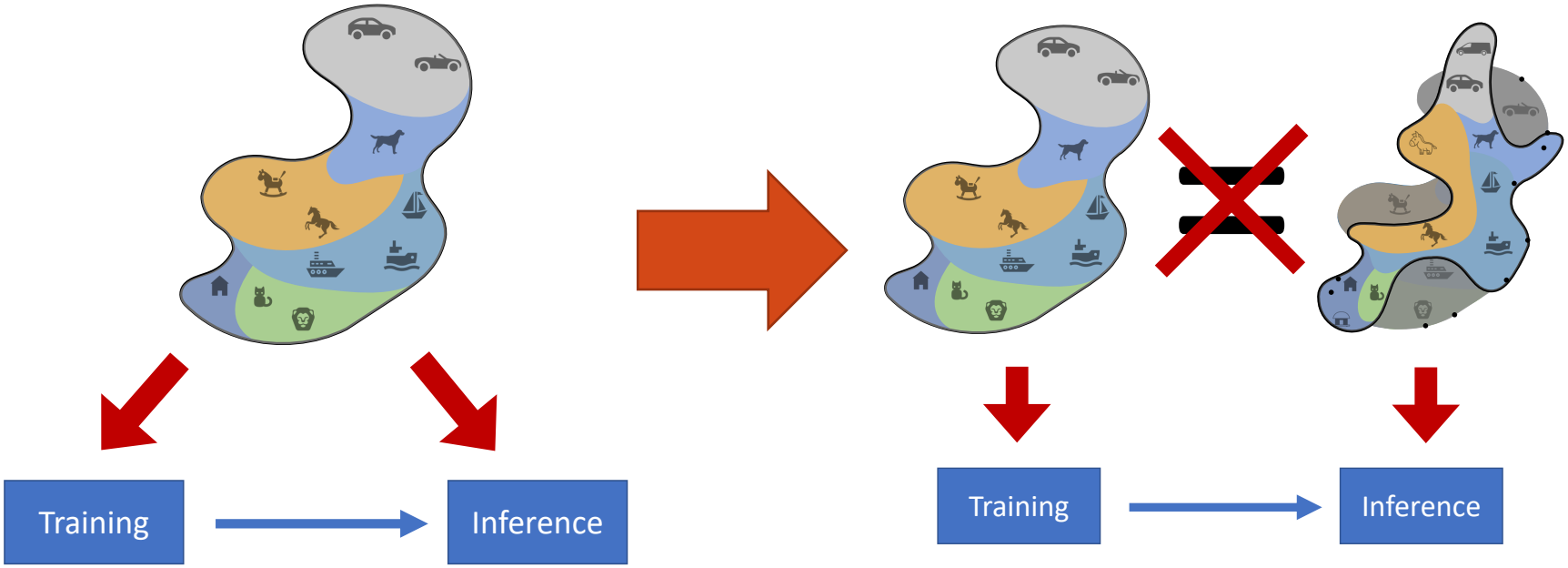
- AI is going to transform industry and business as electricity did about a century ago

——Andrew Ng, Jan. 2017

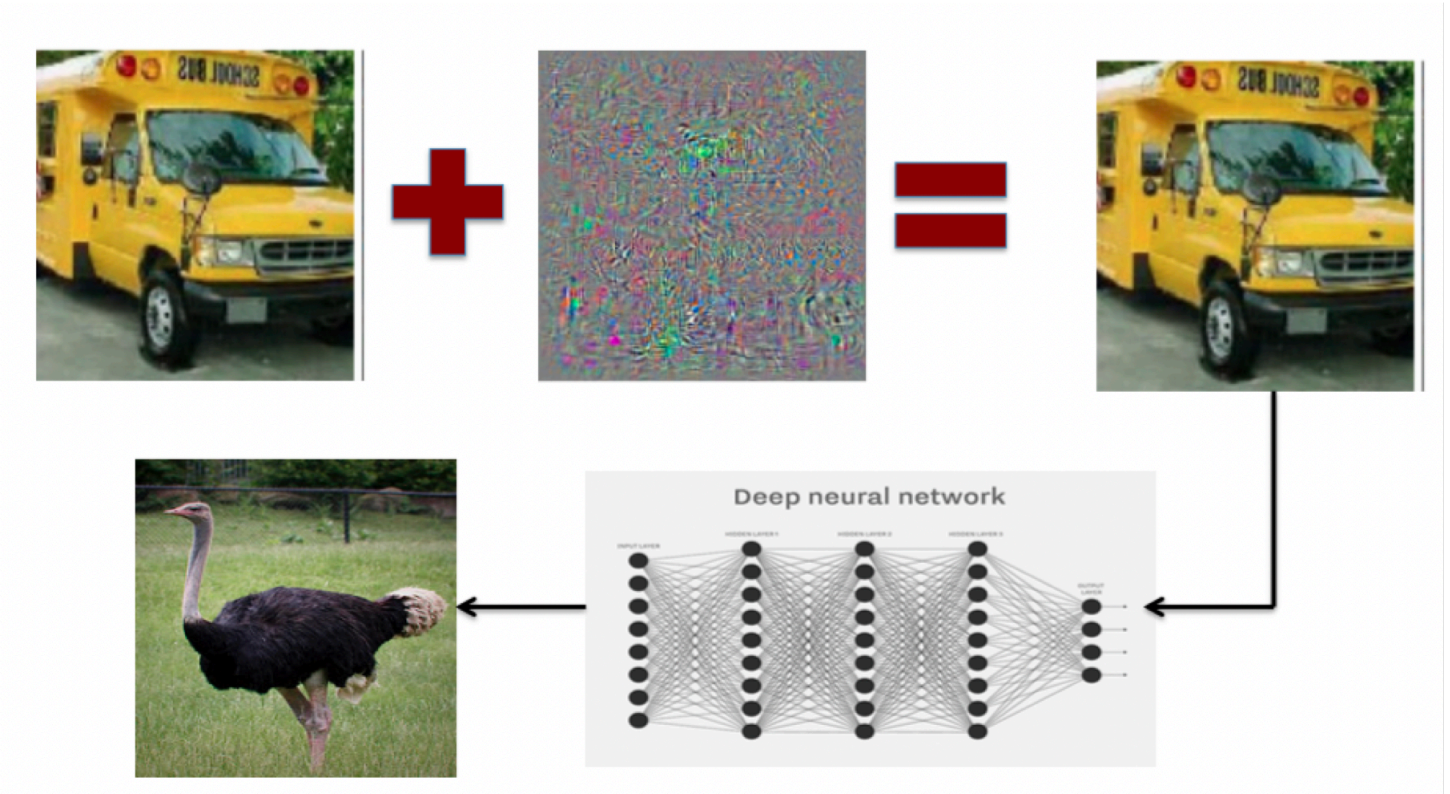
Is ML **truly** ready for real-world deployment?

A Limitation of the ML Framework

- **Measure of performance:** Fraction of mistakes during testing
- **But:** In reality, the distributions we **use** ML on are NOT the ones we **train** it on



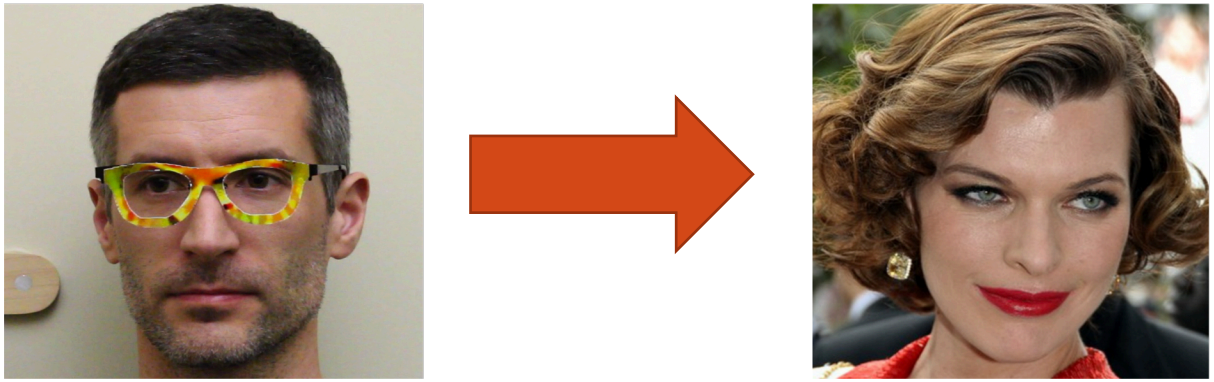
Adversarial School Bus ?



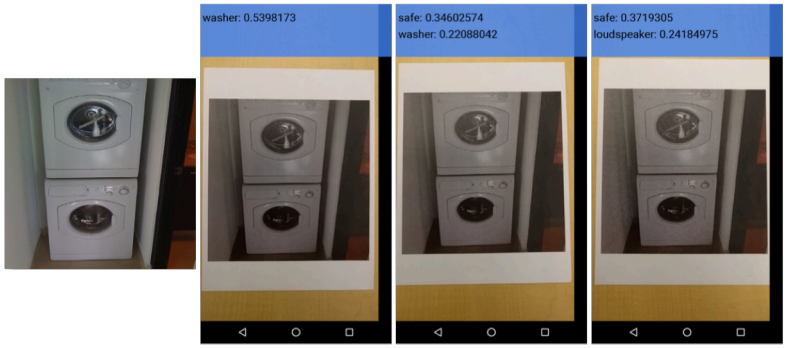
- *Szegedy et al.*, Intriguing properties of neural networks, **ICLR 2014**
- *Biggio, Roli et al.*, Evasion attacks against machine learning at test time, **ECML-PKDD 2013**

Adversarial Glasses

- M. Sharif et al. (ACM CCS 2016) attacked deep neural networks for face recognition with carefully-fabricated eyeglass frames
- When worn by a 41-year-old white male (left image), the glasses mislead the deep network into believing that the face belongs to the famous actress Milla Jovovich



ML Predictions Are (Mostly) Accurate but Brittle



[Kurakin Goodfellow Bengio 2017]

[Athalye Engstrom Ilyas Kwok 2017]

[Sharif Bhagavatula Bauer Reiter 2016]

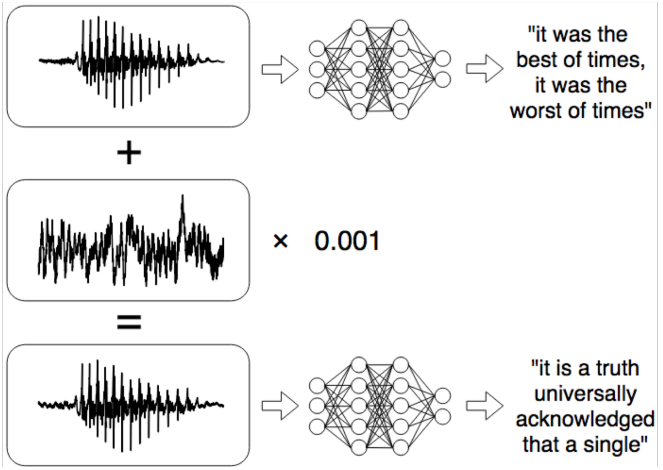
[Eykholt Evtimov 2017]

Beyond image recognition...

Audio **Transcription by Mozilla DeepSpeech**

“without the dataset the article is useless”

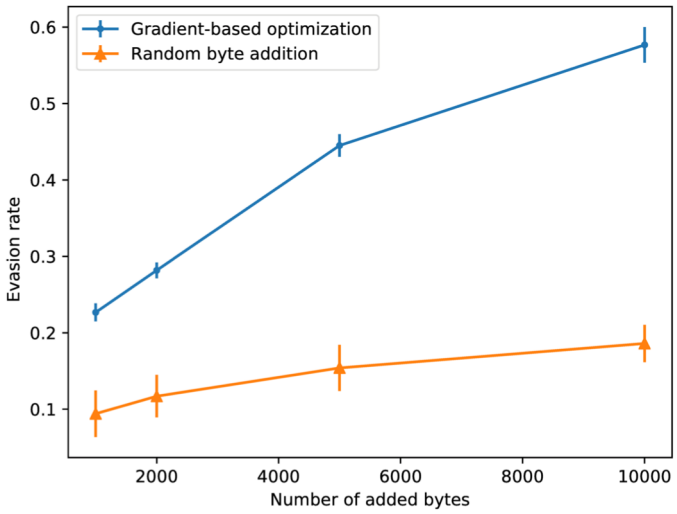
“okay google browse to evil dot com”

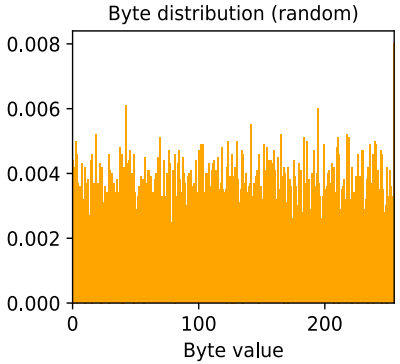


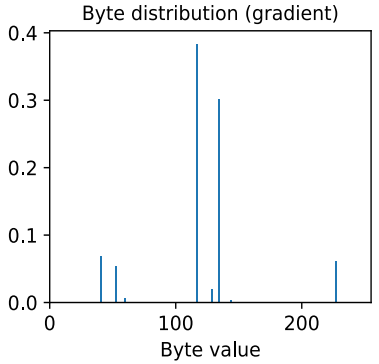
[Carlini Wagner 2018]:
Voice commands that are unintelligible to humans
https://nicholas.carlini.com/code/audio_adversarial_examples/

Beyond image recognition...

- **Deep Neural Networks for EXE Malware Detection**
  - **MalConv**: convolutional deep network trained on raw bytes to detect EXE malware
  - *Gradient-based attacks* can evade it by adding few padding bytes

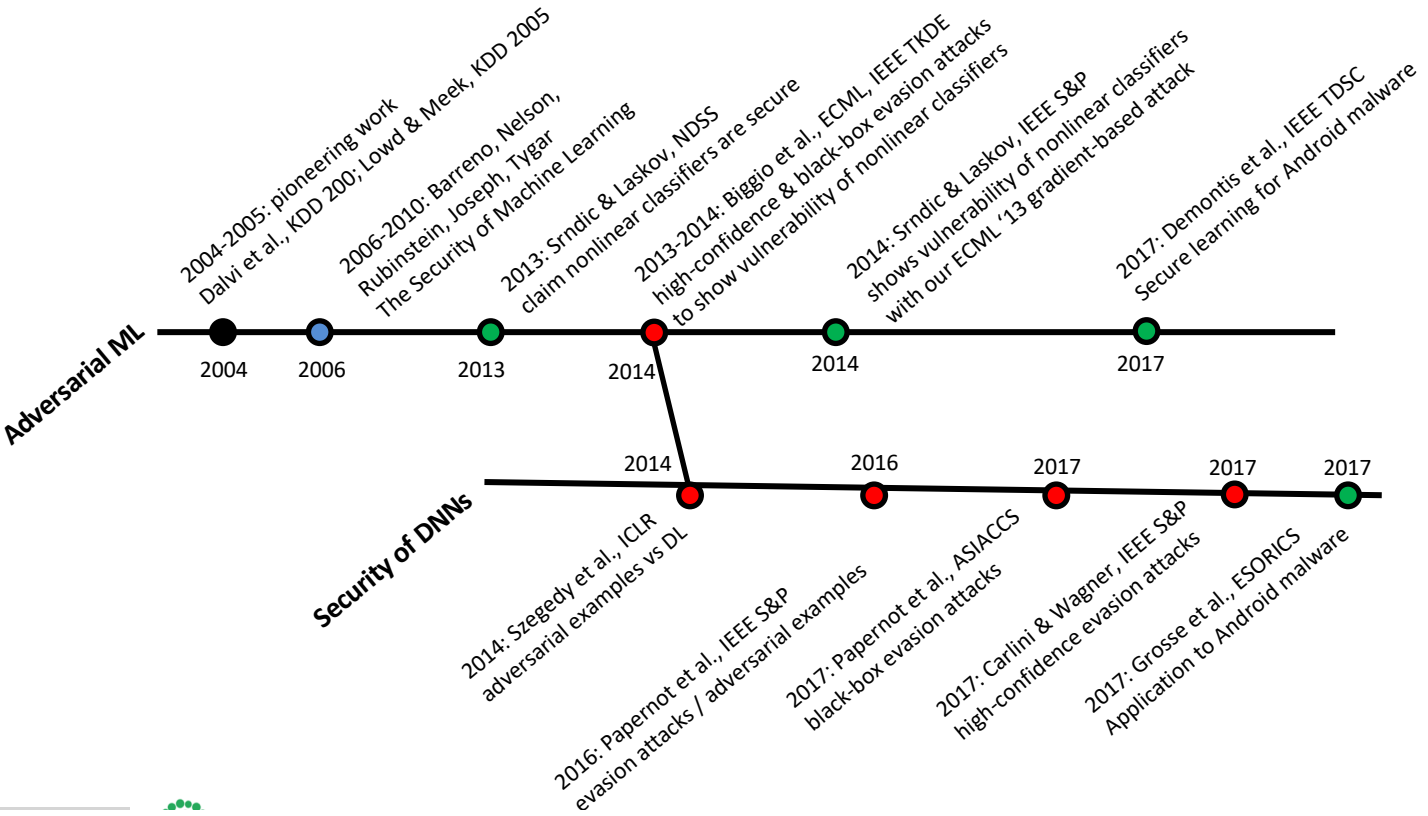






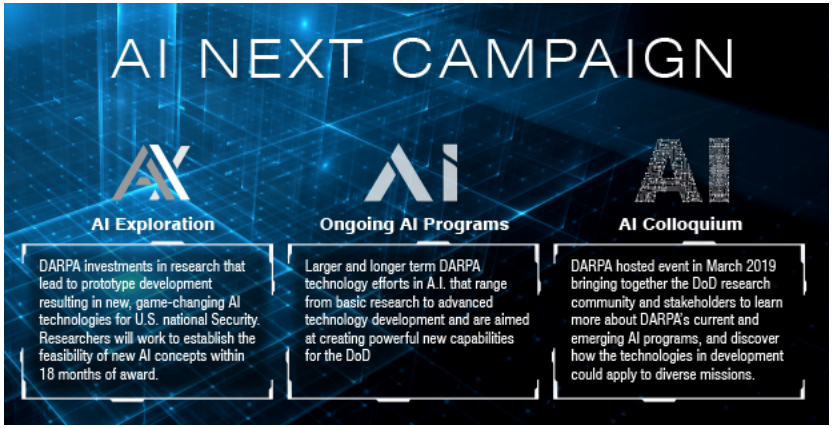
[Kolosniaji, Biggio, Roli et al., Adversarial Malware Binaries, EUSIPCO2018]

Timeline of Learning Security



Adversarial Attacks

- DARPA announced a multi-year investment in new and existing pro
- **Adversarial AI:** ML systems can be easily duped by changes to inputs that would never fool a human. To improve the robustness and reliability of AI systems and enhance the security and resiliency of machine learning and AI technologies



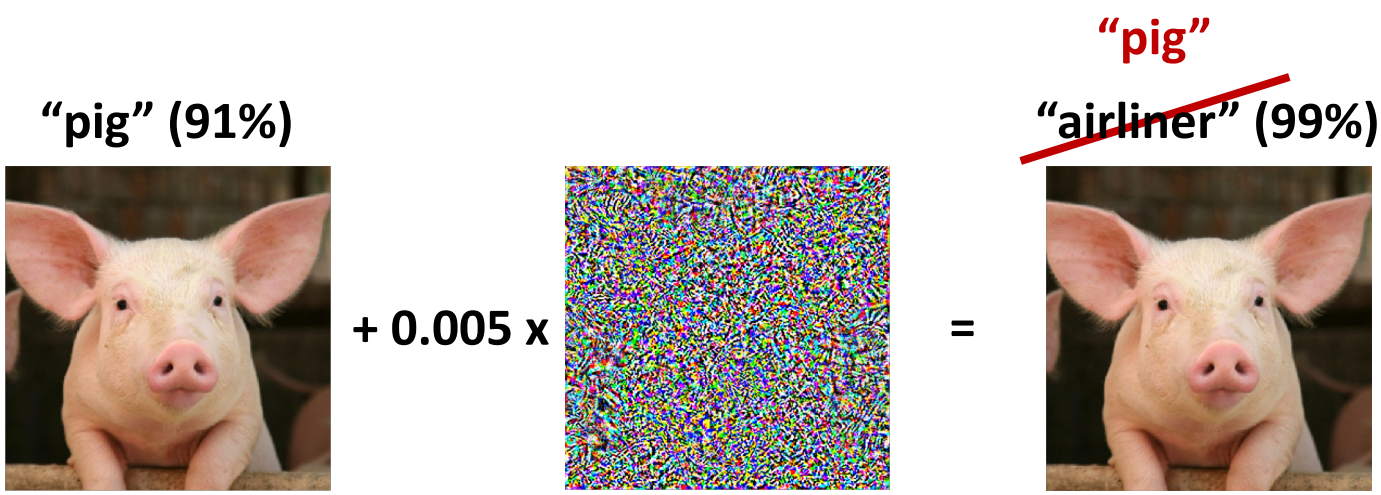
Adversarial Attacks

- We are living exciting time for *machine learning*
- Our work feeds a lot of **consumer technologies** for **personal applications**…

This opens up new big possibilities,
but also new *security risks*

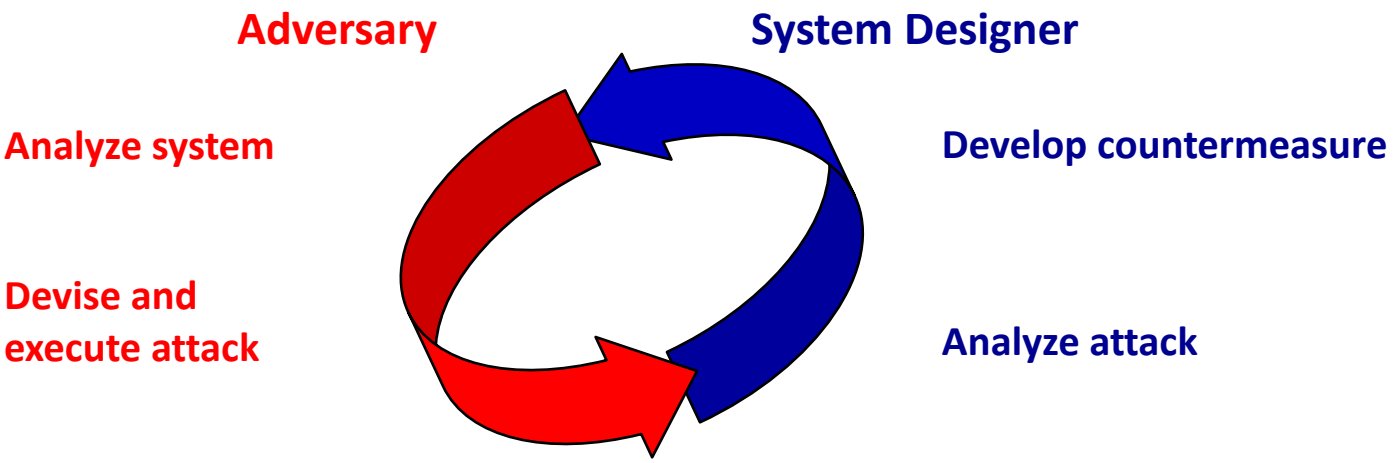
Is ML inherently not reliable ?

- **No:** But we need to re-think how we do ML
- Adversarial aspects = stress-testing our solutions
- Towards Adversarial Robust Models



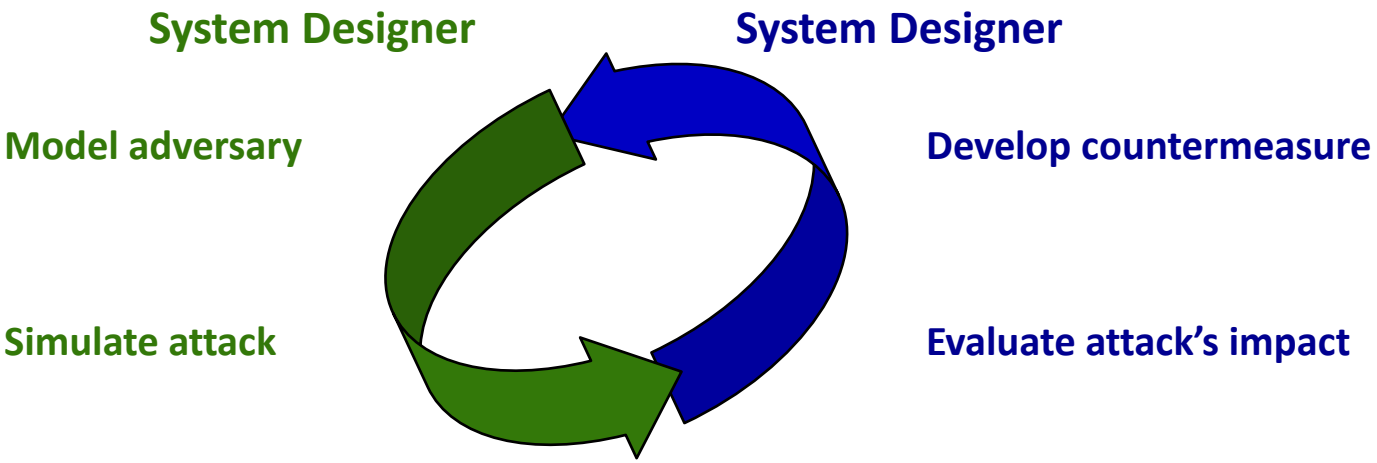
Adversary-aware Machine Learning

- Machine learning systems should be aware of the ***arms race*** with the adversary



Adversary-aware Machine Learning

- Machine learning systems should be aware of the ***arms race*** with the adversary
  - Know your adversary
  - Be proactive
  - Protect your classifier



Adversary's 3D Model

- **Know your adversary :** if you know yourself and your enemy, you need not fear the result of a hundred battles

Adversary's Goal

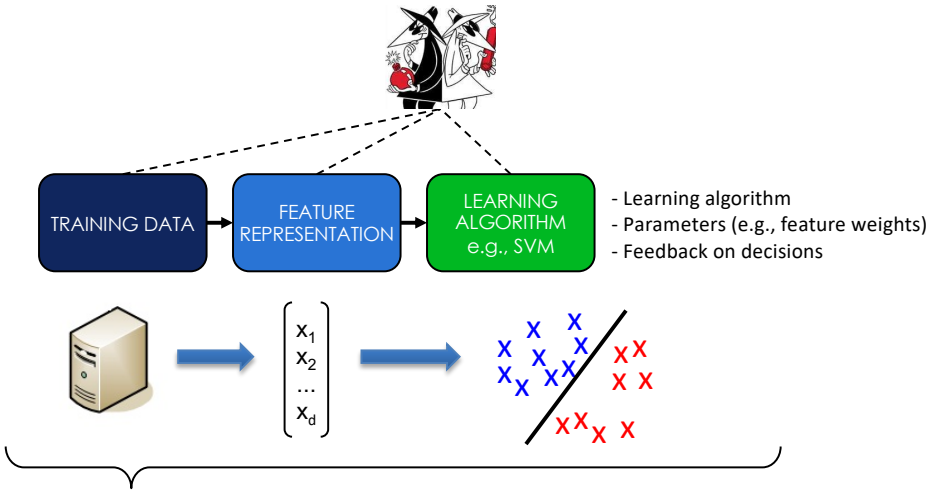
Adversary's Knowledge

Adversary's Capability

Adversary's Goal

- To cause a **security violation**…
- **Integrity:** Misclassifications that do not compromise normal system operation
- **Availability:** Misclassifications that compromise normal system operation
- **Confidentiality / Privacy:** Querying strategies that reveal confidential information on the learning model or its users

Adversary's Knowledge



- Perfect-knowledge (white-box) attacks
  - upper bound on the performance degradation under attack
- Limited-knowledge Attacks
  - Ranging from gray-box to black-box attacks

Adversary's Capability

- Attackers may manipulate training data and/or test data

TRAINING

Influence model at training time to cause subsequent errors at test time
poisoning attacks, backdoors

TEST

Manipulate malicious samples at test time to cause misclassications
evasion attacks, adversarial examples

Adversary's Capability

- Constraints on data manipulation

TRAINING

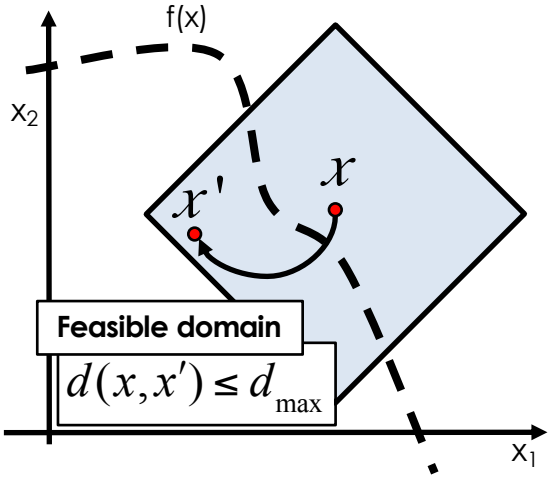
Maximum number of samples that can be added to the training data

- the attacker usually controls only a small fraction of the training samples

TEST

Maximum amount of modifications

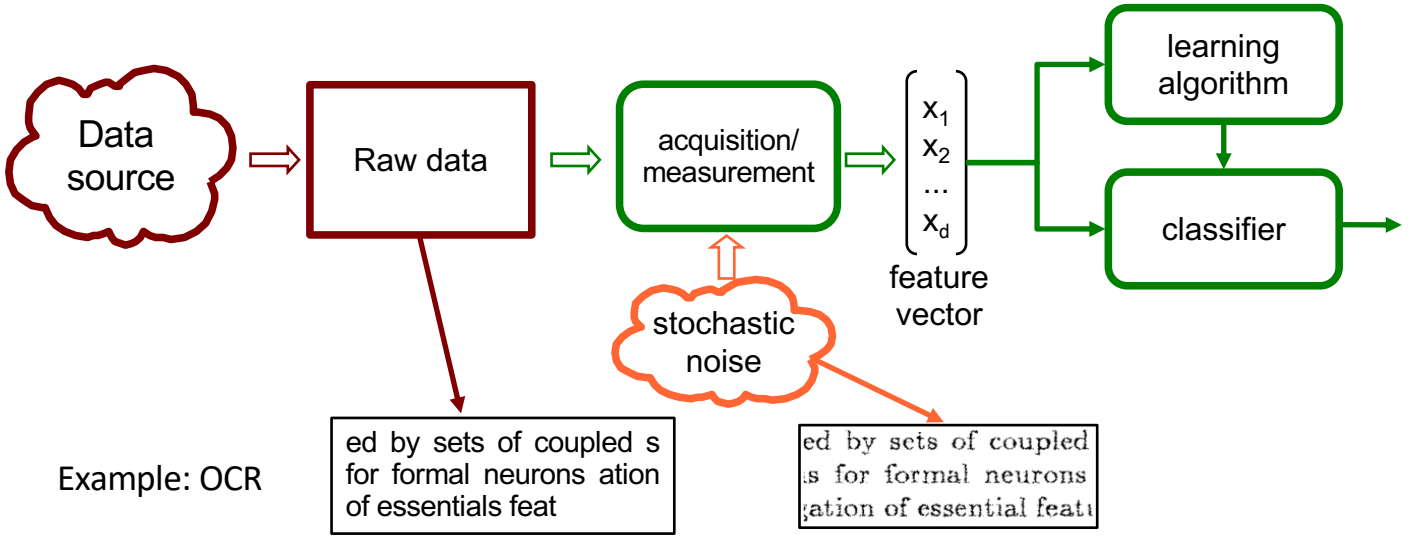
- application-specific constraints in feature space
- e.g., max. number of words that are modified in spam emails



The Classical Statistical Model

- Two implicit assumptions of the model
  - The source of data is given, and it does not depend on the classifier
  - Noise affecting data is stochastic

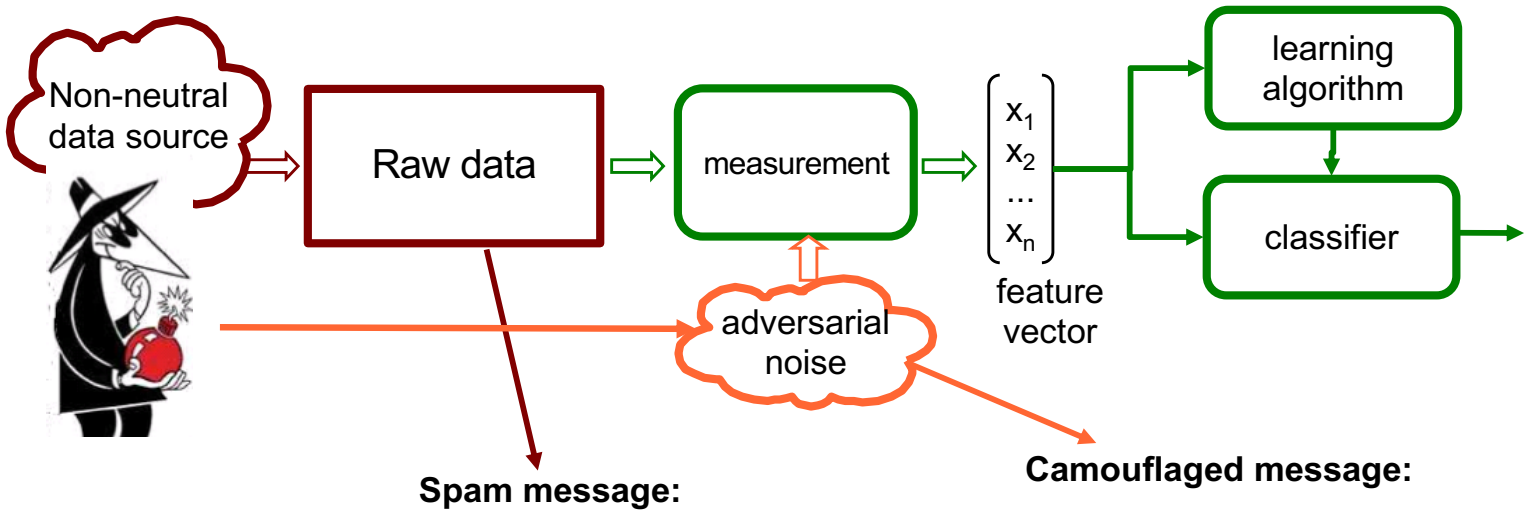
Goal of training: $\min_{\theta} loss(\theta, x, y)$



Adversarial Machine Learning

- The source of data is *not neutral*, it depends on the classifier
- Noise is not stochastic, it is *adversarial*, crafted to maximize the classification error

$$\max_{\delta} loss(\theta, x + \delta, y)$$



The Classical Model Cannot Work

- Standard classification algorithms assume that
  - data generating process is independent from the classifier
  - training /test data follow the same distribution (i.i.d. samples)
- *This is not the case for adversarial tasks!*

- Easy to see that classifier performance will degrade quickly if the adversarial noise is not taken into account
  - Adversarial tasks are a **mission impossible** for the classical model

Adversarial Noise vs. Stochastic Noise

- This distinction is not new…

Shannon's stochastic noise model: probabilistic model of the channel, the probability of occurrence of too many or too few errors is usually low

Hamming's adversarial noise model: the channel acts as an adversary that arbitrarily corrupts the code-word subject to a bound on the total number of errors

Targeted vs Non-targeted

- Non-targeted adversarial examples
  - The goal is to mislead the classifier to predict **any labels** other than the ground truth
  - Most existing work deals with this goal

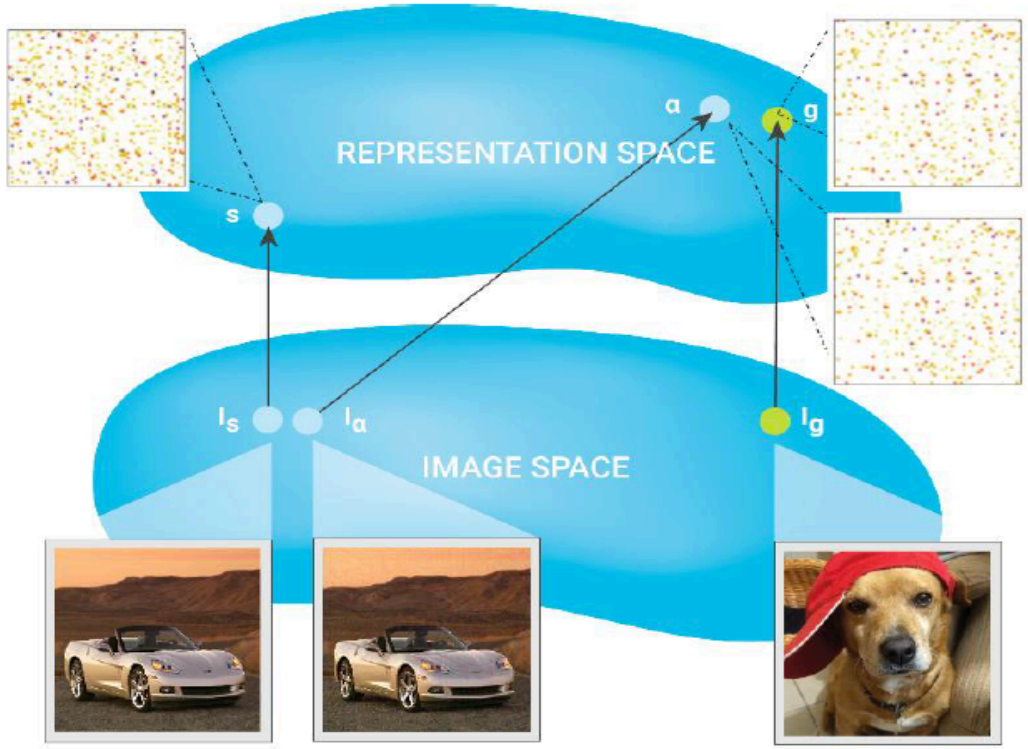
- Targeted adversarial examples
  - The goal is to mislead the classifier to predict a **target label** for an image
  - **Harder!**

Adversarial Sample

$$\mathbf{p} = \underset{\forall \mathbf{x}}{\operatorname{argmin}} \left\| f(\mathbf{x}) - f(\mathbf{t}) \right\|^2 + \beta \left\| \mathbf{x} - \mathbf{b} \right\|^2$$

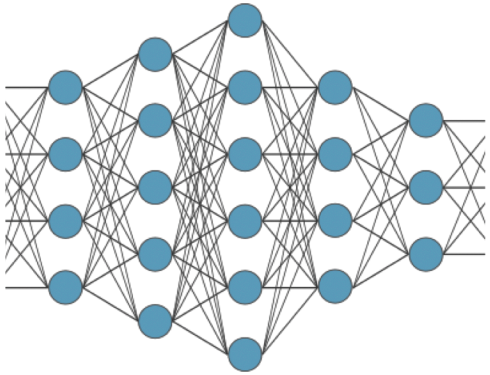
close to target in feature space

close to own class in pixel space



Where Do Adversarial Examples Come From?

- **Goal of training for DNN:**



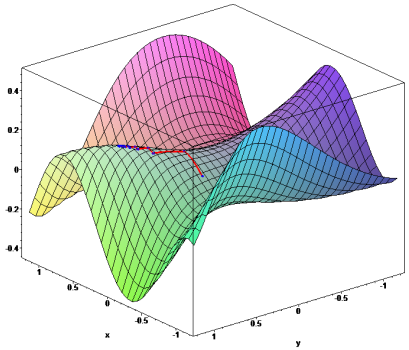
Parameters θ

$$\min_{\theta} loss(\theta, x, y)$$

Input: x Label: y

Model Parameters: θ

Optimize using gradient descent method



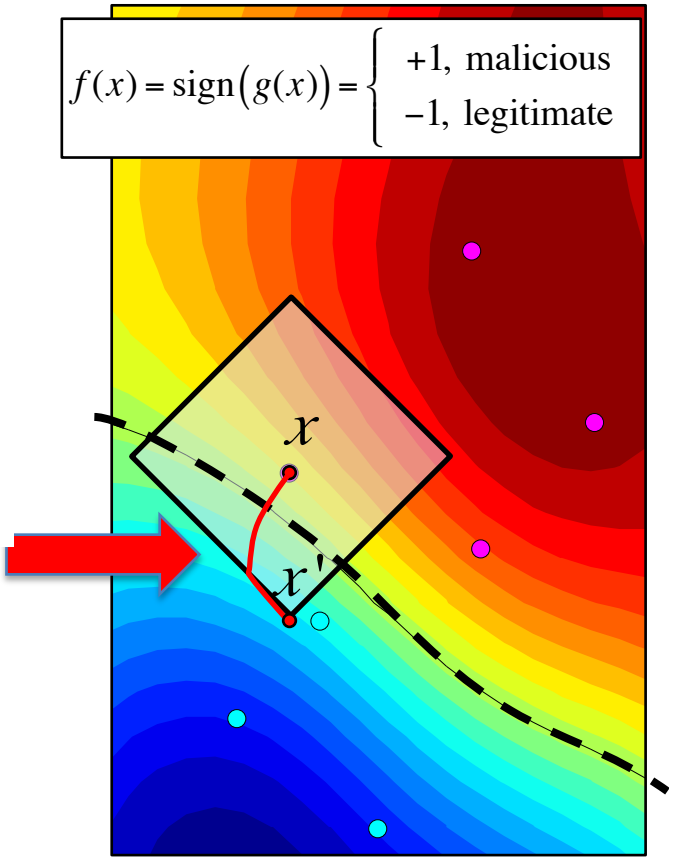
Evasion Attacks against Machine Learning at Test Time

- **Goal:** maximum-confidence *evasion*
- **Knowledge:** *perfect (white-box attack)*
- **Attack strategy:**

$$\min_{x'} g(x')$$

$$\text{s.t.} \|x - x'\|_p \leq d_{\max}$$

- Non-linear, constrained optimization



Where Do Adversarial Examples Come From?

- To get an adversarial example

$$\max_{\delta} loss(\theta, x+\delta, y)$$

$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x+\delta) \\ \text{such that} \quad & C(x+\delta) = t \\ & x+\delta \in [0,1]^n \end{aligned}$$

$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x+\delta) + c \cdot f(x+\delta) \\ \text{such that} \quad & x+\delta \in [0,1]^n \end{aligned}$$

- [Carlini, Wagner, Towards robustness of neural networks. 2017]

An Optimization Approach To Creating Physical Adversarial Examples

- Construct adversarial examples over a distribution of transformations
- Adversarial perturbations are possible in physical world under different conditions and viewpoints

$$\underset{\delta}{\operatorname{argmin}}\ \lambda||\delta||_p + J(f_\theta(x+\delta), y^*)$$

Perturbation/Noise Matrix

Lp norm (L-0, L-1, L-2, ...)

Loss Function

Adversarial Target Label

$$\underset{\delta}{\operatorname{argmin}}\ \lambda||\delta||_p + \frac{1}{k}\sum_{i=1}^{k} J(f_\theta(x_i+\delta), y^*)$$

Fast Gradient-Sign Method （FGSM）

- One of the most famous reliability attacks is FGSM
- Consider model with l_∞ norm constraint:

$$\max_{\delta} l(\mathrm{f}(x+\delta), y) \quad \text{subject to:} \quad ||\delta||_\infty \leq \epsilon$$

- Linearize the objective (loss) around (x,y):

$$l(\mathrm{f}(x+\delta), y) \approx l(f(x), y) + \nabla_x l(f(x), y)\delta$$

- The optimal solution is to maximize distortion along each coordinate

$$\delta^* = \epsilon \,\mathrm{sgn}\{\nabla_x l(f(x), y)\}$$

Iterative Gradient-Sign Attack

- FGSM is only a simple linear approximation
- A much stronger attack is to use *projected gradient descent*, where we iteratively use a linear approximation.
- Suppose that x_t represents an attack input in iteration t.
- In each iteration, we compute the next iterate as follows:

$$x_{t+1} = \text{Proj}_{\epsilon}[x_t + \beta \, \text{sgn}\{\nabla_x l(g(x_t), y)\}]$$

- *The projection step ensures that*
  - $||x_{t+1} - x||_{\infty} \leq \epsilon$
  - *the solution is a valid pixel, usually normalized in [0,1]*

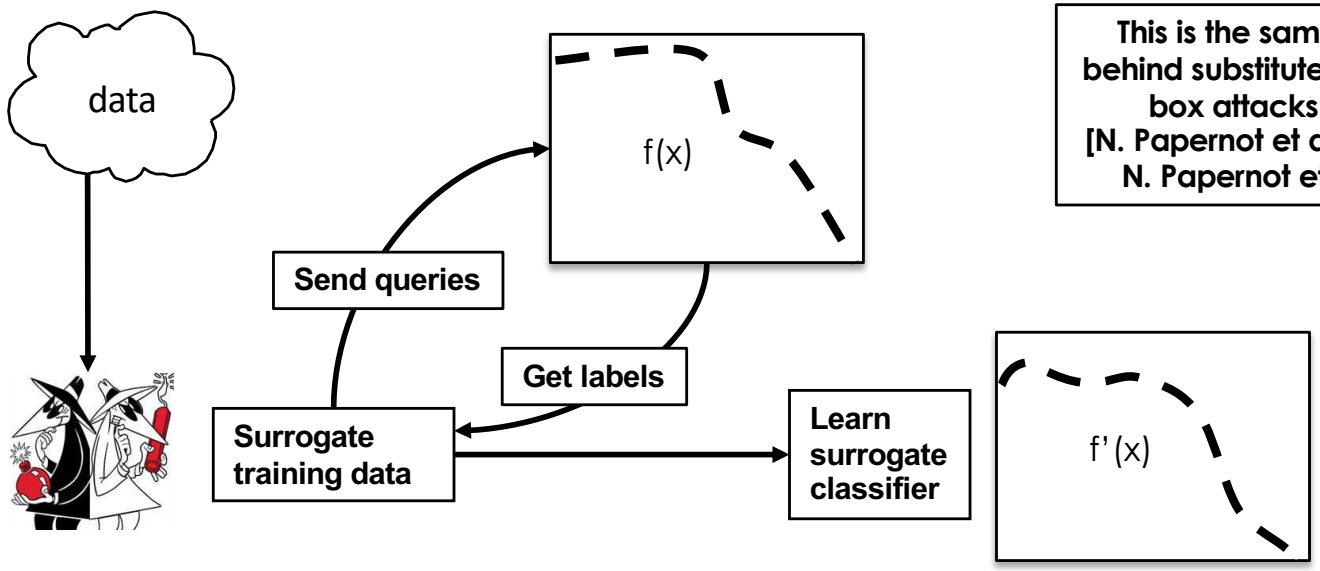
A General Framework for Black-box attacks

- Zero-Query Attack： Transferability-based attack
  - Practical Black-Box Attacks against Machine Learning
  - Ensemble transferability-based attack

- Query Based Attack
  - Finite difference gradient estimation
  - Query reduced gradient estimation
  - Results:similar effectiveness to white-box attack
  - A general active query game model

Bounding the Adversary's Knowledge

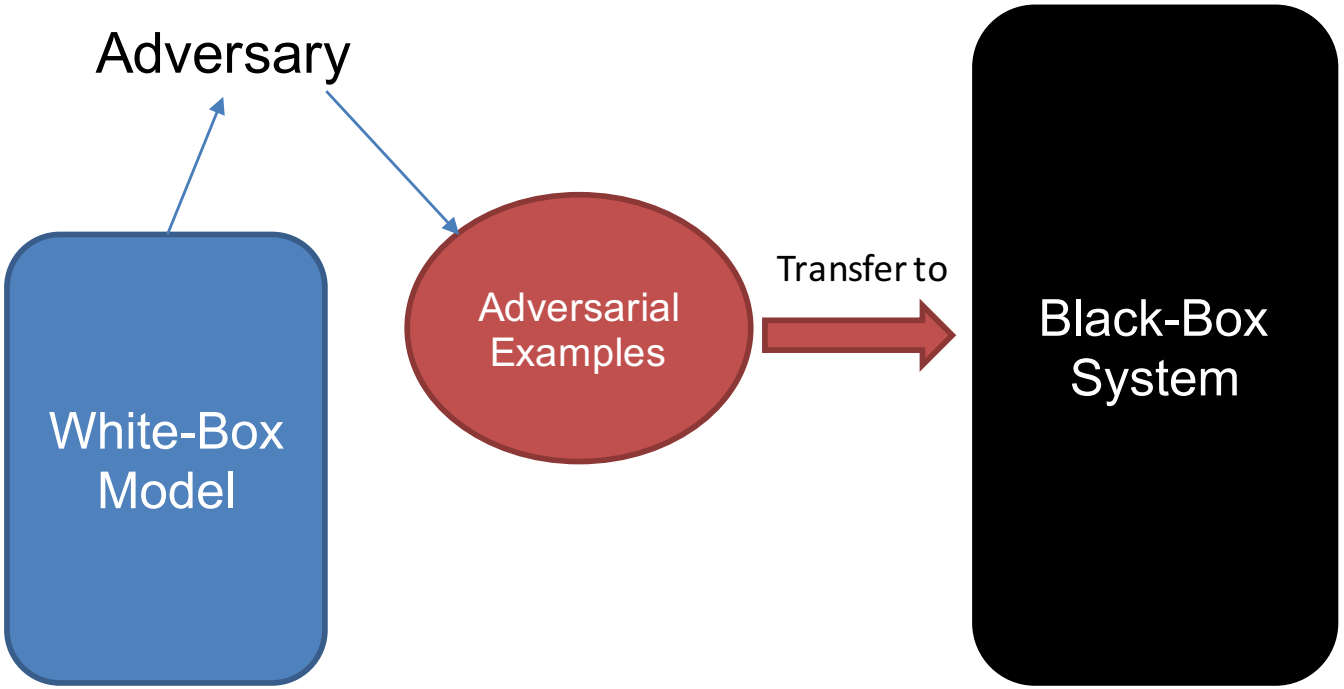
- **Limited-knowledge (gray/black-box) attacks**
  - Only partial feature representation and (possibly) learning algorithm are known



This is the same underlying idea behind substitute models and black-box attacks (*transferability*) [N. Papernot et al., IEEE Euro S&P '16; N. Papernot et al., ASIACCS'17]

Black-box Attacks Based On Transferability

- An adversarial example to remain effective even for the models other than the one used to generate it.



Liu, Chen, Liu, Song. Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

Transferability

- Targeted adversarial example's transferability among two models is poor!

	ResNet152	ResNet101	ResNet50	VGG16	GoogLeNet	Incept-v3
ResNet152	100%	2%	1%	1%	1%	0%
ResNet101	3%	100%	3%	2%	1%	1%
ResNet50	4%	2%	100%	1%	1%	0%
VGG16	2%	1%	2%	100%	1%	0%
GoogLeNet	1%	1%	0%	1%	100%	0%
Incept-v3	0%	0%	0%	0%	0%	100%

Only 2% of the adversarial images generated for VGG16 (row) can be predicted as the targeted label by ResNet50 (column)

Query Based Attack

- The zero-query attack can be viewed as a special case for the query based attack, where the number of queries made is zero
- Finite difference gradient estimation
- Query reduced gradient estimation

Query Based attacks

- Finite difference gradient estimation
  - Given d-dimensional vector x, we can make $2d$ queries to estimate the gradient as below

$$\mathrm{FD}_{\mathbf{x}}(g(\mathbf{x}), \delta) = \begin{bmatrix} \frac{g(\mathbf{x}+\delta\mathbf{e}_1)-g(\mathbf{x}-\delta\mathbf{e}_1)}{2\delta} \\ \vdots \\ \frac{g(\mathbf{x}+\delta\mathbf{e}_d)-g(\mathbf{x}-\delta\mathbf{e}_d)}{2\delta} \end{bmatrix}$$

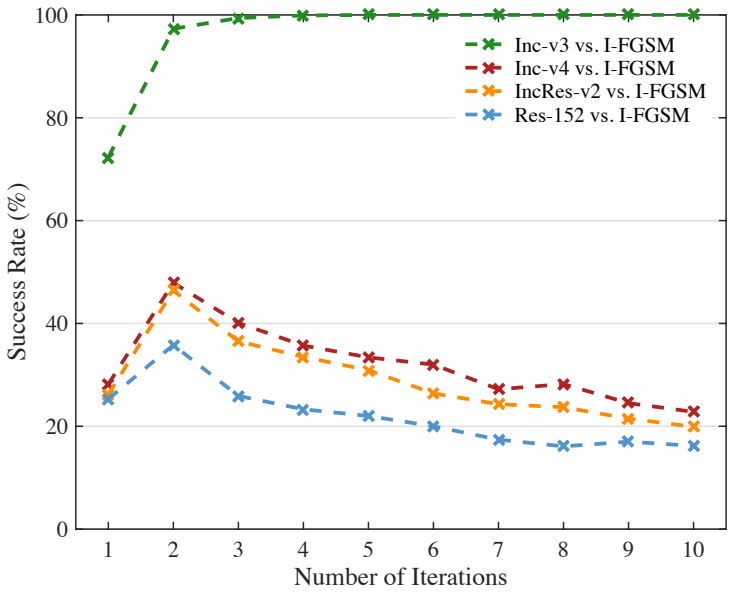
  - An example of approximate FGS with finite difference

$$x_{adv} = \mathbf{x} + \epsilon \cdot \mathrm{sign}\left(\mathrm{FD}_{\mathbf{x}}(\ell_f(\mathbf{x}, y), \delta)\right)$$

- Query reduced gradient estimation
  - Random grouping
  - PCA

Limitations of Black-box Attacks (1)

- One-step methods (FGSM) have poor white-box attack ability;
- Iterative methods have poor transferability;
- Trade-off between transferability and attack ability, makes black-box attacks less effective.



Limitations of Black-box Attacks (2)

- Train a substitute network to fully characterize the behavior of the black-box model
  - Require full prediction confidence;
  - Require tremendous queries;
- Hard to deploy for models trained on large-scale dataset
- Impossible for cases without querying

- Our solution: alleviate the trade-off between transferability and attack ability.

Momentum Iterative FGSM

$$x_0^* = x, \quad x_{t+1}^* = \mathrm{clip}\left(x_t^* + \alpha \cdot \mathrm{sign}\left(\nabla_x L(x_t^*, y)\right)\right)$$

Momentum

$$x_0^* = x, g_0 = 0$$

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x L(x_t^*, y)}{\|\nabla_x L(x_t^*, y)\|_1}$$

$$x_{t+1}^* = \mathrm{clip}(x_t^* + \alpha \cdot \mathrm{sign}(g_{t+1}))$$

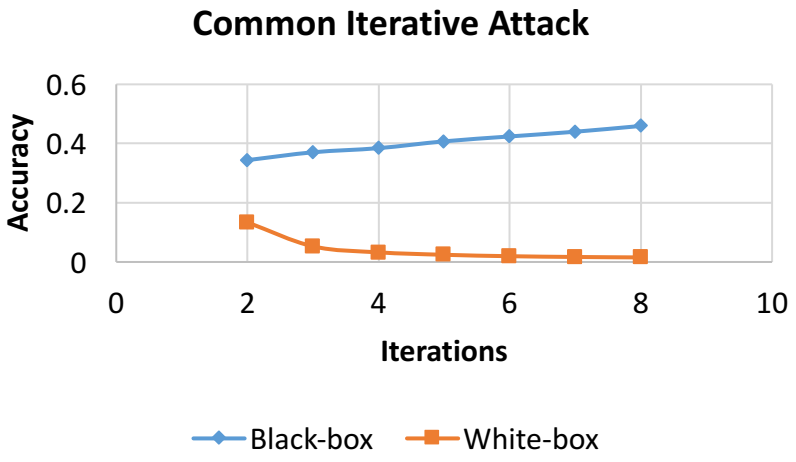
Optimization with Momentum

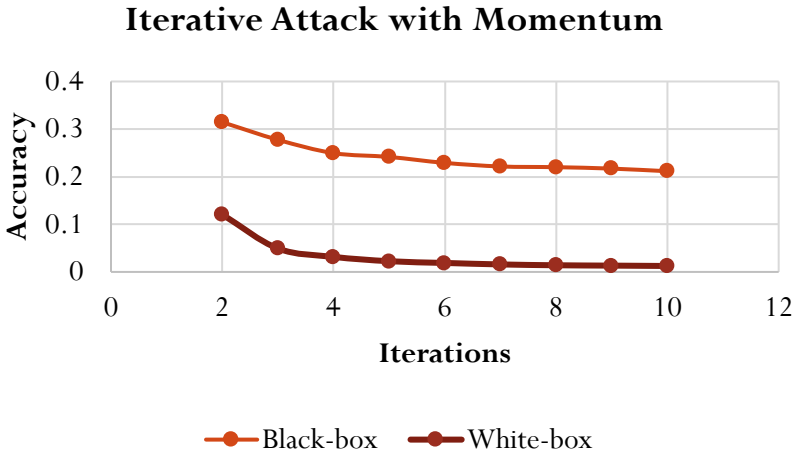
- Accelerate gradient descent;
- Escape from poor local minima and maxima;
- Stabilize update directions of stochastic gradient descent;
- Momentum can be used for adversarial attacks
  - It is still a white-box attack method but has strong black-box attack ability (transferability)

Take home message

The common iterative attack:
More iteration → Less transferability

The iterative attack with momentum:
More iteration → More transferability





Results in NIPS

- Non-targeted

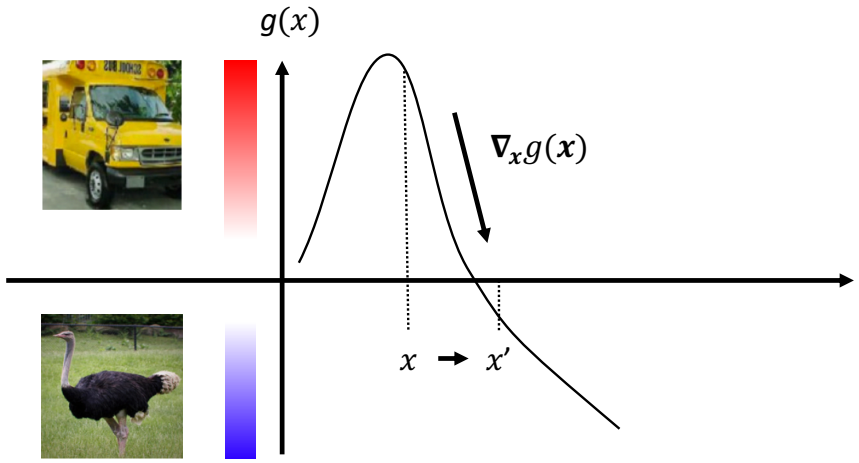
Team Name	Score
TSAIL	0.78164
Sangxia	0.77685
Stanford & Sun	0.77402
iwiwi	0.76898

- Targeted

Team Name	Score
TSAIL	0.40221
Sangxia	0.36877
FatFingers	0.36802
Anil Thomas	0.36455

Why Is Machine Learning So Vulnerable?

- Learning algorithms tend to overemphasize some features to discriminate among classes
- Large sensitivity to changes of such input features:



- Different classifiers tend to find the same set of **relevant features**
  - that is why attacks can *transfer* across models!

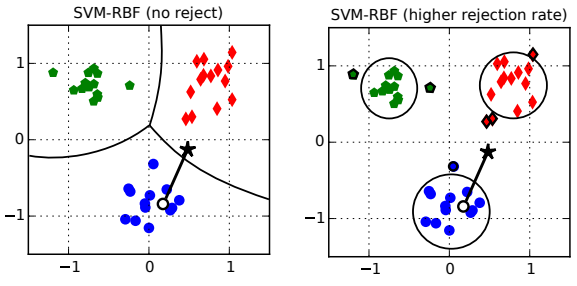
Security Measures against Evasion Attacks

- **Defences**
  - **Model hardening**: classify correctly in spite of the noise.
  - **Input preprocessing**: mask the adversarial noise

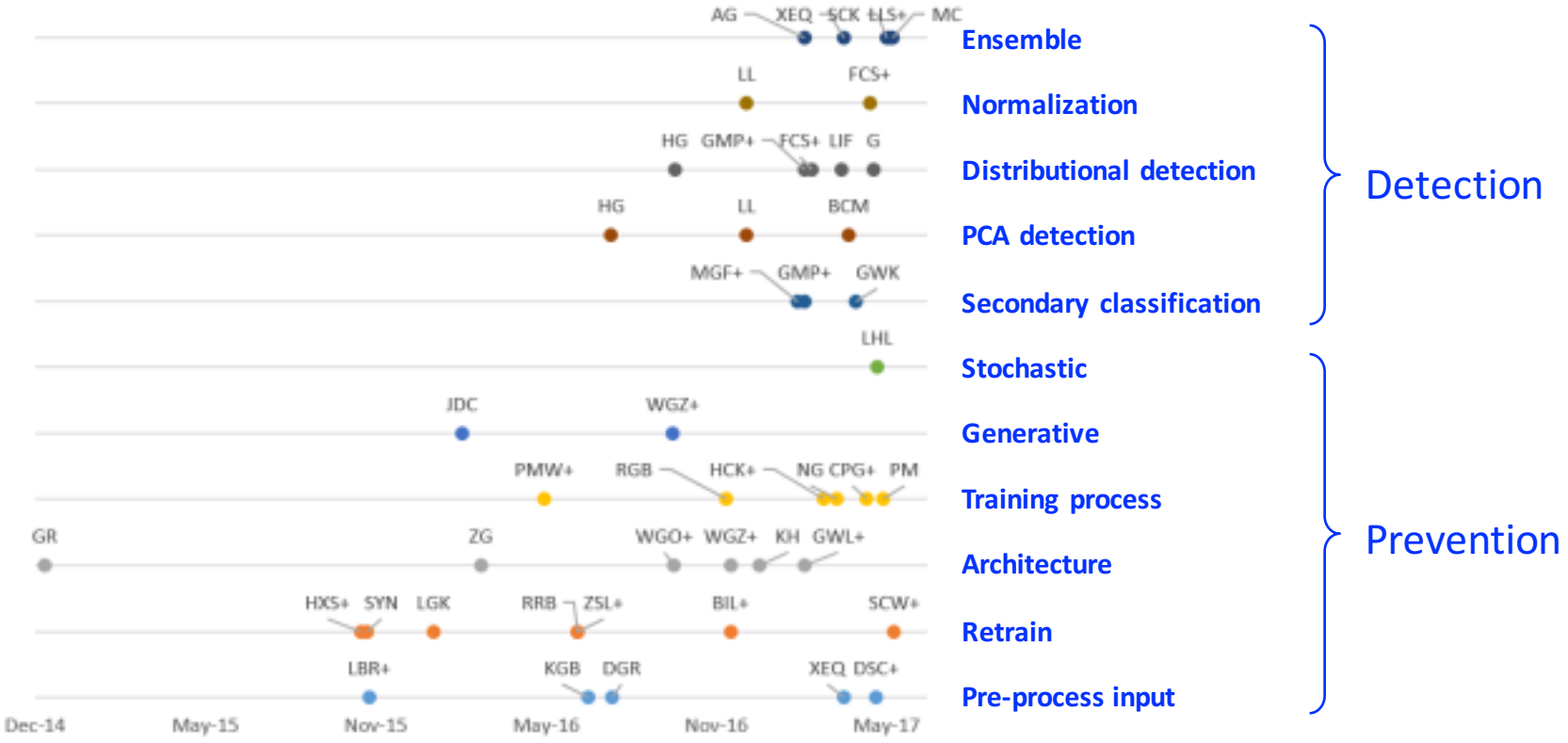
$$\min_{w} \sum_i \max_{||\delta_i|| \le \epsilon} \ell(y_i, f_w(x_i + \delta_i))$$

bounded perturbation!

- **Rejection / detection**
  - Reject adversarial samples without classifying them based on a specialized side model.
  -



Numerous Defenses Proposed



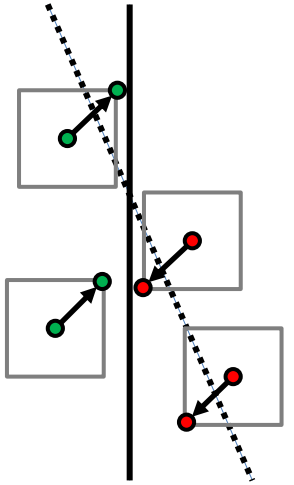
The big picture：robust optimization

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

Part II: training a robust classifier

$$\min_{\theta} \sum_{x,y \in S} \max_{\delta \in \Delta} \text{Loss}\left(x+\delta, y; \theta\right)$$

Part I: creating an adversarial example (or ensuring one does not exist)

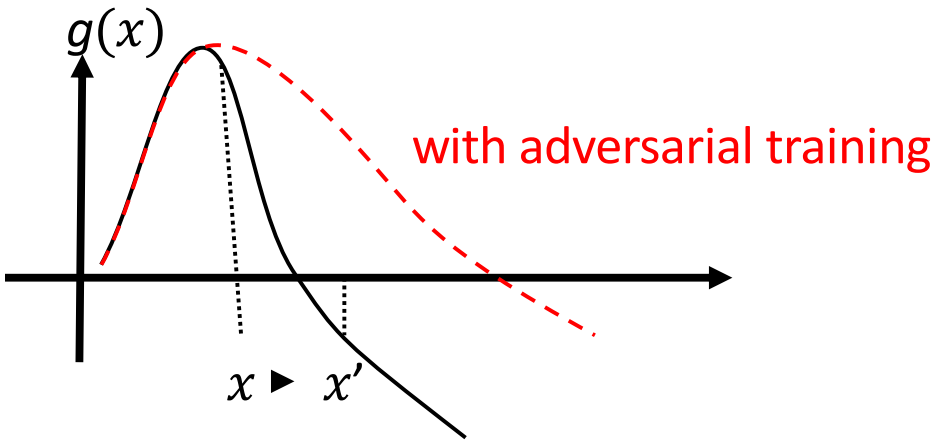


Adversarial Training

- Starting with pre-trained model or from scratch
- Fraction of adversarial samples per mini-batch
- Replace original samples with adversarial ones, or keep both
- Pre-computing of adversarial samples vs computing them on-the-fly on the current version of the trained model
- Types of attacks to be used.

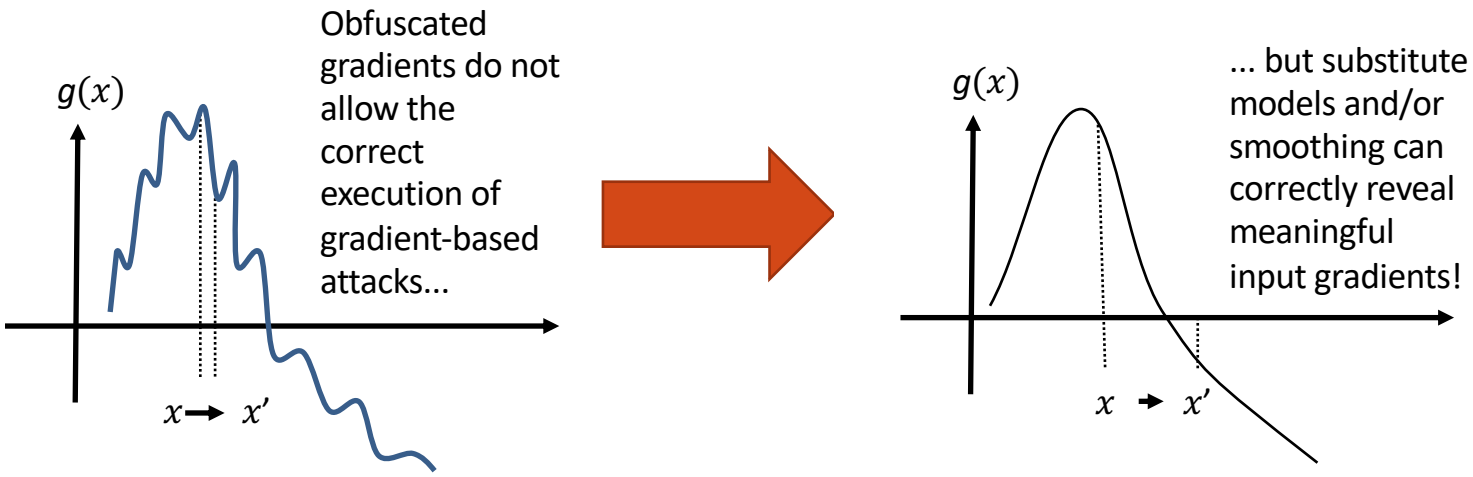
Adversarial Training and Regularization

- Adversarial training can also be seen as a form of regularization, which penalizes the (dual) norm of the input gradients
- The net effect of these techniques is to make the prediction function of the classifier smoother

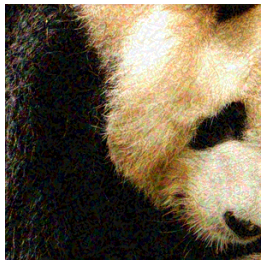


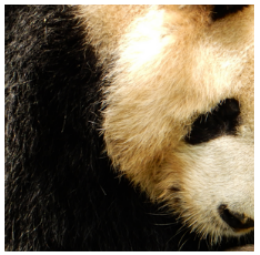
Ineffective Defenses: Obfuscated Gradients

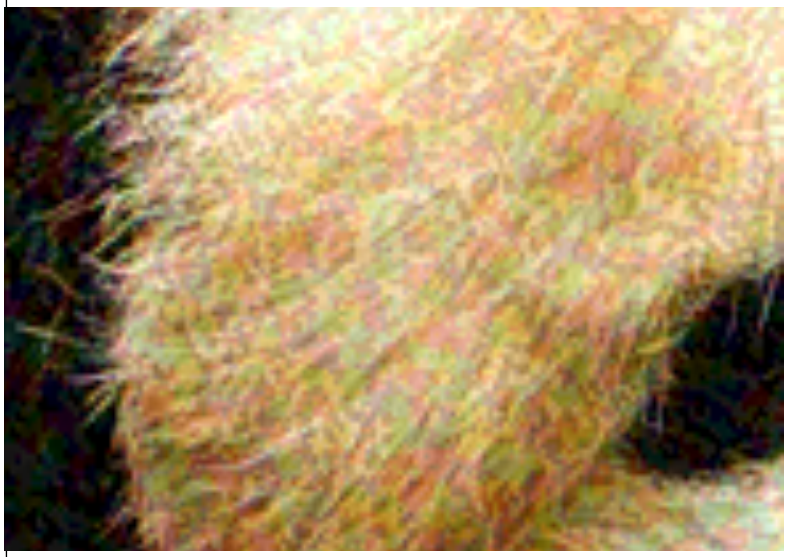
- Adversarial training may not result in models that are truly more **robust**, but that have **obfuscated gradients** which only hinder (standard) gradient attacks [Tramèr et al., 2018]
- Work by Carlini & Wagner (SP' 17) has shown that
  - recent defenses rely on obfuscated/masked gradients
  - they can be circumvented



Undistinguishable?





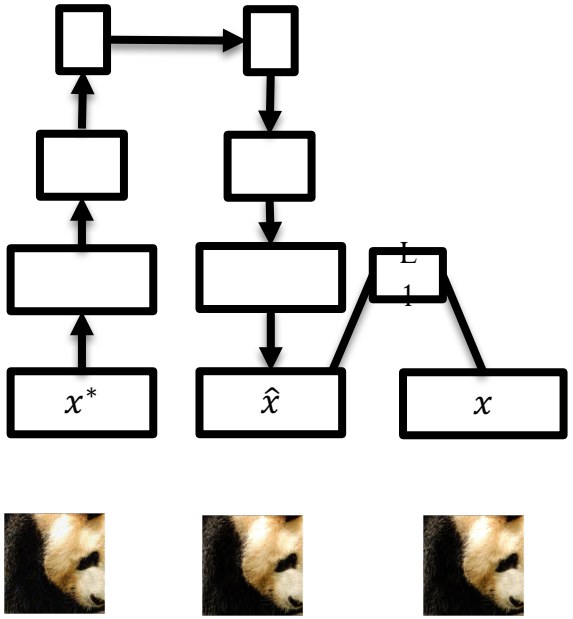


Basic idea

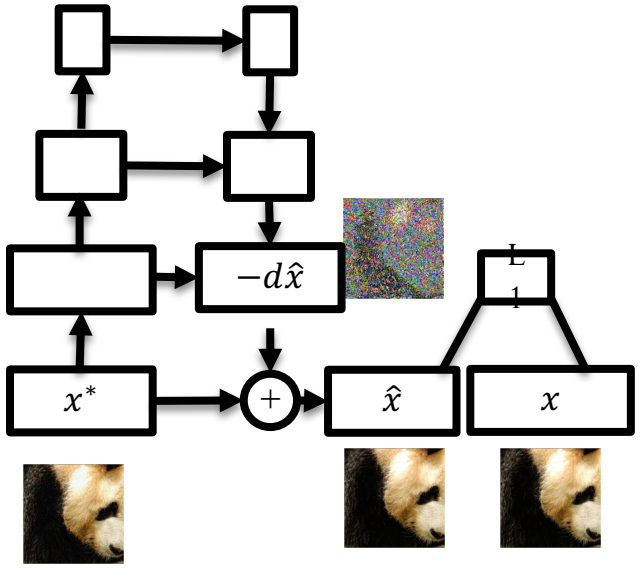
- Denoise the image
  - $X + dX \rightarrow Y'$
  - $X + dX \rightarrow \hat{X} \rightarrow Y$
- Denoise method
  - Traditional denoiser (median filter, BM3D)
  - Denoise by neural network

Denoising network structures

Denoising Autoencoder



Denoising Additive U-Net



Training examples

- Source image:
  - 30000 images
- Attacks

Table 1: Adversarial images generated by different models for training and testing.

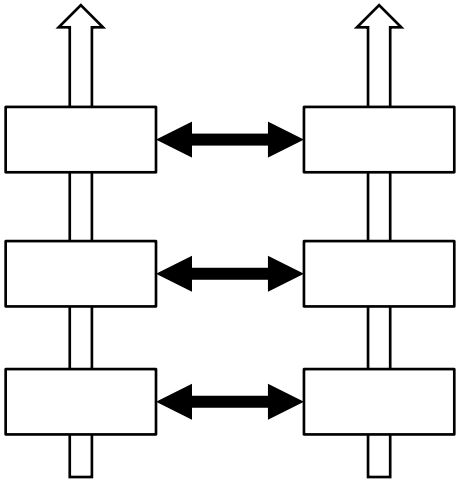
	Attacking method	Attacked model	ϵ
TrainSet and ValSet	FGSM	IncV3	[1,16]
	FGSM	IncResV2	
	FGSM	Res	
	FGSM	IncV3/IncResV2/Res	
	IFGSM2	IncV3/IncResV2/Res	
	IFGSM4	IncV3/IncResV2/Res	
	IFGSM8	IncV3/IncResV2/Res	
WhiteTestSet	FGSM	IncV3	{4,16}
	IFGSM4	IncV3/IncResV2/Res	
BlackTestSet	FGSM	IncV4	{4,16}
	IFGSM4	IncV4	

Denoise effect of different methods

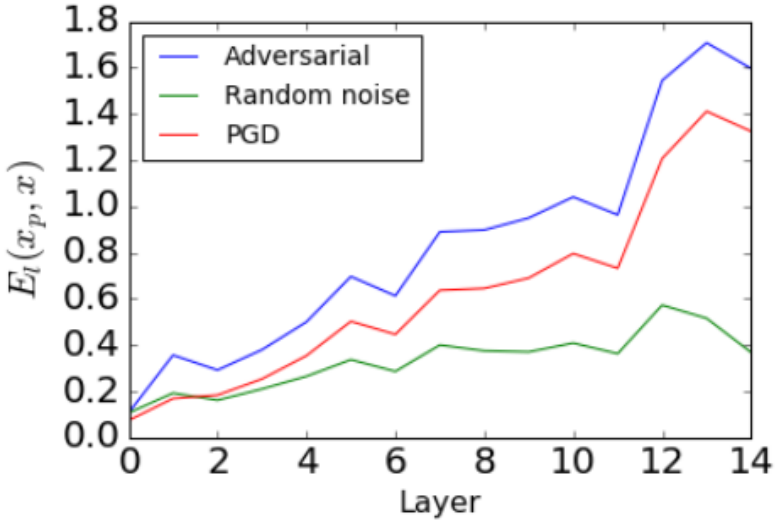
Defense	Clean	WhiteTestSet		BlackTestSet	
		$\epsilon = 4$	$\epsilon = 16$	$\epsilon = 4$	$\epsilon = 16$
NA	**0.0000**	0.0177	0.0437	0.0176	0.0451
DAE	0.0360	0.0359	0.0360	0.0360	0.0369
DUNET	0.0150	**0.0140**	**0.0164**	**0.0140**	**0.0181**
NA	**76.7%**	14.5%	14.4%	61.2%	41.0%
DAE	58.3%	**51.4%**	**36.7%**	55.9%	48.8%
DUNET	75.3%	20.0%	13.8%	**67.5%**	**55.7%**

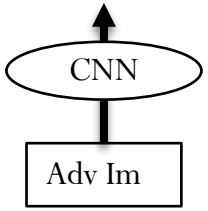
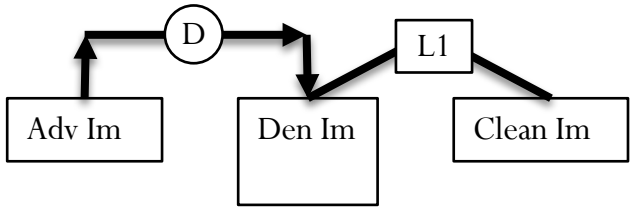
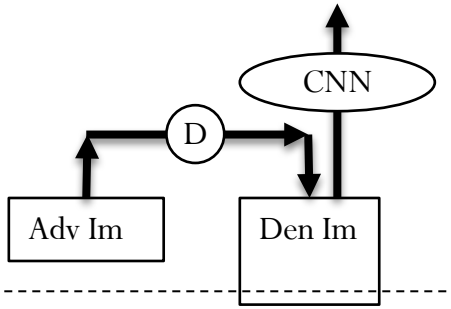
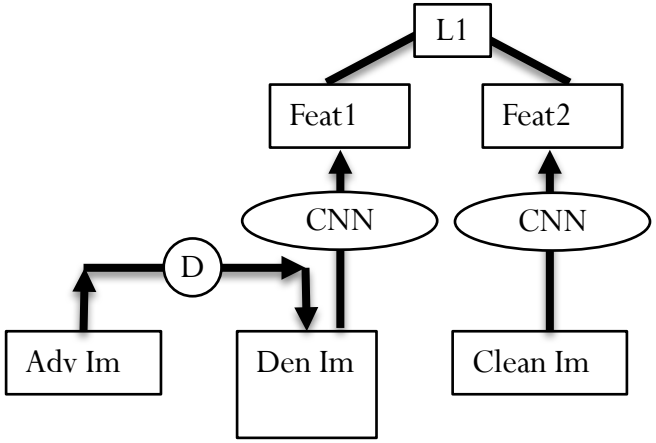
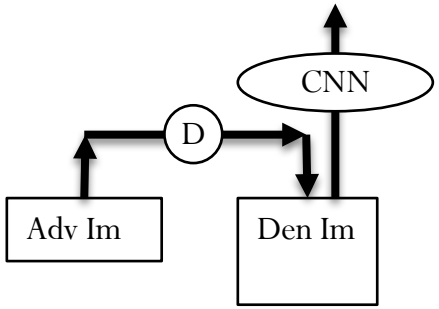
- Unexpected:
  - Convolutional Autoencoder is very poor at reconstructing large images
  - Although most noise is removed, the accuracy doesn't increase

Error amplification effect

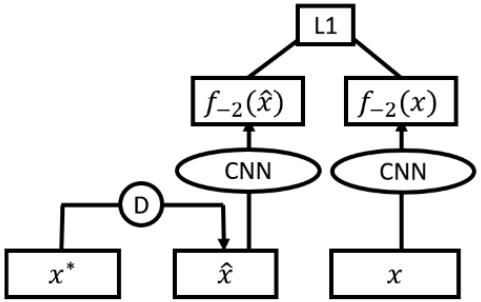


$$E_l(x_p, x) = |f_l(x_p) - f_l(x)|/|f_l(x)|.$$

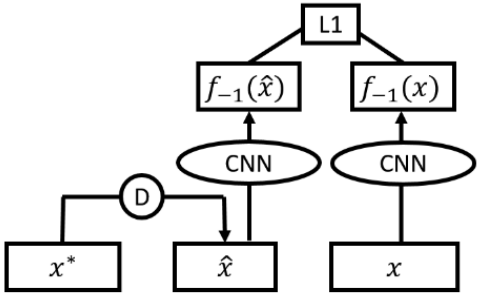


	Train	Test
Naive		
Pixel Guided Denoiser (PGD)		
High-level Representation Guided Denoiser (HGD)		

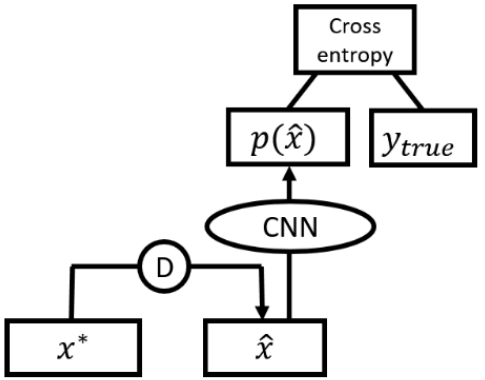
Variation of HGD



Feature guided denoiser
FGD



Logits guided denoiser
LGD



Class label guided denoiser
CGD

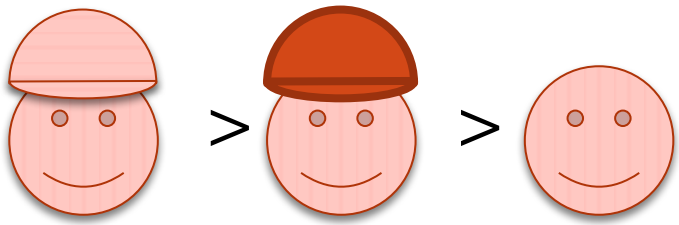
Transferability of HGD

- Train 750 classes, test other 250 classes

Defense	Clean	WhiteTestSet		BlackTestSet	
		$\epsilon = 4$	$\epsilon = 16$	$\epsilon = 4$	$\epsilon = 16$
NA	**76.6%**	14.5%	14.4%	61.2%	41.0%
LGD	76.3%	**73.9%**	**65.7%**	**74.8%**	**72.2%**

- Borrow HGD from other model

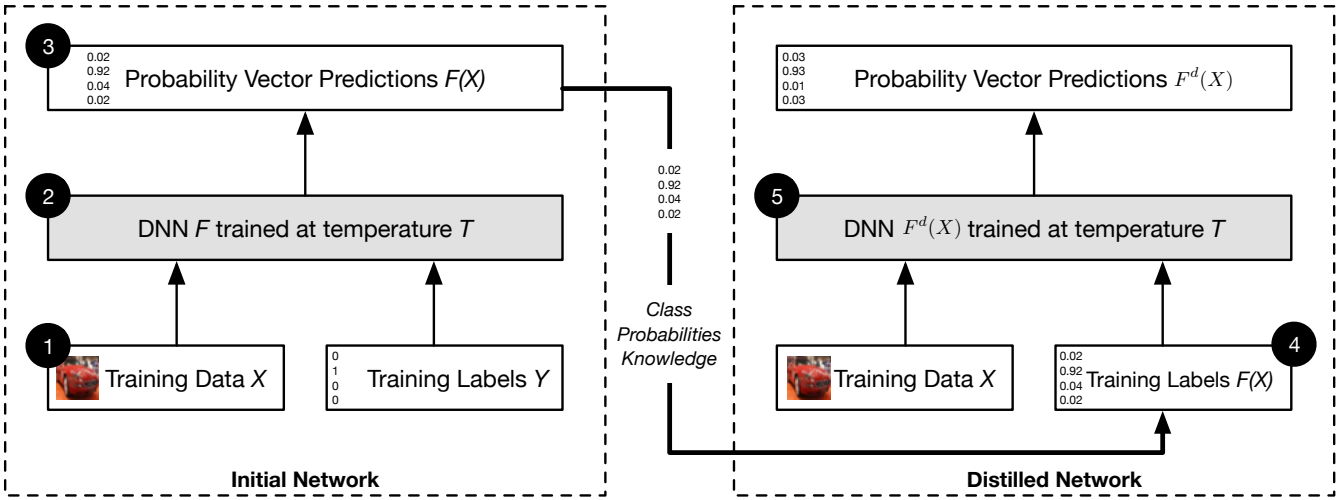
Denoiser for Resnet	Clean	WhiteTestSet		BlackTestSet	
		$\epsilon = 4$	$\epsilon = 16$	$\epsilon = 4$	$\epsilon = 16$
NA	**78.5%**	63.3%	38.4%	67.8%	48.6%
IncV3 guided LGD	77.4%	75.8%	71.7%	76.1%	72.7%
Resnet guided LGD	78.4%	**76.1%**	**72.9%**	**76.5%**	**74.6%**



Model Hardening

- **Defensive distillation** [Papernot et al., 2015b]
  - Train a teacher model on the original data
  - Train a student model based on the predictions of the teacher
  - Minimize the cross entropy

$$\mathrm{H}\left(F^d(X), F(X)\right) = \mathrm{H}(F(X)) + \mathrm{KL}\left(F(X) \parallel F^d(X)\right)$$

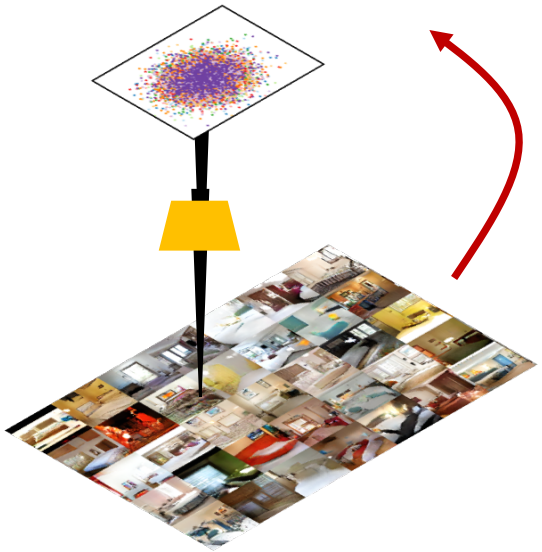


Learn a Robust Deep Network

(Pang, Du, Zhu, ICML 2018)

LDA is more efficient than LR

- Efron et al. (1975) show that if data distributes as a mixture of Gaussian (MoG), then linear discriminant analysis (LDA) is **more efficient** than logistic regression.
- However, practical data hardly distributes as a mixture of Gaussian.



Learn a deep network to map the data into a simple MoG distribution

Definition of Robustness

- The robustness on a point with label i (Moosavi-Dezfoolo et al. , CVPR 2016):

$$\min_{j \neq i} d_{i,j} ,$$

where $d_{i,j}$ is the minimal distance of a point with label i to an adversarial example with label j.

- We further define the robustness of the classifier as:

$$\mathbf{RB} = \min_{i,j \in [L]} \mathbb{E}(d_{i,j}) .$$

Robustness w.r.t Gaussian parameters

- We prove that

$$\mathbb{E}(d_{i,j}) = \sqrt{\frac{2}{\pi}}\exp\left(-\frac{\Delta_{i,j}^2}{8}\right) + \frac{1}{2}\Delta_{i,j}\left[1 - 2\Phi(-\frac{\Delta_{i,j}}{2})\right]$$

$$\mathbf{RB} \approx \overline{\mathbf{RB}} = \frac{1}{2}\min_{i,j\in[L]} \Delta_{i,j},$$

where $\Delta_{i,j}$ is the **Mahalanobis distance** between two Gaussian components of label i and j in the mixture of Gaussian.

A MoG distribution with maximal M-distance can maximize $\overline{\mathbf{RB}}$.

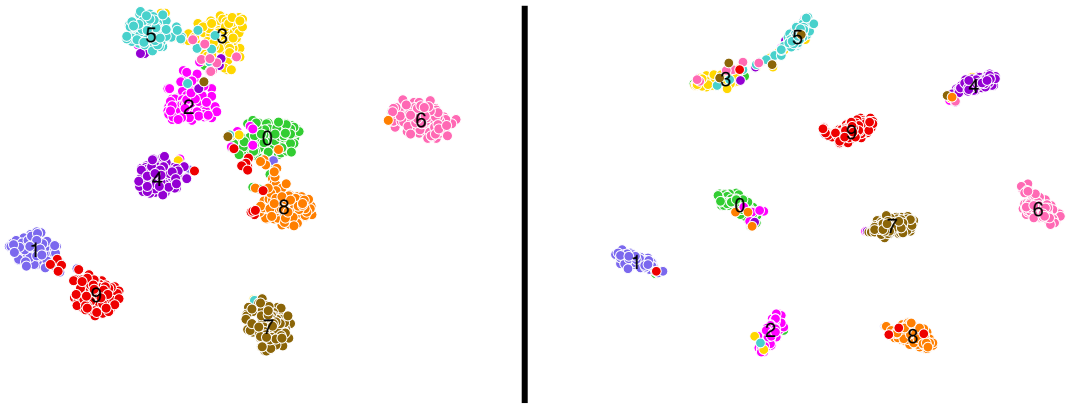
MM-LDA Networks

- MM-LDA networks first map x to a latent feature z via DNN; then applies MM-LDA on z
  - We have $z = f(x; \theta)$
  - The classification model is

$$P(y=k|z) = \frac{P(z|y=k)P(y=k)}{P(z)} = \frac{\pi_k \mathcal{N}(z|\mu_k^*, \mathrm{I})}{\sum_{i=1}^{L} \pi_i \mathcal{N}(z|\mu_i^*, \mathrm{I})}.$$

- Then, we can train MM-LDA network as usual using SGD to minimize cross-entropy loss
  - **No extra cost, as compared to vanilla DNN**

More orderly distribution in the feature space



SR networks **MM-LDA networks**

t-SNE visualization on the test set of CIFAR-10

Better robustness on iterative-based attacks

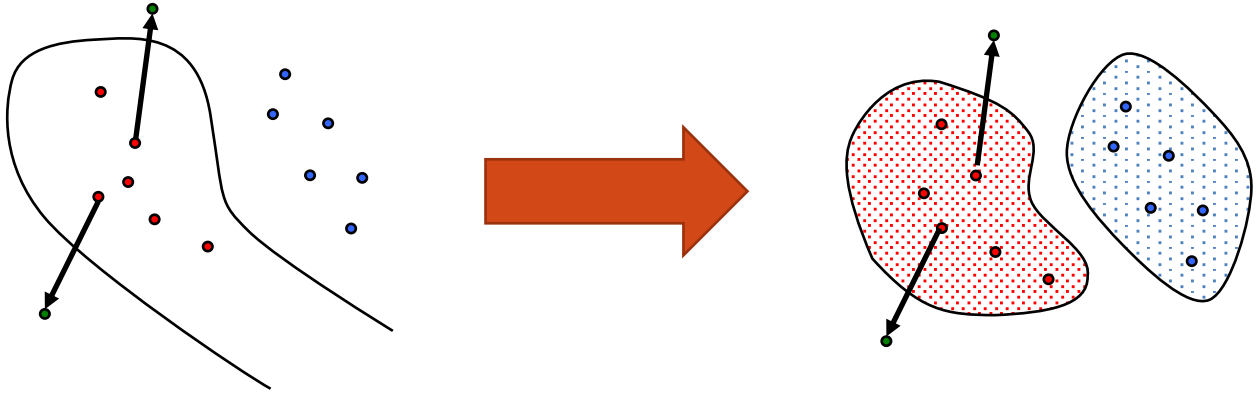
Table 1. Classification accuracy (%) on adversarial examples of MNIST and CIFAR-10. The investigated values of perturbation are 0.04, 0.12, and 0.20. **Boldface** indicates the best result under certain combination of a value of perturbation and an attacking method.

Perturbation	Model	MNIST				CIFAR-10			
		FGSM	BIM	ILCM	JSMA	FGSM	BIM	ILCM	JSMA
0.04	Resnet-32 (SR)	93.6	87.9	94.8	92.9	20.0	5.5	0.2	65.6
	Resnet-32 (SR) + SAT	86.7	68.5	98.4	-	24.4	7.0	0.4	-
	Resnet-32 (SR) + HAT	88.7	96.3	**99.8**	-	30.3	5.3	1.3	-
	Resnet-32 (MM-LDA)	**99.2**	**99.2**	99.0	**99.1**	**91.3**	**91.2**	**70.0**	**91.2**
0.12	Resnet-32 (SR)	28.1	3.4	20.9	56.0	10.2	4.1	0.3	20.5
	Resnet-32 (SR) + SAT	40.5	8.7	88.8	-	88.2	6.9	0.1	-
	Resnet-32 (SR) + HAT	40.3	40.1	92.6	-	44.1	8.7	0.0	-
	Resnet-32 (MM-LDA)	**99.3**	**98.6**	**99.6**	**99.7**	**90.7**	**90.1**	**42.5**	**91.1**
0.20	Resnet-32 (SR)	15.5	0.3	1.7	25.6	10.7	4.2	0.6	11.5
	Resnet-32 (SR) + SAT	17.3	1.1	69.4	-	**91.7**	9.4	0.0	-
	Resnet-32 (SR) + HAT	10.1	10.5	46.1	-	40.7	6.0	0.2	-
	Resnet-32 (MM-LDA)	**97.5**	**97.3**	**96.6**	**99.6**	89.5	**89.7**	**31.2**	**91.8**

MM-LDA networks can significantly improve accuracy compared to adversarial training (fine-tune) baselines, even with much less computational cost.

Detecting & Rejecting Adversarial Examples

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

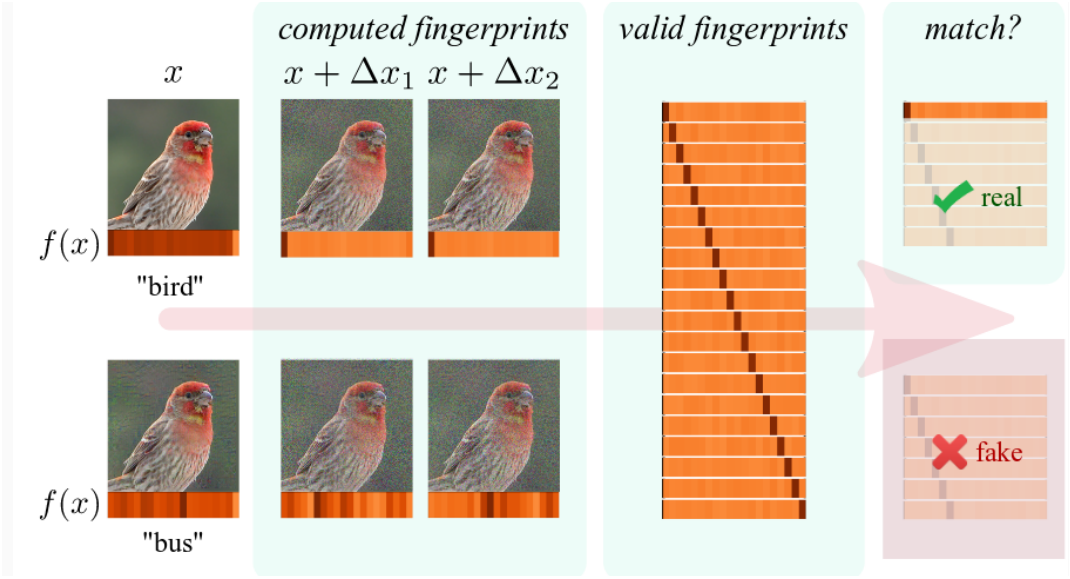


blind-spot evasion
(not even required to mimic the target class)

rejection of adversarial examples through enclosing of legitimate classes

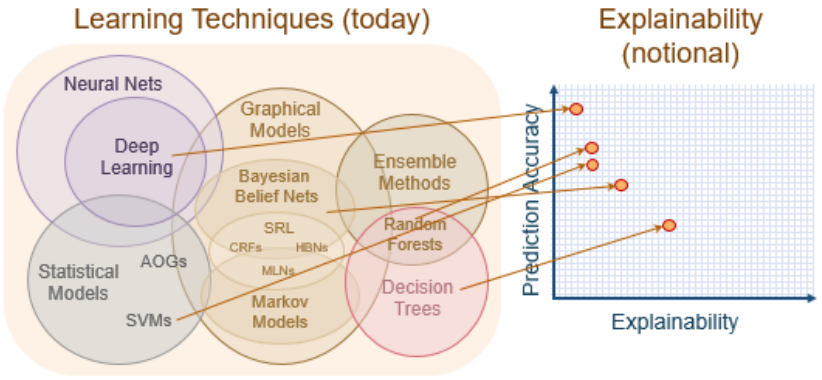
Neural fingerprinting [Dathathri et al.]

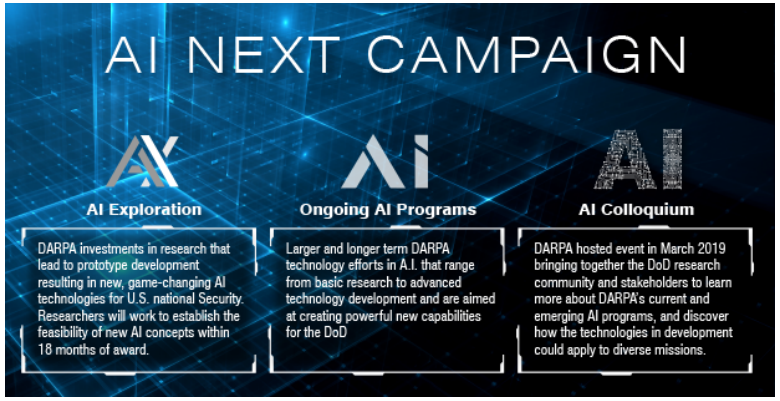
- Encode *secret* fingerprint patterns into the behavior of a neural network around the input data.
- This pattern characterizes the net- work's expected behavior around real data and can thus be used to reject fake data
- Fingerprints are a fixed set of Δx *and* test all fingerprints at prediction time to detect attack



Explainable Artificial Intelligence (XAI)

- Develop algorithms that facilitate users to understand the rationale for the system's decisions
  - Provide an explanation of individual decisions
  - Understand the system's overall strengths and weaknesses
  - Predict how the system will behave in the future
  - Correct the system's mistakes

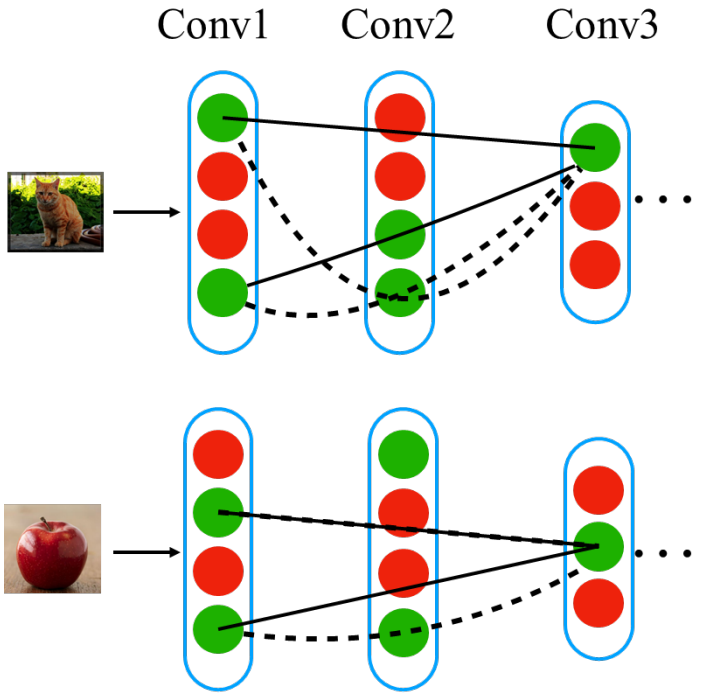




Critical Data Routing Paths

Idea: identifying the critical data routing paths (CDRPs) in the network for each given input

- Critical Nodes: important channels of layers' output
  - if they were set to 0, the performance would deteriorate

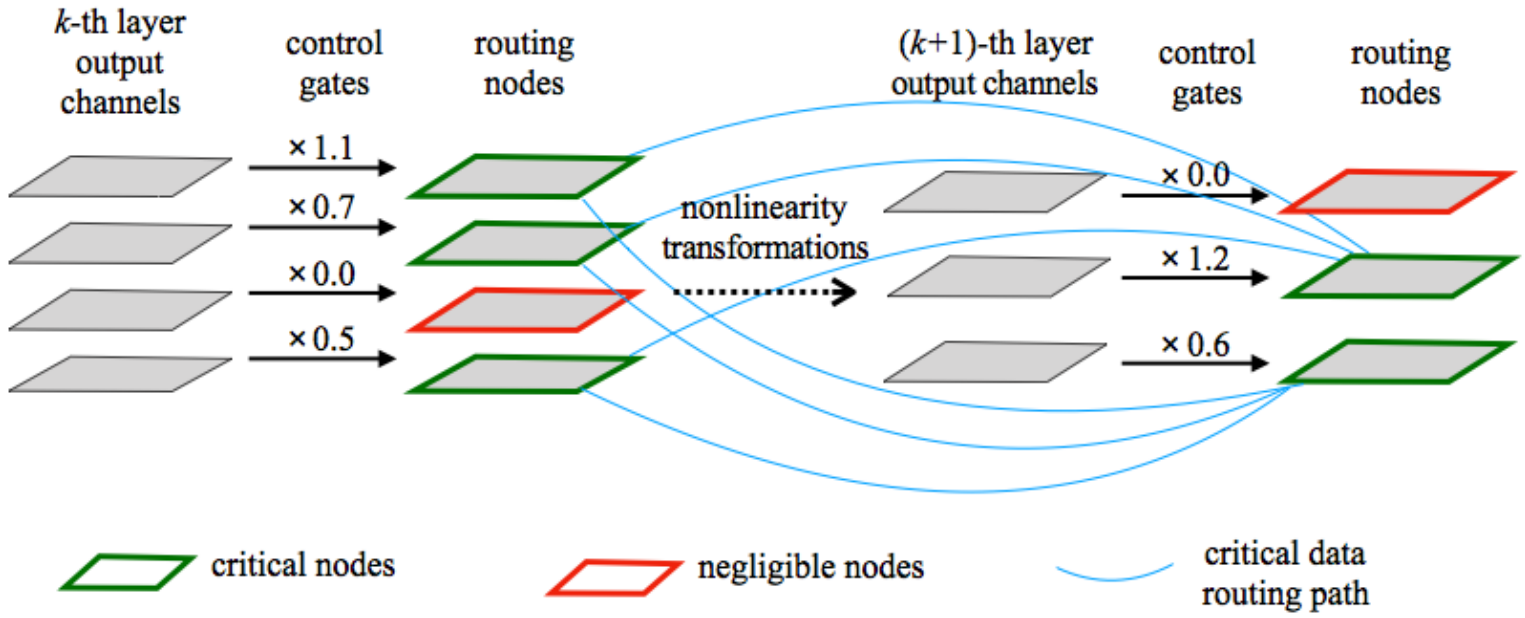


Identifying Critical Routing Paths

Y. Wang et al, "Interpret Neural Networks by Identifying Critical Data Routing Paths", in *IEEE Conference on Computer Vision and Pattern Recognition* (**CVPR**), Salt Lake City, USA, 2018.

Method

- Channel-wise control gates
  - inspired by model pruning [Liu17, He17, Anwar17]
  - associate scalar value wλth each layer's output channel
  - a group of control gates λ$_k$ultiplied to k-th layer's output channel-wise



Problem Formulation

- Learning control gates by Distillation Guided Pruning (DGP)
  - inspired by knowledge distillation [Hinton15]
  - all the control gates $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \cdots, \boldsymbol{\lambda}_K\}$
  - pretrained network : $f_\theta(\cdot)$

$$\min_{\boldsymbol{\Lambda}} \quad \mathcal{L}(f_\theta(\boldsymbol{x}), f_\theta(\boldsymbol{x}; \boldsymbol{\Lambda})) + \gamma \sum_k |\boldsymbol{\lambda}_k|_1$$

$$s.t. \quad \boldsymbol{\lambda}_k \succeq 0, k = 1, 2, \cdots, K,$$

cross entropy

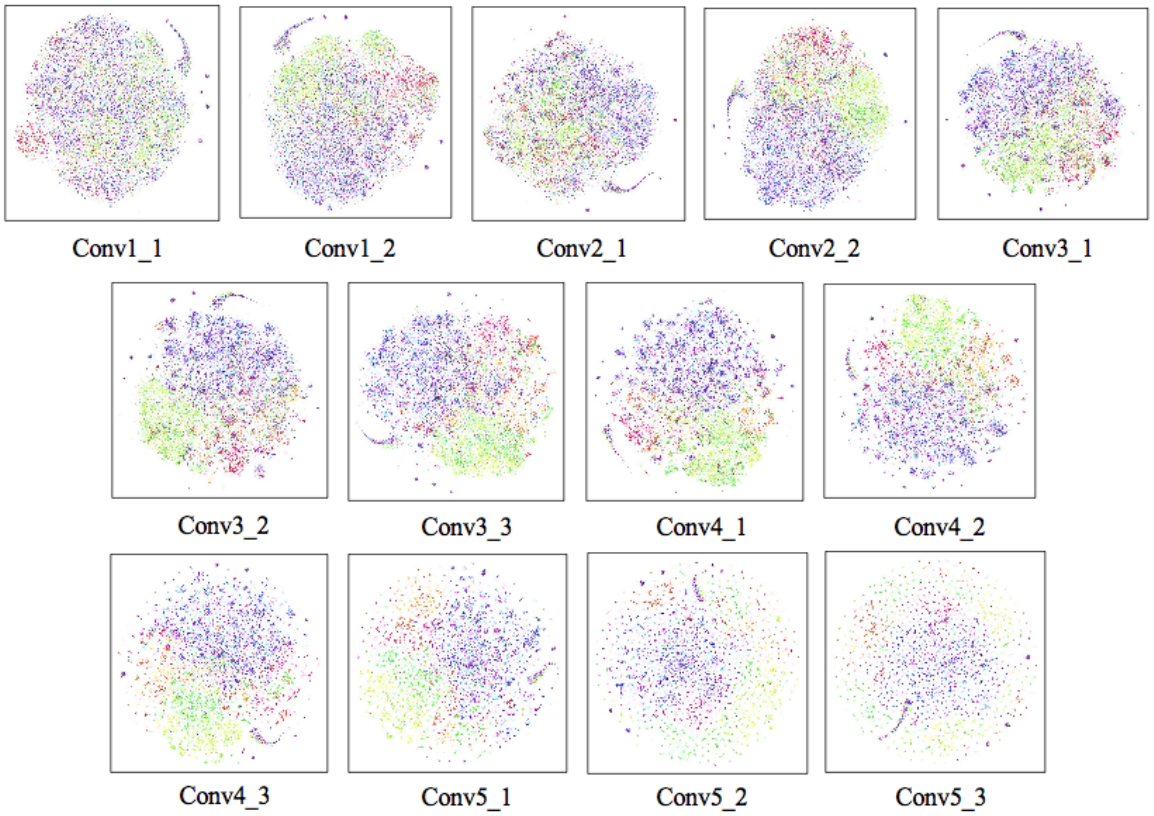
Problem Formulation (cont.)

- Routing paths representation
  - Denote $\Lambda^* = \{\boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*, \cdots, \boldsymbol{\lambda}_K^*\}$s the optimized control gates
  - the identified critical data routing paths (CDRPs)

$$\boldsymbol{v} = \text{concatenate}([\boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*, \cdots, \boldsymbol{\lambda}_K^*]).$$

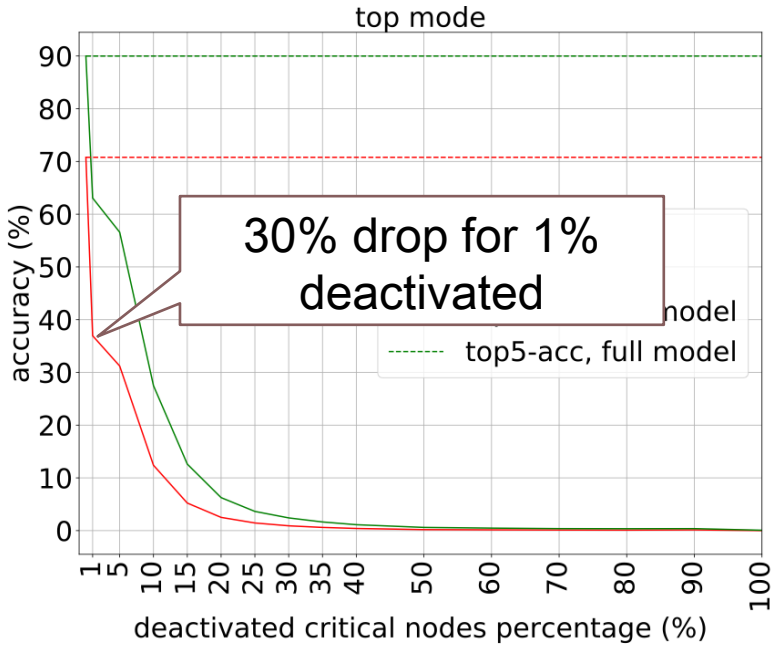
  - the actual critical nodes can be selected based on binary mask by thresholding: $\boldsymbol{v}$

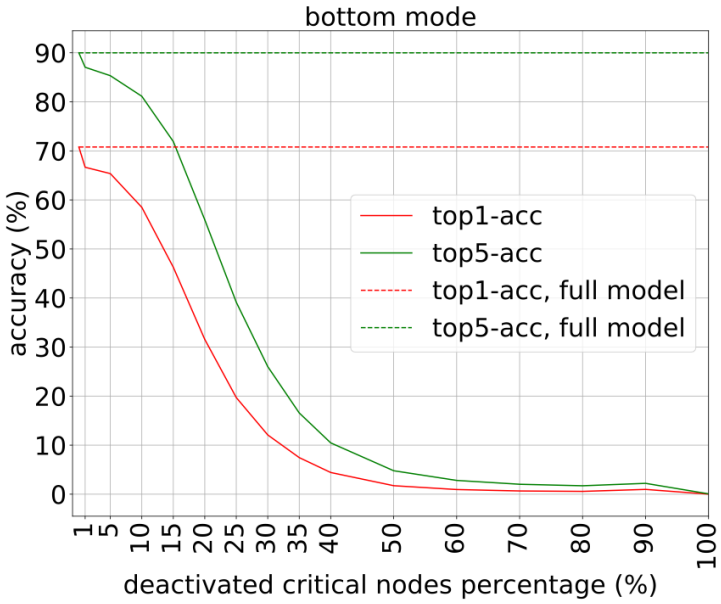
Experiments

- Functional process of intra-layer routing nodes
  - All the individual critical nodes in a certain layer composing the intra-layer routing nodes.
  - 

Experiments

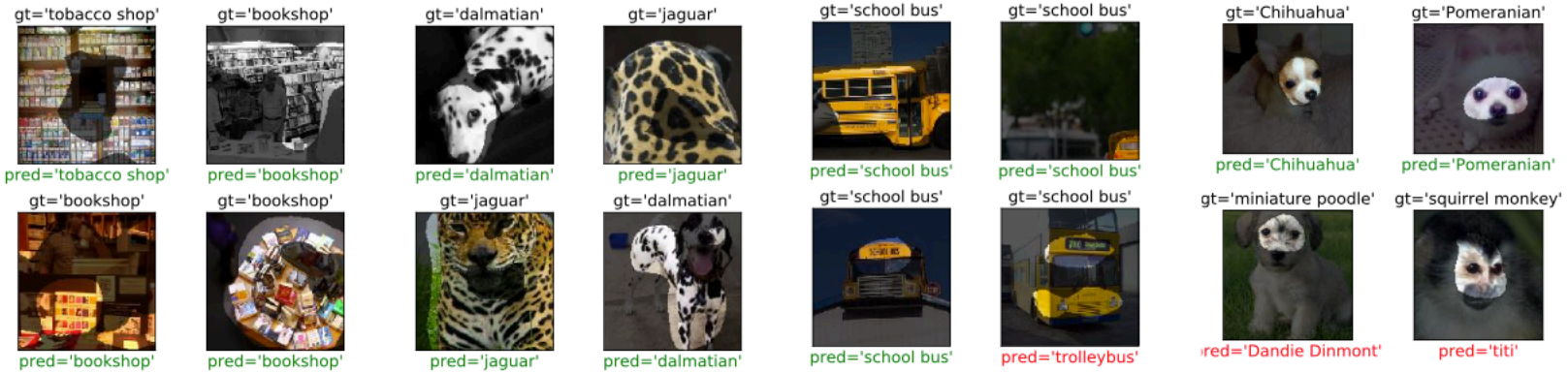
- Ablation study (VGG-16)
  - partially deactivate the critical nodes on the identified CDRPs in the full model
  - deactivates the critical nodes with larger control gates (Top Mode) or smaller (Bottom Mode)





Experiments

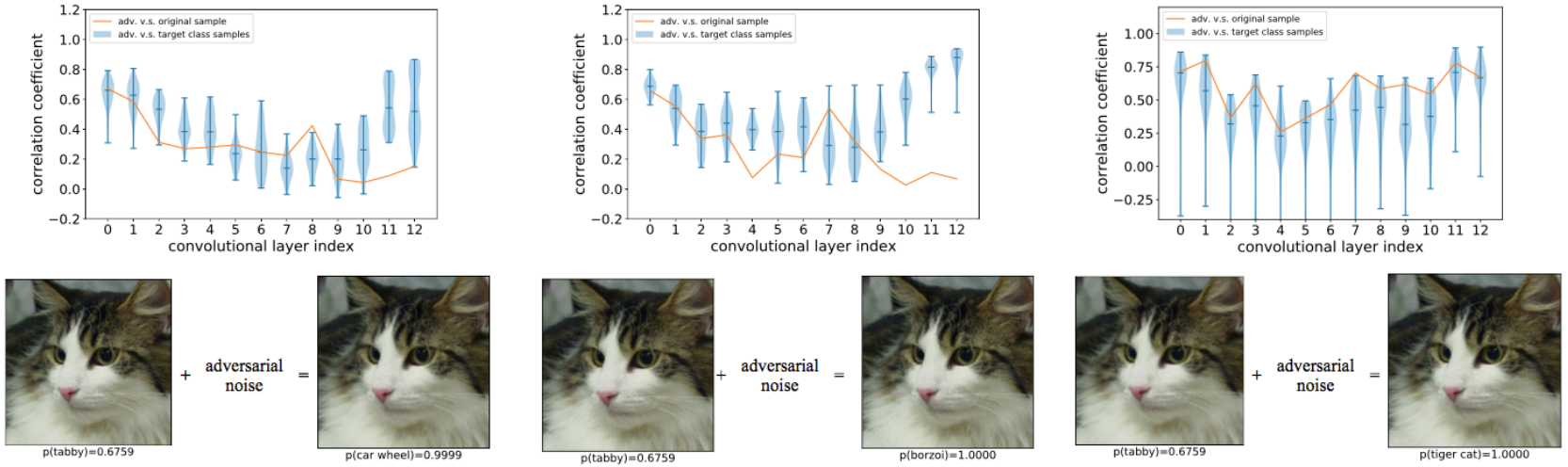
- Semantic Concepts Emerge in CDRPs
  - Individual critical node as dummy object/part detector
  - select the top k images whose corresponding control gates values on this node rank highest.



(a) critical node 156 (b) critical node 72 (c) critical node 64 (d) critical node 55

Adversarial Samples Detection

- CDRPs divergence between real and adversarial image



(a) Target adversarial attacking with target class: 'car wheel'

(b) Non-target adversarial attacking resulting in implicit target class: 'borzoi'

(c) Target adversarial attacking with semantic-closer target class: 'tiger cat'

Figure 9: Layerwise correlation coefficients of CDRPs between adversarial images and original class/target class image. In the upper part of each sub-figure, the correlation coefficients between adversarial image's routing paths and original image's routing paths are plotted in orange color. The violinplot summarizes the correlation coefficients of CDRPs between each of 50 target class images and adversarial image. (a) and (b) show that with ascending the layer level, the CDRPs of adversarial image diverge from the original image's routing paths and follow similar routing paths with those of target class images. However, when target class is semantic-closer to original class, the divergence between routing paths is not obvious

Adversarial sample detection (cont.)

- We randomly sample 1/5/10 images and 1 image of each class from ImageNet training and validation datasets as training and test samples
- Each sample is used to generate an adversarial sample by iterative FGSM [Goodfellow14]

Num. of training samples	Num. of test samples	random forest	adaboost	gradient boosting
1	1	0.8792	0.8877	0.9051
5	1	0.8942	0.9057	0.9189
10	1	0.9041	0.9104	0.9147
50	50	0.9289	0.9084	0.9220

	Num. of training samples	1	5	10
AlexNet	random forest	0.7220	0.7325	0.7770
	adaboost	0.7415	0.7505	0.7630
	gradient boosting	0.7525	0.7590	0.7845
Resnet-50	random forest	0.9120	0.9190	0.9210
	adaboost	0.9145	0.9195	0.9200
	gradient boosting	0.9255	0.9315	0.9345

Three commandments of Secure/Safe ML

- You shall not train on data you don't fully trust
  - because of data poisoning
- You shall not let anyone use your model (or observe its outputs) unless you completely trust them
  - because of model stealing and black box attacks
- You shall not fully trust the predictions of your model
  - because of adversarial examples

Conclusion

- Explore both **black-box** and **white-box** threat models
- Evaluate defences against **strong attacks** (e.g. BIM, C&W)

Thank You!