Secure Learning in Adversarial Deep Neural Networks

Bo Li
UC, Berkeley

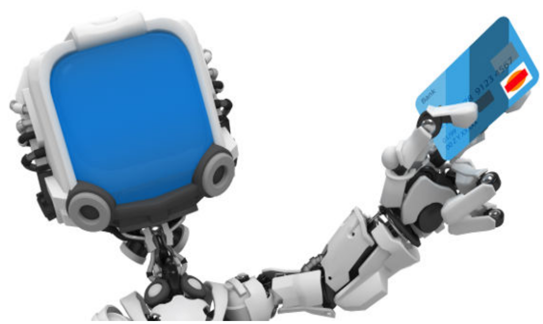
Machine Learning in Physical World

Autonomous Driving

Healthcare

Smart City

Malware Classification



Fraud Detection

Biometrics Recognition

Security & Privacy Problems


Sections

The Washington Post

WorldViews

Syrian hackers claim AP hack that tipped stock market by $136 billion. Is it terrorism?

By Max Fisher April 23, 2013

AP The Associated Press @AP Follow

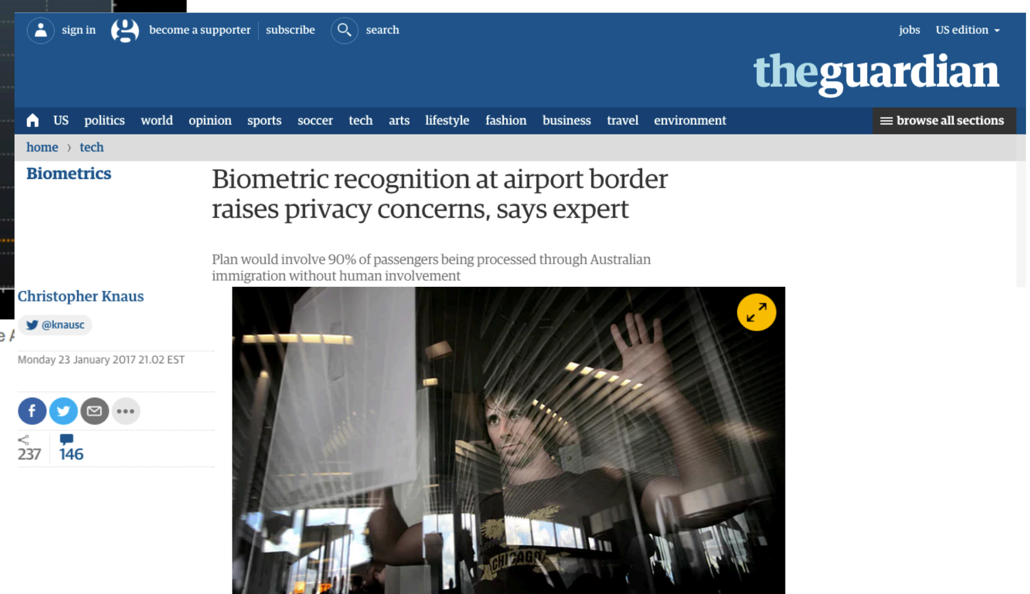
Breaking: Two Explosions in the White House and Barack Obama is injured

Reply Retweet Favorite More

579 RETWEETS 19 FAVORITES

This chart shows the Dow Jones Industrial Average during Tuesday afternoon's drop, caused by a fake A

Security Problems


sign in become a supporter subscribe search jobs US edition

theguardian

US politics world opinion sports soccer tech arts lifestyle fashion business travel environment browse all sections

home › tech

Biometrics

Biometric recognition at airport border raises privacy concerns, says expert

Plan would involve 90% of passengers being processed through Australian immigration without human involvement

Christopher Knaus

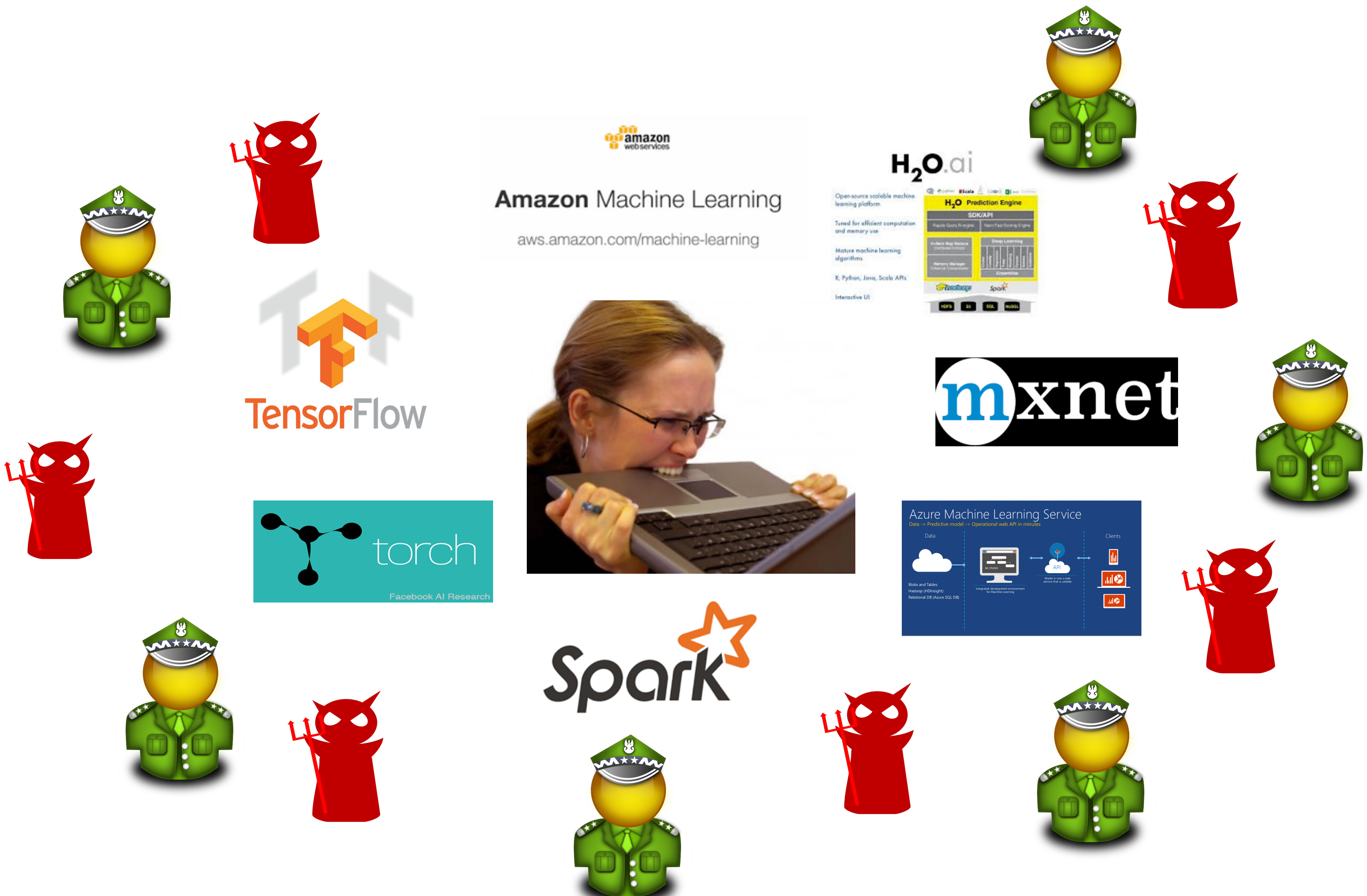
@knausc

Monday 23 January 2017 21.02 EST

237 146

Privacy Concerns

We Are in Adversarial Environments



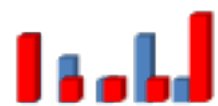
While cybersecurity R&D needs are addressed in greater detail in the NITRD Cybersecurity R&D Strategic Plan, some cybersecurity risks are specific to AI systems. ***One key research area is “adversarial machine learning”****, that explores the degree to which AI systems can be compromised by “contaminating” training data, by modifying algorithms, or by making subtle changes to an object that prevent it from being correctly identified….*

- National Science and Technology Council

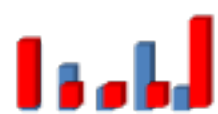
2016

Perils of Stationary Assumption

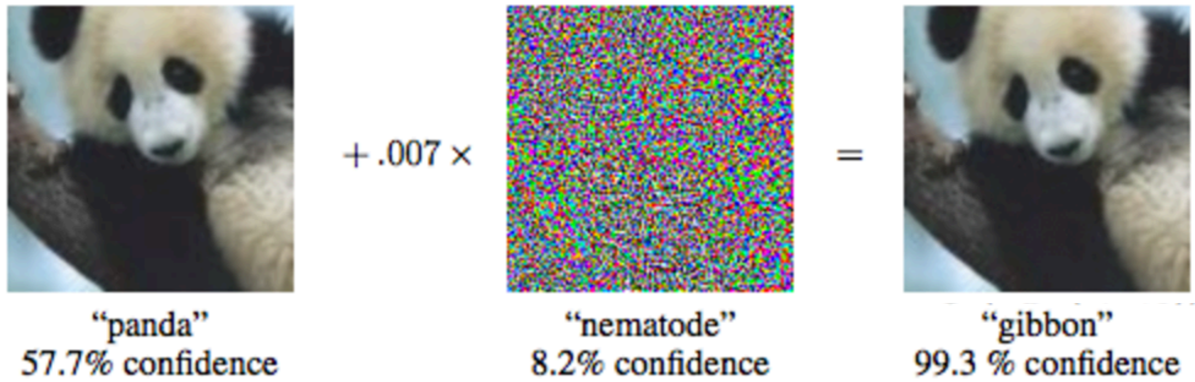
Traditional machine learning approaches assume

Training Data 

$\approx$

Testing Data 

Adversarial Examples



$$\boldsymbol{\eta} = \epsilon \operatorname{sign}(\nabla_x J(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}))$$

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *ICLR 2015.*
Li, Bo, Yevgeniy Vorobeychik, and Xinyun Chen. "A General Retraining Framework for Scalable Adversarial Classification." ICLR. (2016).

Optimization Based Attack

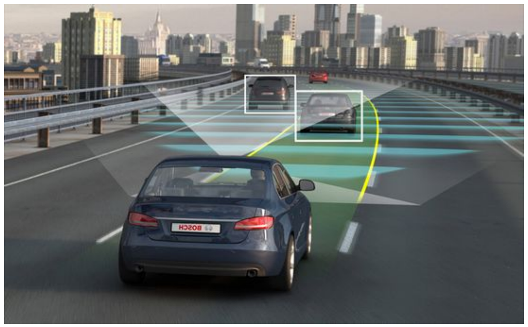
$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x+\delta) \\ \text{such that} \quad & C(x+\delta) = t \\ & x+\delta \in [0,1]^n \end{aligned}$$

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

	Best Case				**Average Case**				**Worst Case**			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
Our L_0	8.5	100%	5.9	100%	16	100%	13	100%	33	100%	24	100%
JSMA-Z	20	100%	20	100%	56	100%	58	100%	180	98%	150	100%
JSMA-F	17	100%	25	100%	45	100%	110	100%	100	100%	240	100%
Our L_2	1.36	100%	0.17	100%	1.76	100%	0.33	100%	2.60	100%	0.51	100%
Deepfool	2.11	100%	0.85	100%	–	-	–	-	–	-	–	-
Our L_∞	0.13	100%	0.0092	100%	0.16	100%	0.013	100%	0.23	100%	0.019	100%
Fast Gradient Sign	0.22	100%	0.015	99%	0.26	42%	0.029	51%	–	0%	0.34	1%
Iterative Gradient Sign	0.14	100%	0.0078	100%	0.19	100%	0.014	100%	0.26	100%	0.023	100%

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

Autonomous Driving is the Trend…



However, What We Can See Everyday…

The Physical World Is... Messy

Varying Physical Conditions (Angle, Distance, Lighting, …)

Physical Limits on Imperceptibility

Fabrication/Perception Error (Color Reproduction, etc.)

Digital Noise (What you want)

What is printed

What a camera may see

Background Modifications*

Image Courtesy, OpenAI

[Evtimov, Eykholt, Fernandes, Kohno, Li, Prakash, Rahmati, and Song, *2017*]

An Optimization Approach To Creating Robust Physical Adversarial Examples

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

Perturbation/Noise Matrix

Adversarial Target Label

Lp norm (L-0, L-1, L-2, …)

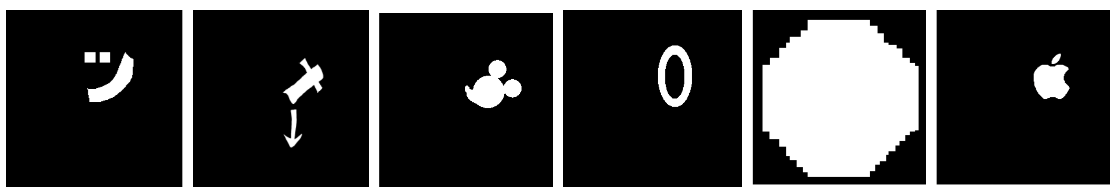
Loss Function

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

$$\left\{ \quad \right\}$$

Optimizing Spatial Constraints (Handling Limits on Imperceptibility)

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



Subtle Poster

Camouflage Sticker

Mimic vandalism

“Hide in the human psyche”

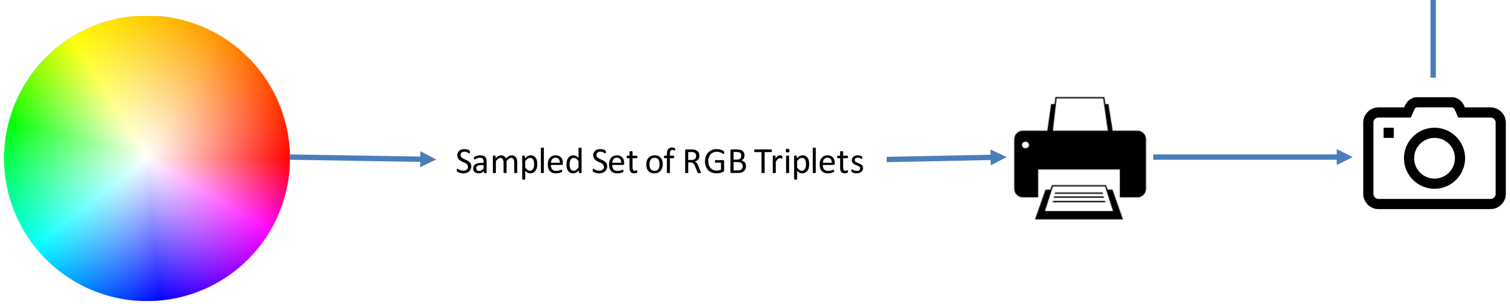
Handling Fabrication/Perception Errors

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

$$NPS(\delta) = \sum_{\hat{p} \in \delta} \prod_{p' \in P} |\hat{p} - p'|$$

P is a set of printable RGB triplets

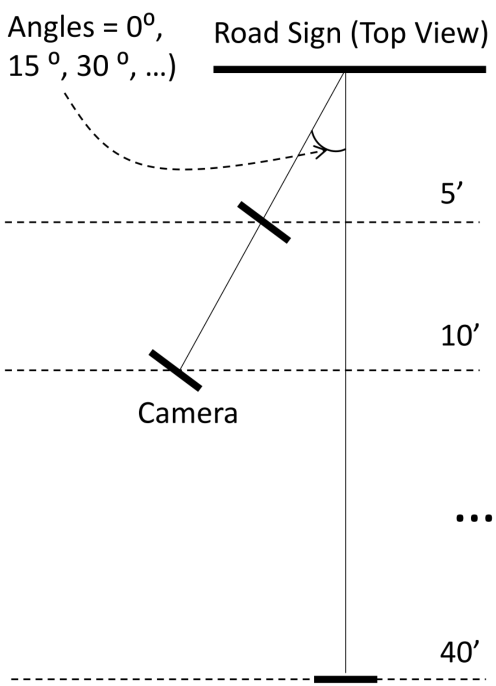
Color Space



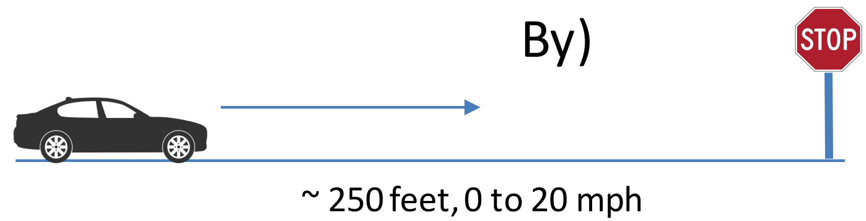
NPS based on Sharif et al., “Accessorize to a crime,” CCS 2016

How Can We Realistically Evaluate Attacks?

Lab Test (Stationary)



Field Test (Drive-By)



Record video

Sample frames every k frames

Run sampled frames through DNN

Lab Test Summary (Stationary)

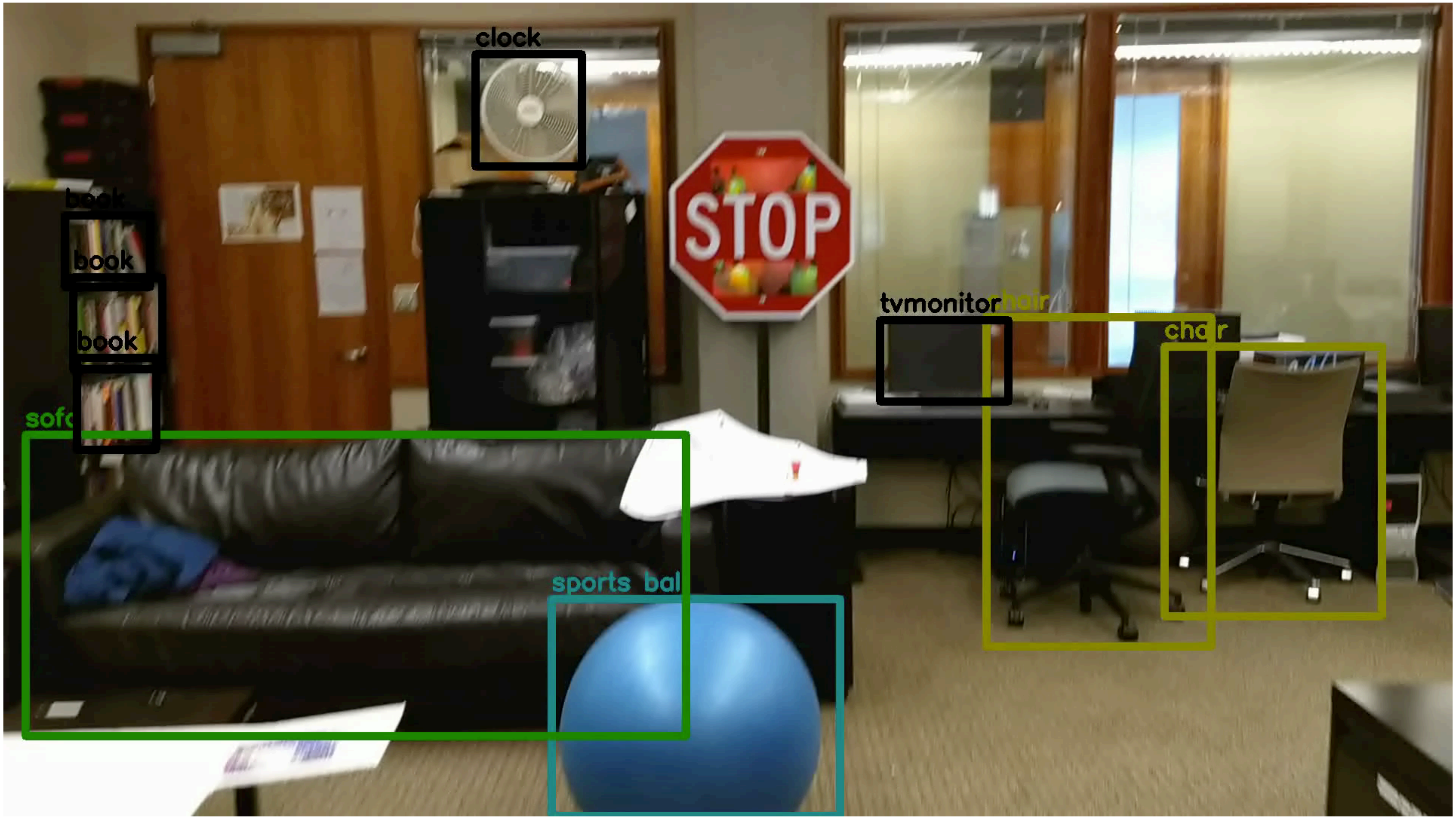
Subtle Poster | Subtle Poster | Camo Graffiti | Camo Art | Camo Art

Target Class: Speed Limit 45

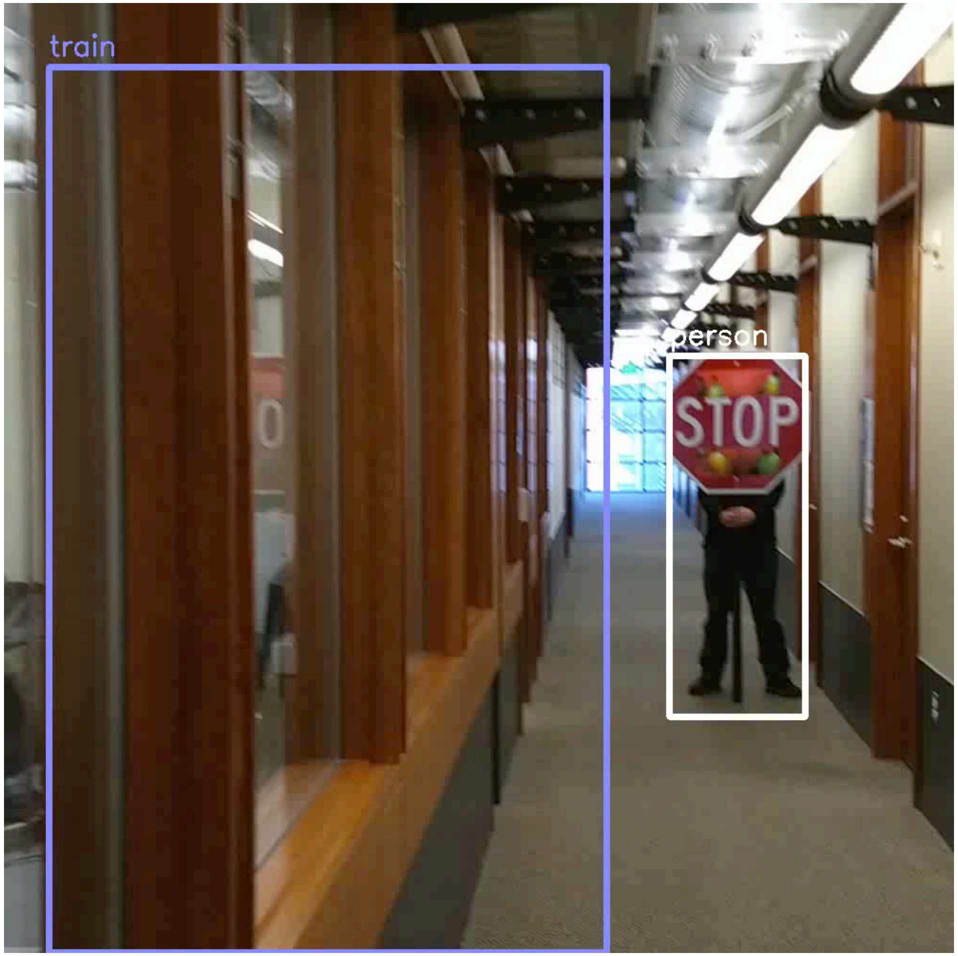
Art Perturbation

Subtle Perturbation

Physical Attacks Against Detectors



Physical Attacks Against Detectors



Adversarial Examples in Physical World

Adversarial perturbations are possible in physical world under different conditions and viewpoints, including the distances and angles.

Different approaches to optimize the objective

- Fast approaches
  - Fast gradient sign ($d = ||\cdot||_\infty$): $x^* = x + B\mathrm{sgn}\left(\nabla_x \ell(f_\theta(x), y)\right)$
  - Fast gradient ($d = ||\cdot||_2$): $x^* = x + B\left(\frac{\nabla_x \ell(f_\theta(x),y)}{||\nabla_x \ell(f_\theta(x),y)||_2}\right)$
- Iterative approaches
  - E.g., use a SGD optimizer, such as Adam, to optimize

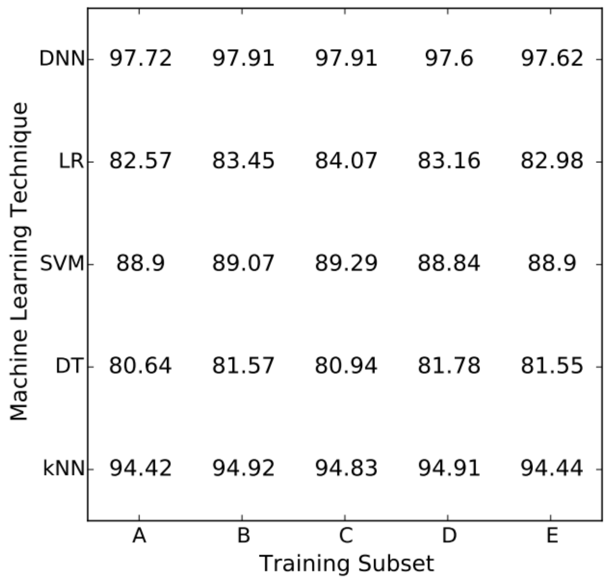
$$\max_{x^*} \ell(f_\theta(x^*), y) + \lambda d(x, x^*)$$

- Optimization $\underset{\delta}{\operatorname{argmin}}\ \lambda||\delta||_p + J(f_\theta(x+\delta), y^*)$
- **Need to know model f_θ**

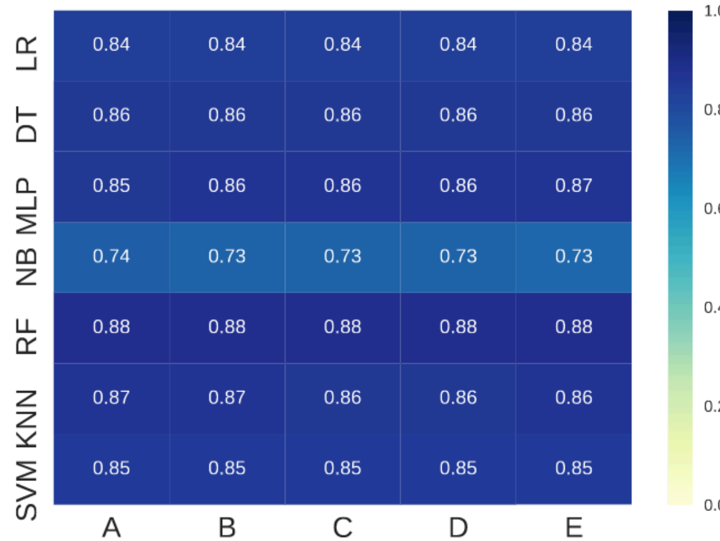
A General Framework for Black-box attacks

- Zero-Query Attack
  - Random perturbation
  - Difference of means
  - *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 whitebox attack
  - A general active query game model

Transferability



MNIST



PDF Malware

Papernot, McDaniel, Goodfellow, Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. 2016
Xiao, Li, Malware Evasion Attacks Based on Generative Adversarial Networks (GANs), 2017.

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!**

Ground truth: running shoe

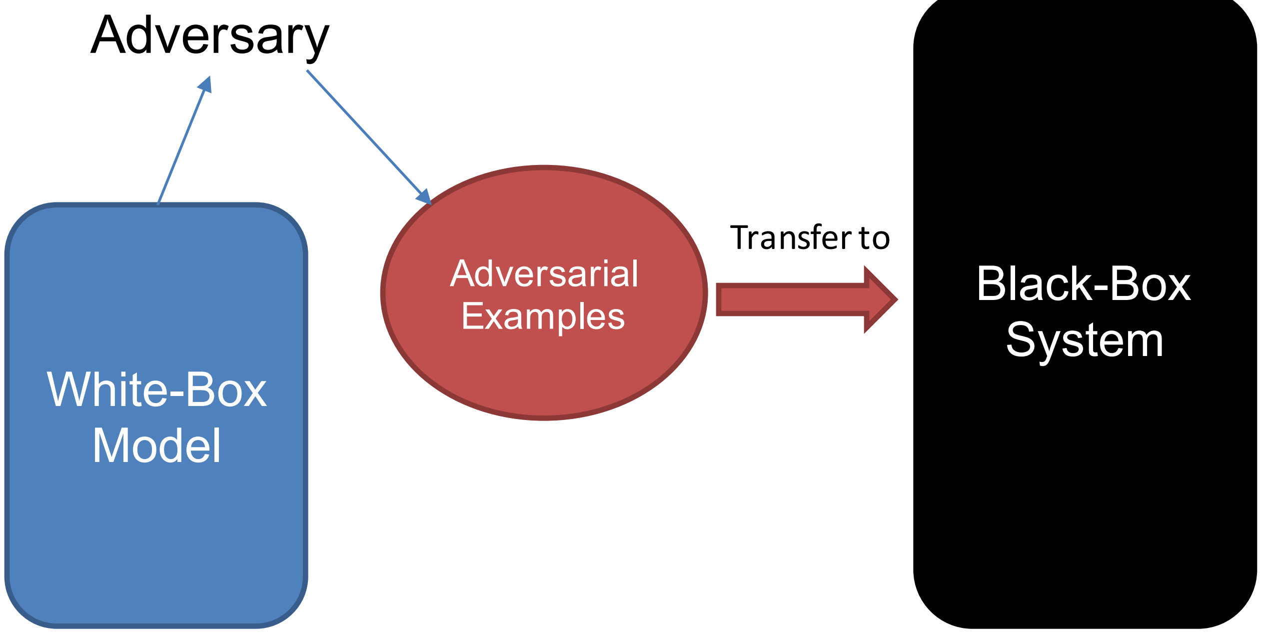
VGG16	**Military uniform**
ResNet50	Jigsaw puzzle
ResNet101	Motor scooter
ResNet152	Mask
GoogLeNet	Chainsaw

Targeted Adversarial Example's Transferability Among **Two Models** is **Poor**!

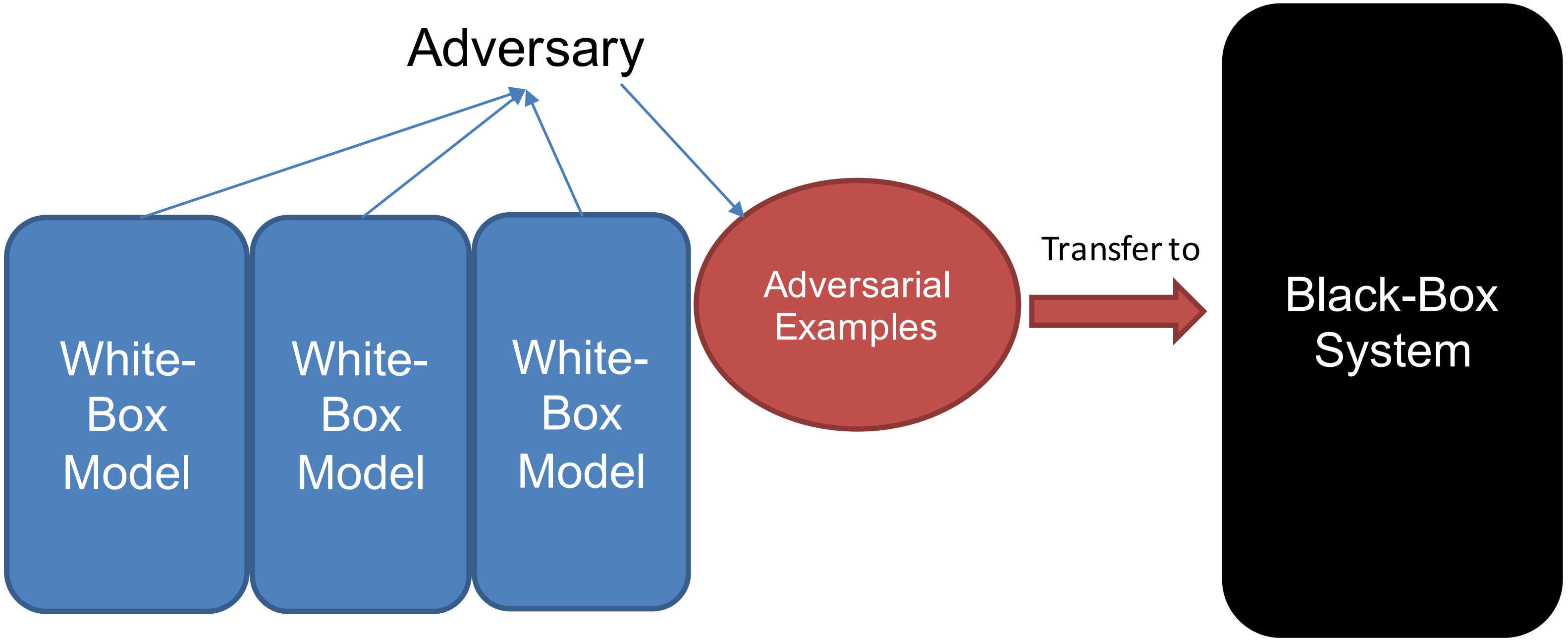
	ResNet152	ResNet101	ResNet50	VGG16	GoogLeNet	Incept-v3
ResNet152	100%	2%	1%	1%	1%	0%
ResNet101	3%	100%	3%	2%	1%	1%
ResNet50	4%	2%	100%	1%	1%	0%
VGG16	2%	1%	2%	100%	1%	0%
GoogLeNet	1%	1%	0%	1%	100%	0%
Incept-v3	0%	0%	0%	0%	0%	100%

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

Black-box Attacks Based On Transferability



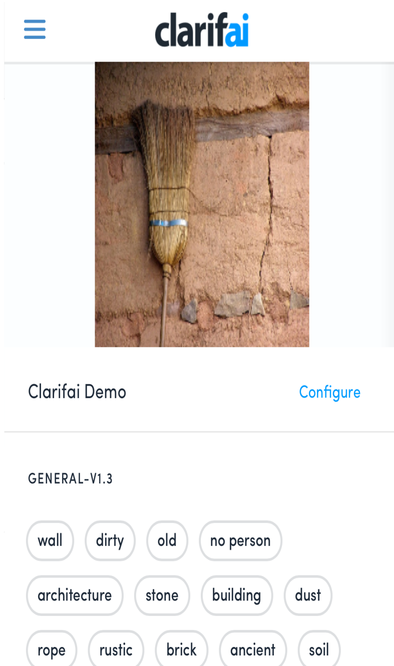
Ensemble Targeted Black-box Attacks Based On Transferability



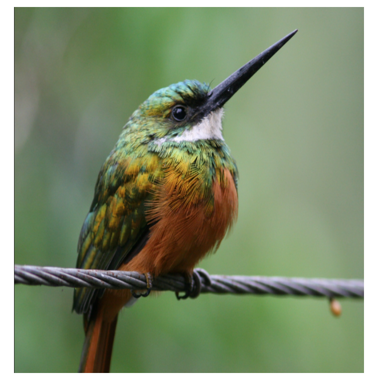
Liu, Chen, Liu, Song. Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

Clarifai.com

Ground truth from ImageNet: broom

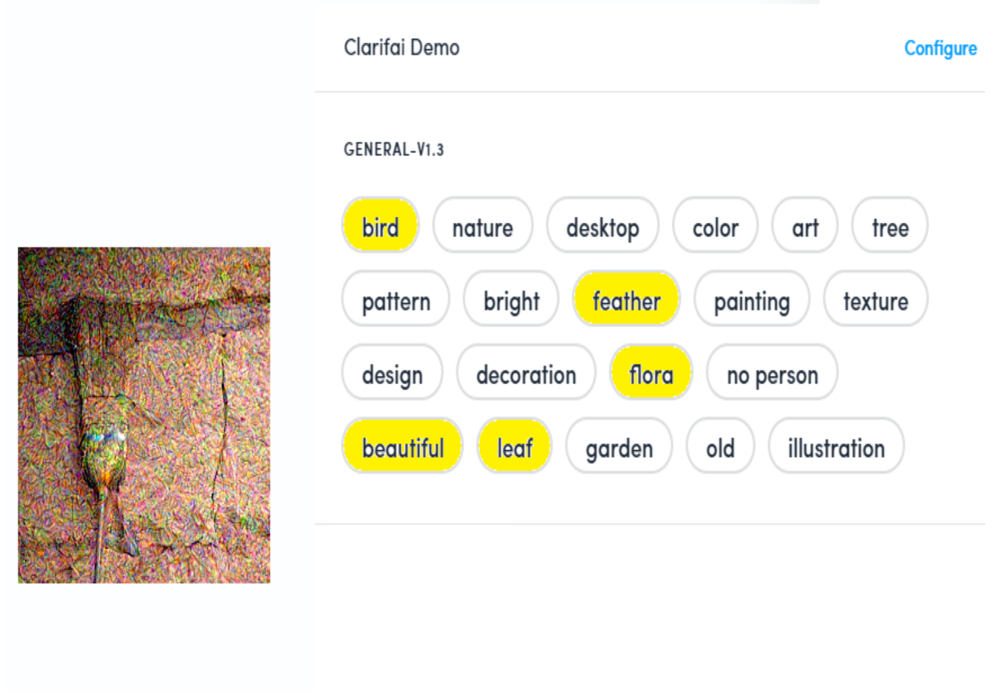


jacamar



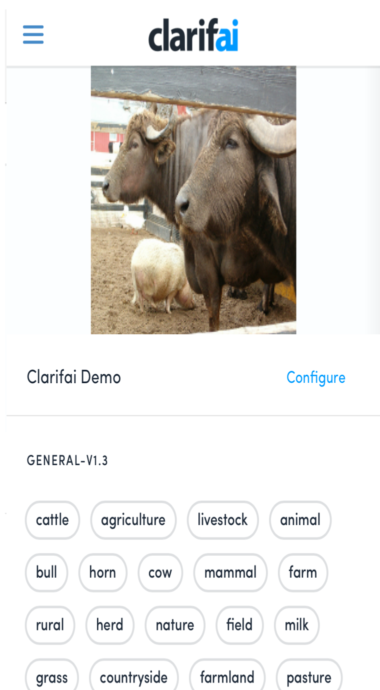
Adversarial Example on Clarifai.com

- Ground truth: broom
- Target label: **jacamar**



Clarifai.com

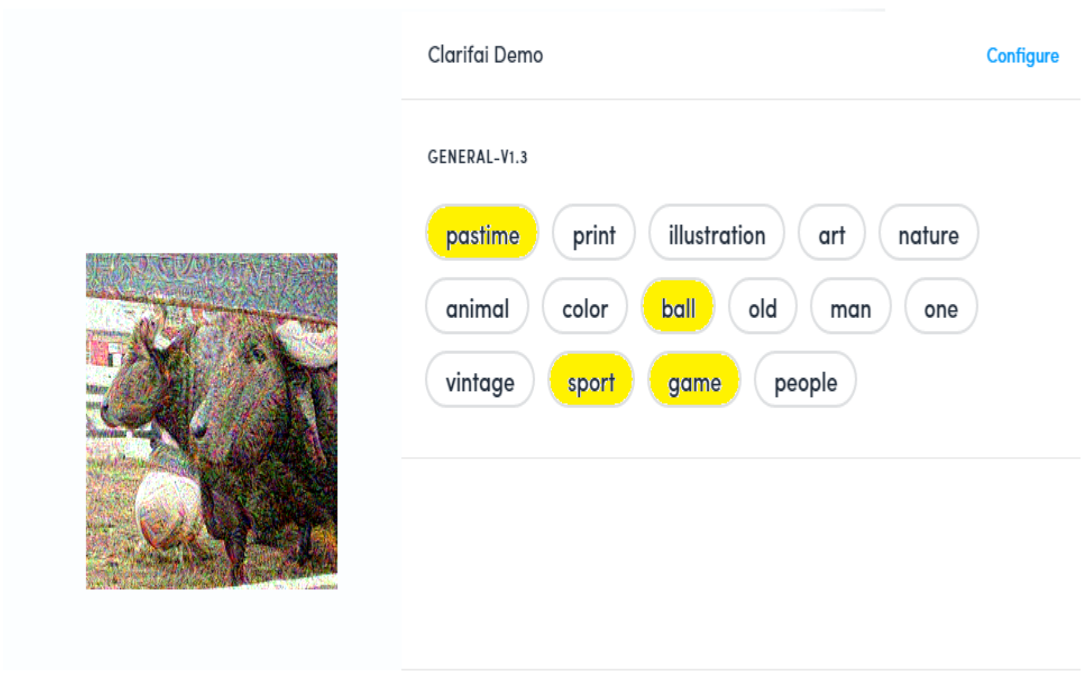
Ground truth on ImageNet: Waterbuffalo



rugby ball

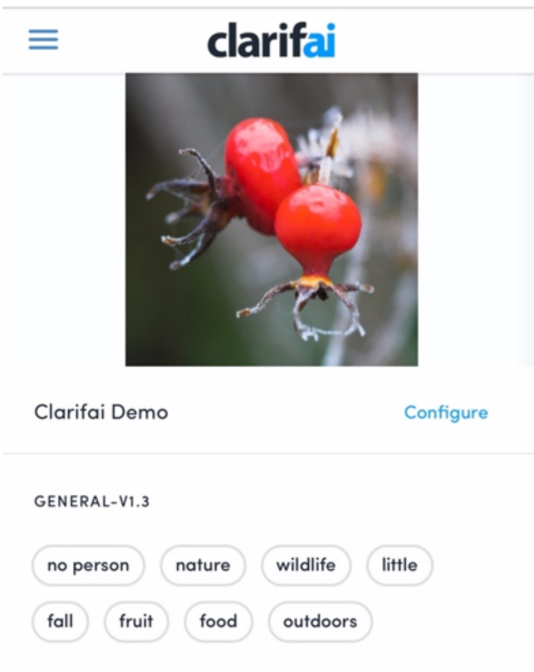
Adversarial Example on Clarifai.com

- Ground truth: water buffalo
- Target label: **rugby ball**



Clarifai.com

Ground truth from ImageNet: rosehip



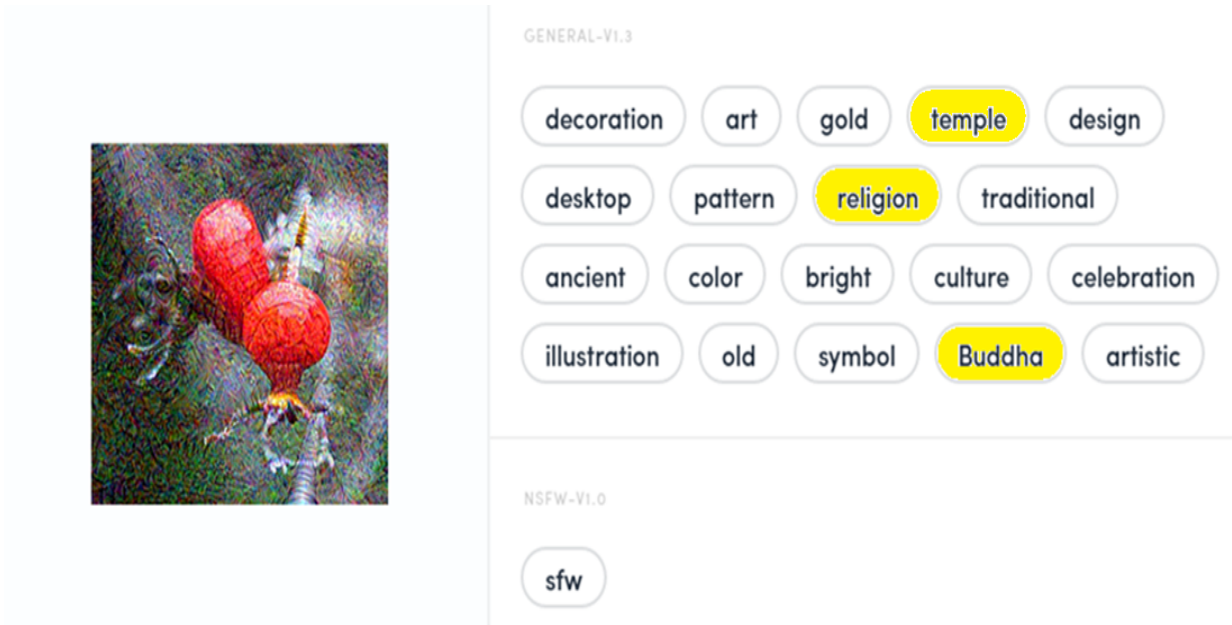
?

stupa

LCLS17. Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

Adversarial Example on Clarifai.com

- Ground truth: rosehip
- Target label: **stupa**



LCLS17. Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

Black-box attacks

- Zero-Query Attack (Previous methods)
  - Random perturbation
  - Difference of means
  - Transferability based attack
- Query Based Attack (Our methods)
  - Finite difference gradient estimation
  - Query reduced gradient estimation

The zero-query attack can be viewed as a special case for the query based attack, where the number of queries made is zero

[Bhagoji, Li, He, Dawn, 2017]

Query Based attacks

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

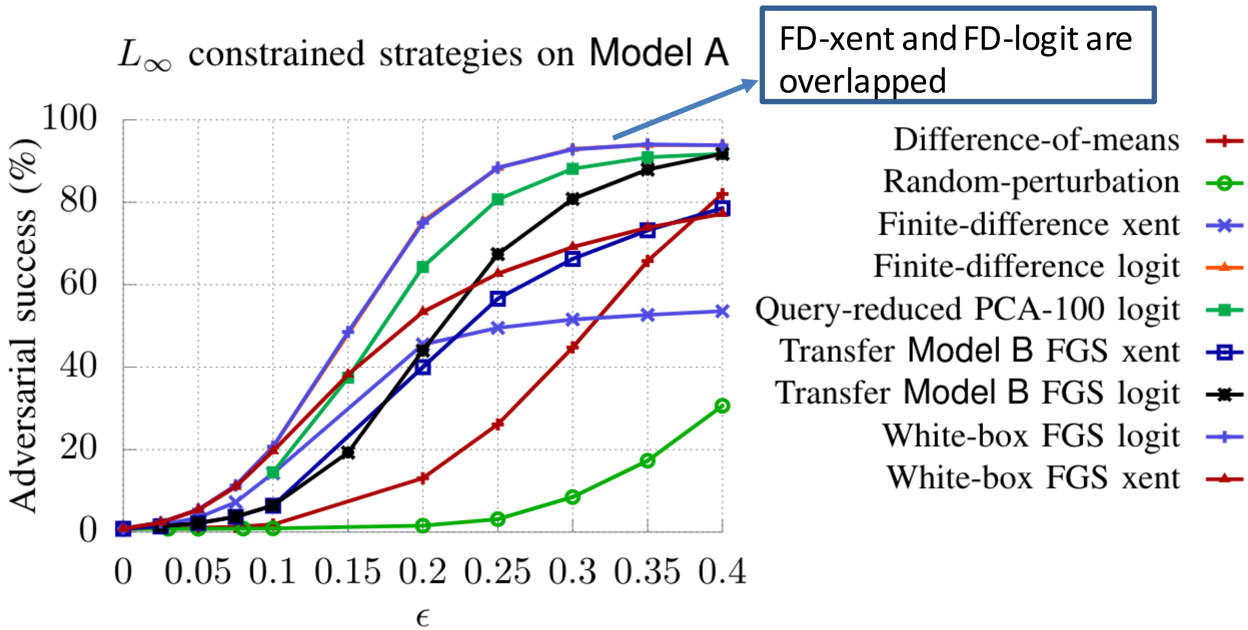
$$\mathrm{FD}_{\mathbf{x}}(g(\mathbf{x}),\delta) = \begin{bmatrix} \frac{g(\mathbf{x}+\delta\mathbf{e}_1)-g(\mathbf{x}-\delta\mathbf{e}_1)}{2\delta} \\ \vdots \\ \frac{g(\mathbf{x}+\delta\mathbf{e}_d)-g(\mathbf{x}-\delta\mathbf{e}_d)}{2\delta} \end{bmatrix}$$

  - An example of approximate FGS with finite difference

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

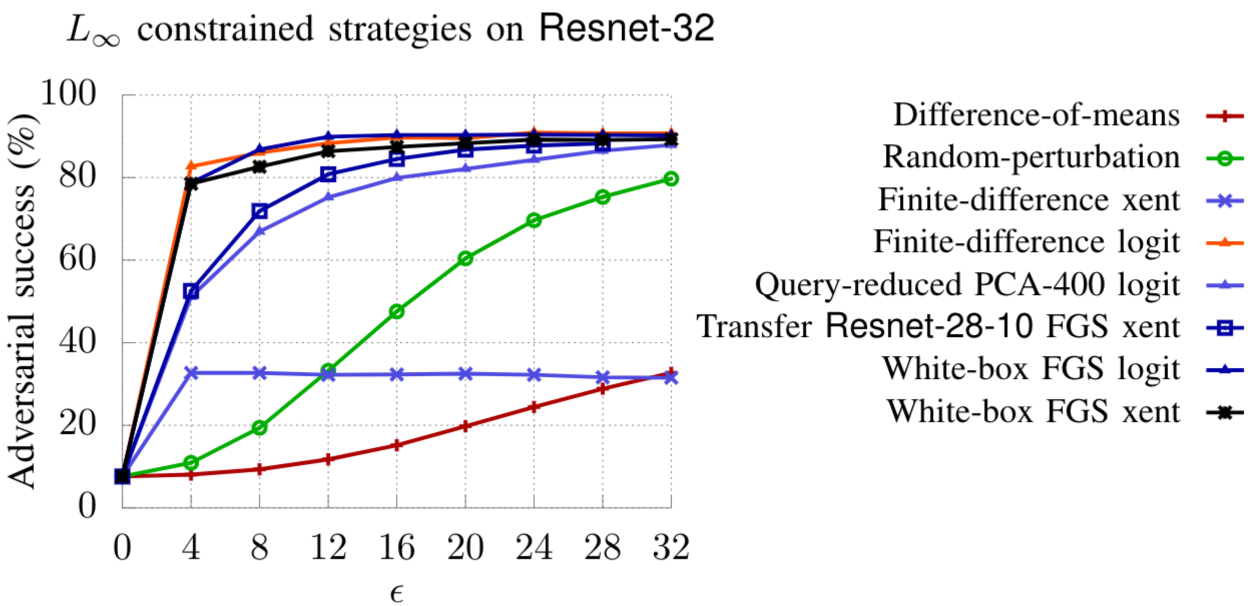
Similarly, we can also approximate for logit-based loss by making 2d queries

- Query reduced gradient estimation
  - Random grouping
  - PCA



Effectiveness of various single step black-box attacks on MNIST. The y-axis represents the variation in adversarial success as ϵ increases.

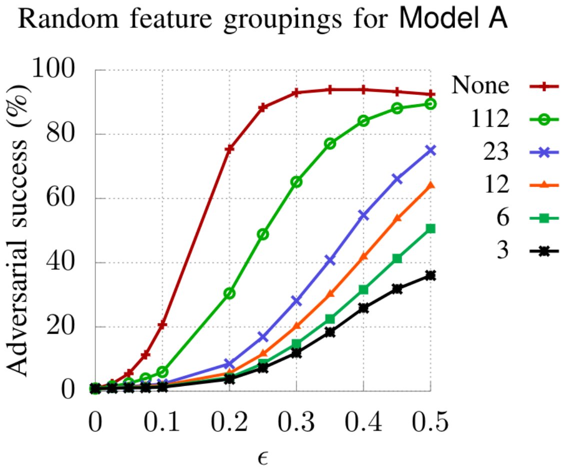
Finite Differences method outperform other black-box attacks and achieves similar attach success rate with the white-box attack

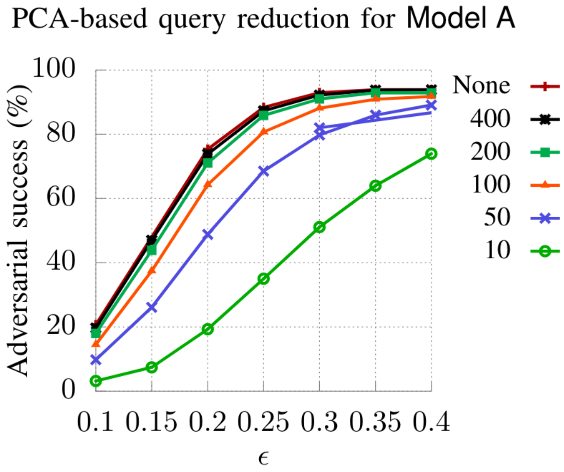


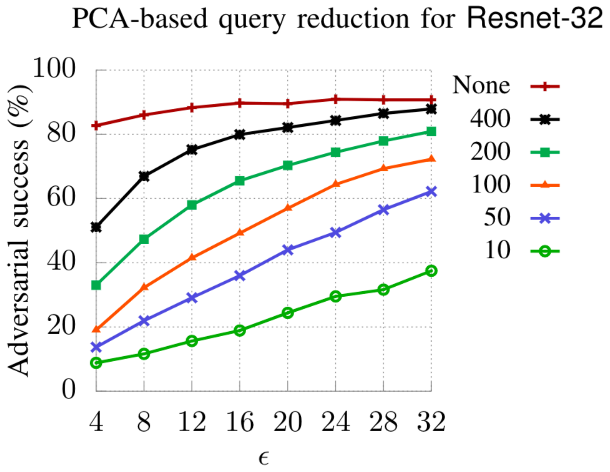
Effectiveness of various single step black-box attacks on CIFAR-10. The y-axis represents the variation in adversarial success as ϵ increases.

Finite Differences method outperform other black-box attacks and achieves similar attach success rate with the white-box attack

Gradient Estimation Attack with Query Reduction







Adversarial success rates for Gradient Estimation attacks with query reduction on Model A (MNIST) and Resnet-32 (CIFAR-10).

Finite Differences method with query reduction perform approximately similar with the gradient estimation black-box attack

Black-box Attack Clarifai

Original image, classified as "drug" with a confidence of 0.99

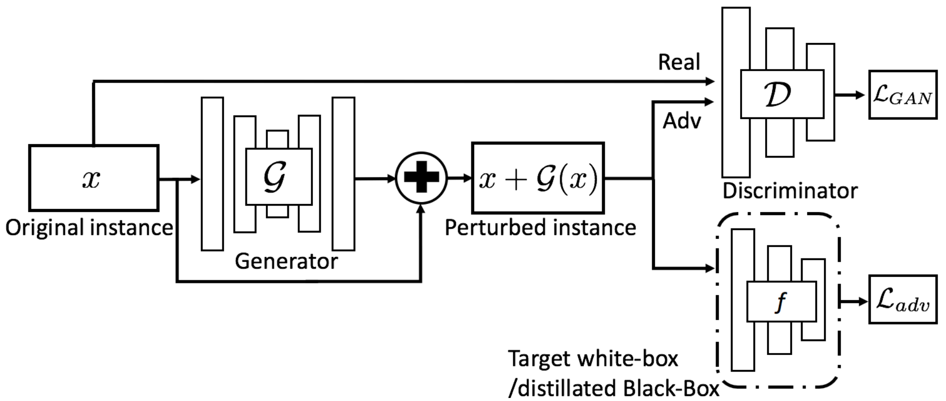
Adversarial example, classified as "safe" with a confidence of 0.96

The Gradient Estimation black-box attack on Clarifai's Content Moderation Model

Black-box Attacks

Black-box attacks are possible on deep neural networks with query access.
The number of queries needed can be reduced.

Generating Adversarial Examples with Adversarial Networks



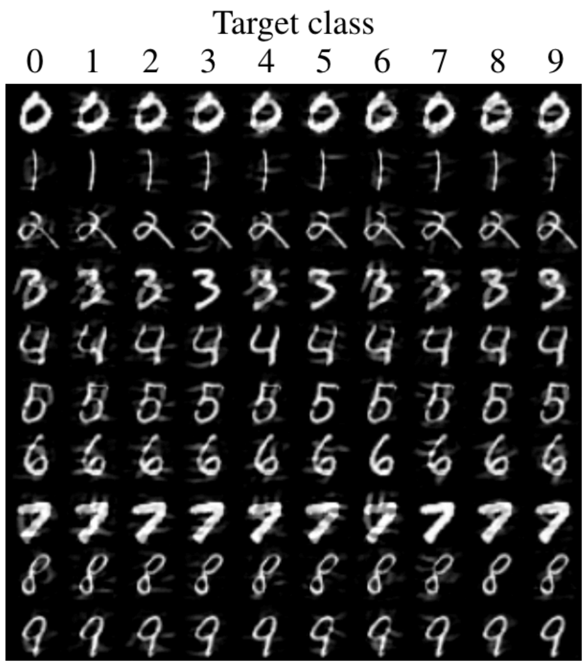
Black-box can be performed here via distillation

$$\mathcal{L}_{GAN} = \mathbb{E}_{x \sim \mathcal{P}_{\text{data}}(x)} \log \mathcal{D}(x) + \mathbb{E}_{x \sim \mathcal{P}_{\text{data}}(x)} \log(1 - \mathcal{D}(x + \mathcal{G}(x)))$$

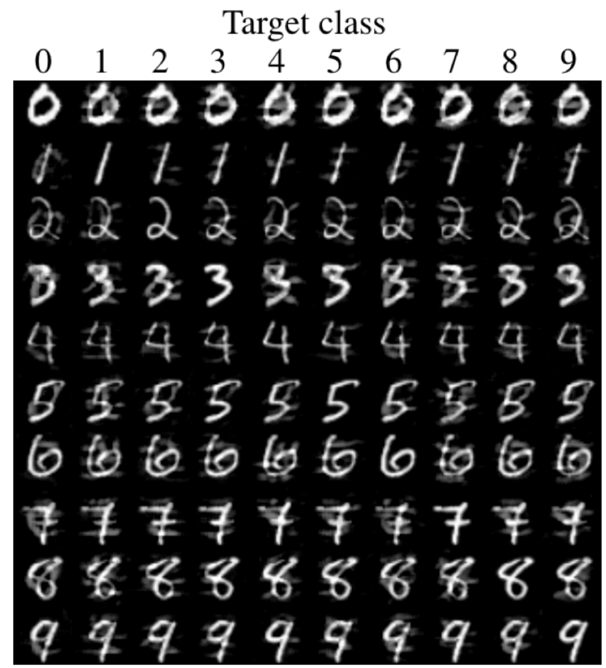
$$\mathcal{L} = \mathcal{L}_{adv}^{f} + \alpha \mathcal{L}_{GAN} + \beta \mathcal{L}_{hinge}$$

The GAN loss here tries to ensure the diversity of adversarial examples

[Chaowei Xiao, Bo Li, Jun-yan Zhu, Warren He, Mingyan Liu, Dawn Song, 2017]

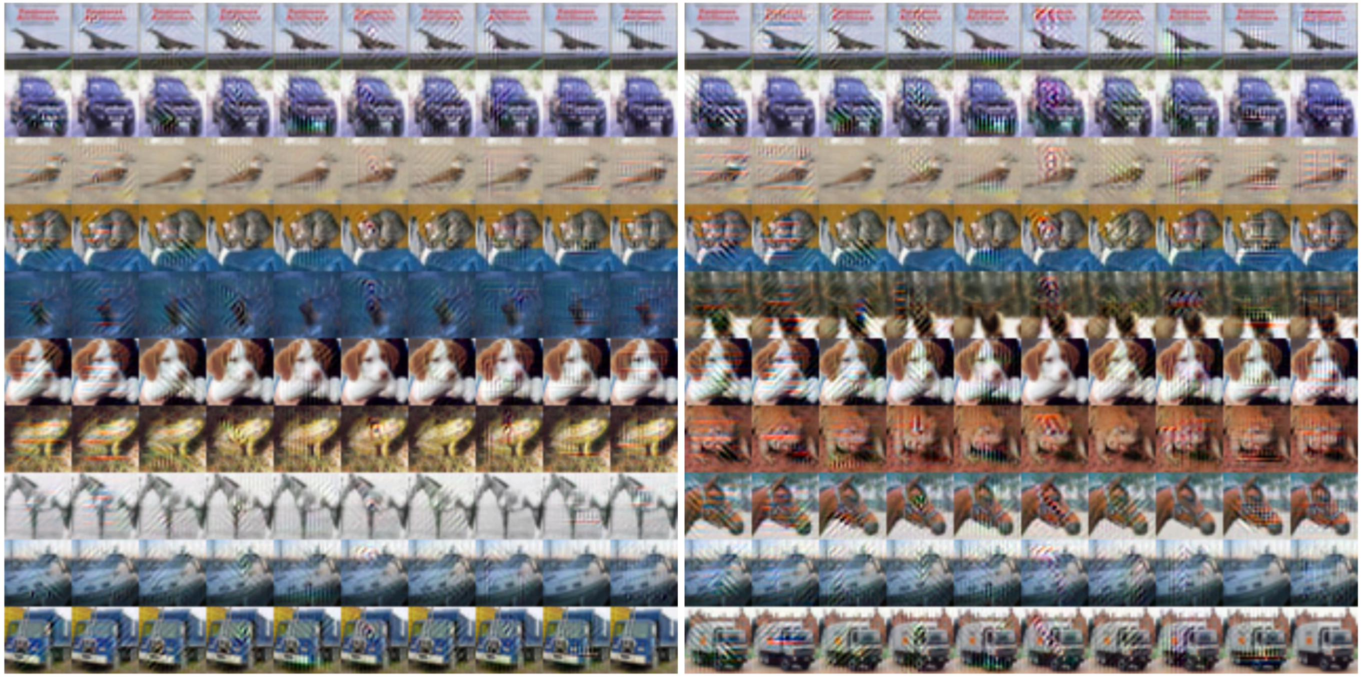


Semi-white box attack on MNIST



Black-box attack on MNIST

The perturbed images are very close to the original ones. The original images lie on the diagonal.



(a) Semi-whitebox setting

(b) Black-box setting

The perturbed images are very close to the original ones. The original images lie on the diagonal.

Poodle Ambulance Basketball Electric guitar


(a) Strawberry


(b) Toy poodle


(c) Buckeye


(d) Toy poodle

Attack Effectiveness Under Defenses

Data	Model	Defense	FGSM	Opt.	AdvGAN
MNIST	A	Adv.	4.3%	4.6%	**8.0%**
		Ensemble	1.6%	4.2%	**6.3%**
		Iter.Adv.	4.4%	2.96%	**5.6%**
	B	Adv.	6.0%	4.5%	**7.2%**
		Ensemble	2.7%	3.18%	**5.8%**
		Iter.Adv.	**9.0%**	3.0%	6.6%
	C	Adv.	2.7%	2.95%	**18.7%**
		Ensemble	1.6%	2.2%	**13.5%**
		Iter.Adv.	1.6%	1.9%	**12.6%**
CIFAR	ResNet	Adv.	13.10%	11.9%	**16.03%**
		Ensemble.	10.00%	10.3%	**14.32%**
		Iter.Adv	22.8%	21.4%	**29.47%**
	Wide ResNet	Adv.	5.04%	7.61%	**14.26%**
		Ensemble	4.65%	8.43%	**13.94 %**
		Iter.Adv.	14.9%	13.90%	**20.75%**

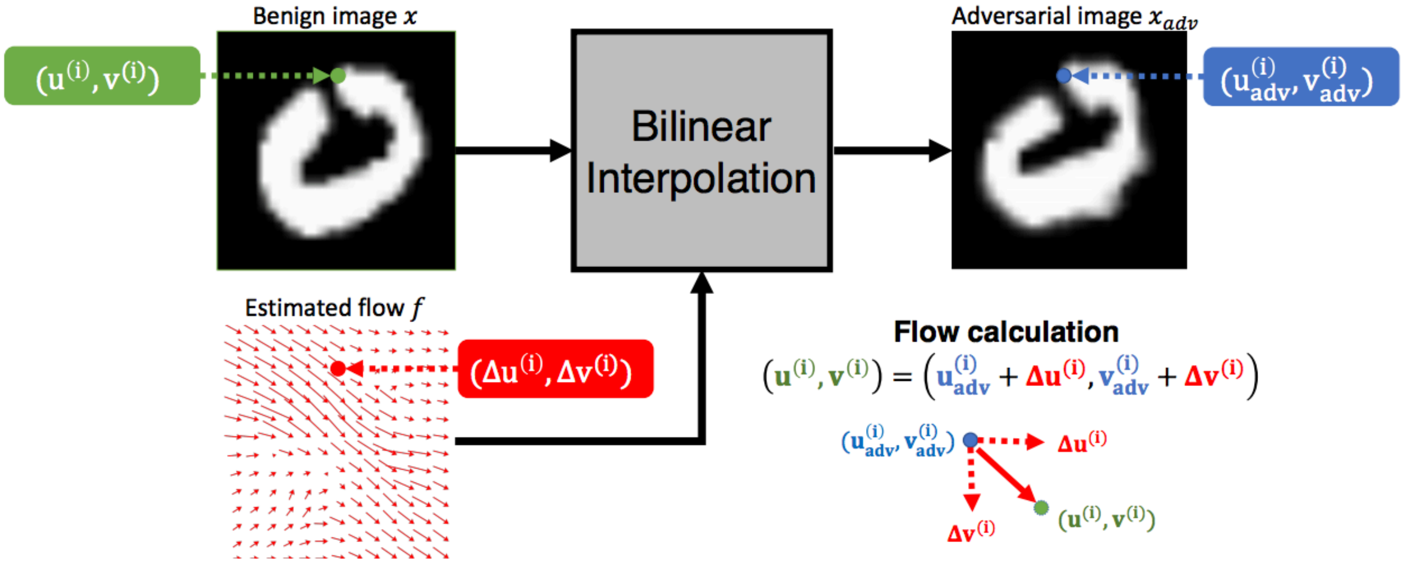
Attack success rate of adversarial examples generated by AdvGAN in semi-whitebox setting under defenses on MNIST and CIFAR-10

Attack Effectiveness Under Defenses

Black-Box Leaderboard (Original Challenge)

Attack	Submitted by	Accuracy	Submission Date
AdvGAN from "Generating Adversarial Examples with Adversarial Networks"	AdvGAN	**92.76%**	Sep 25, 2017
PGD against three independently and adversarially trained copies of the network	Florian Tramèr	93.54%	Jul 5, 2017
FGSM on the CW loss for model B from "Ensemble Adversarial Training [...]"	Florian Tramèr	94.36%	Jun 29, 2017
FGSM on the CW loss for the naturally trained public network	(initial entry)	96.08%	Jun 28, 2017
PGD on the cross-entropy loss for the naturally trained public network	(initial entry)	96.81%	Jun 28, 2017
Attack using Gaussian Filter for selected pixels on the adversarially trained public network	Anonymous	97.33%	Aug 27, 2017
FGSM on the cross-entropy loss for the adversarially trained public network	(initial entry)	97.66%	Jun 28, 2017
PGD on the cross-entropy loss for the adversarially trained public network	(initial entry)	97.79%	Jun 28, 2017

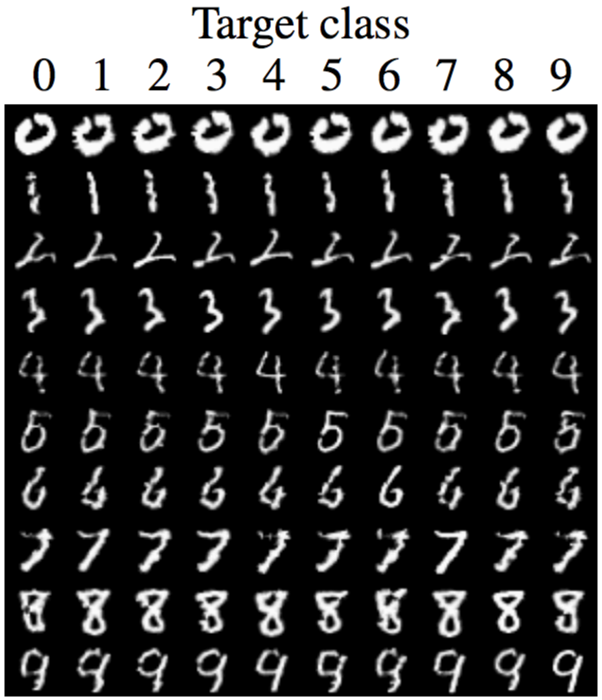
Spatially Transformed Adversarial Examples



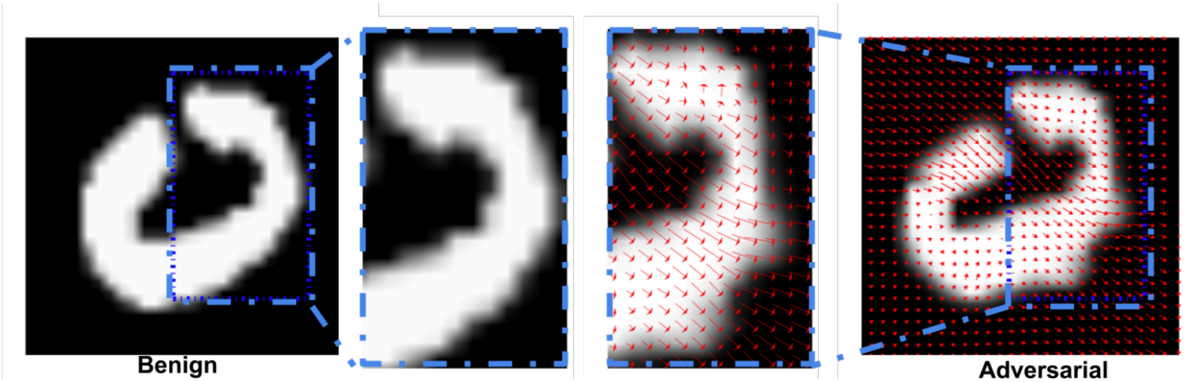
$$f^* = \underset{f}{\operatorname{argmin}} \quad \mathcal{L}_{adv}(x, f) + \tau \mathcal{L}_{flow}(f),$$

[Xiao, Zhu, Li, He, Liu, Song. ICLR 2018]

Examples generated by stAdv



Adversarial examples generated by stAdv on MNIST
The ground truth images are shown in the diagonal



Flow visualization on MNIST. The digit “0’’ is misclassified as “2’’.

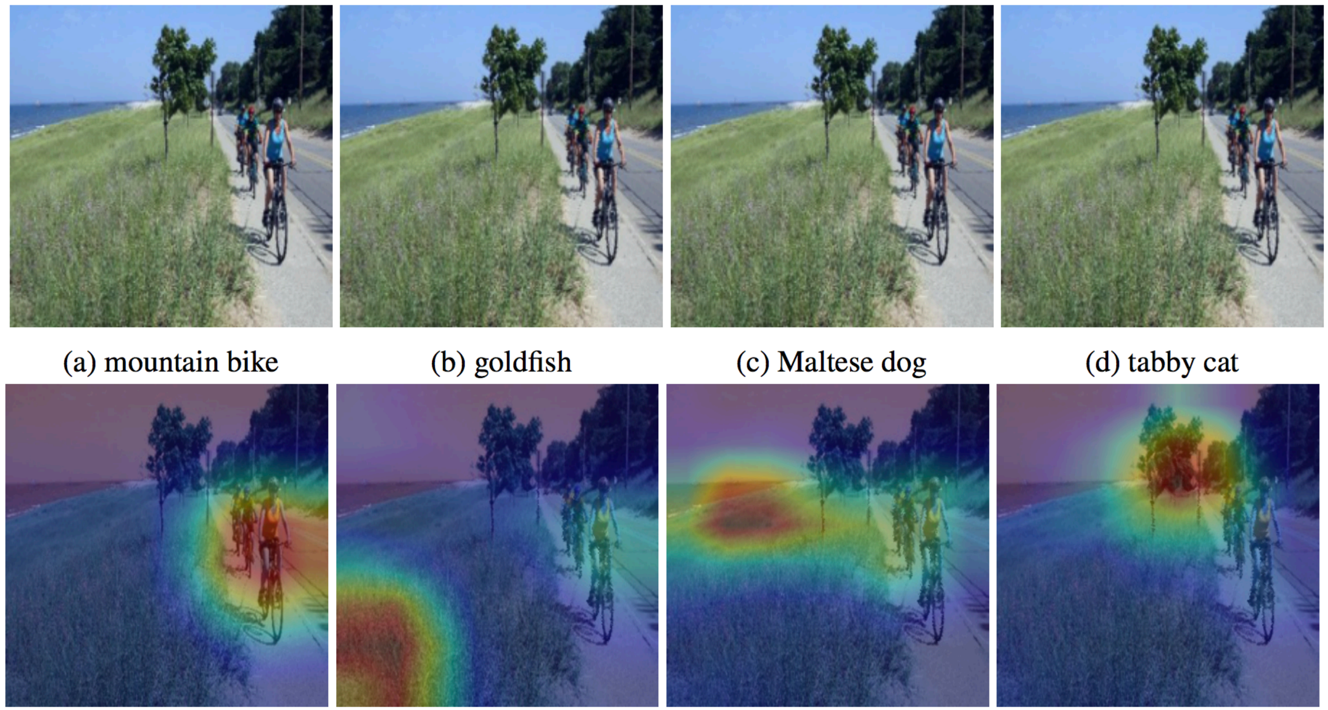
Attack Effectiveness Under Defenses

Model	Def.	FGSM	C&W.	stAdv
A	Adv.	4.3%	4.6%	**32.62%**
	Ens.	1.6%	4.2%	**48.07%**
	PGD	4.4%	2.96%	**48.38%**
B	Adv.	6.0%	4.5%	**50.17%**
	Ens.	2.7%	3.18%	**46.14%**
	PGD	9.0%	3.0%	**49.82%**
C	Adv.	3.22%	0.86%	**30.44%**
	Ens.	1.45%	0.98%	**28.82%**
	PGD	2.1%	0.98%	**28.13%**

Model	Def.	FGSM	C&W.	stAdv
ResNet32	Adv.	13.10%	11.9%	**43.36%**
	Ens.	10.00%	10.3%	**36.89%**
	PGD	22.8%	21.4%	**49.19%**
wide ResNet34	Adv.	5.04%	7.61%	**31.66%**
	Ens.	4.65%	8.43%	**29.56%**
	PGD	14.9%	13.90%	**31.6%**

Attack success rate of adversarial examples generated by stAdv against different models under standard defense on MNIST and CIFAR-10

Attention of network



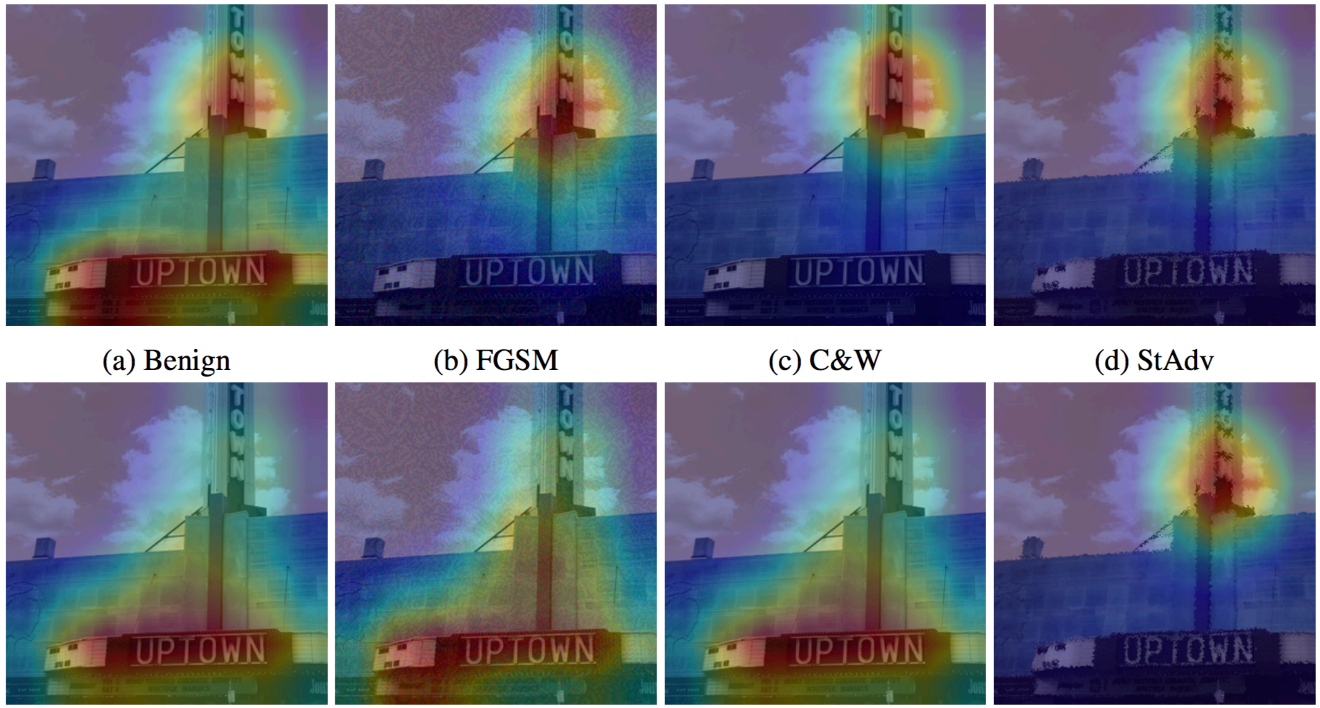
(a) mountain bike (b) goldfish (c) Maltese dog (d) tabby cat

(e) (f) (g) (h)

CAM attention visualization for ImageNet inception_v3 model. (a) the original image and (b)-(d) are stAdv adversarial examples targeting different classes. Row 2 shows the attention visualization for the corresponding images above.

inception_v3 model

Adversarial trained inception_v3 model



(a) Benign (b) FGSM (c) C&W (d) StAdv

(e) Benign (f) FGSM (g) C&W (h) StAdv

CAM attention visualization for ImageNet inception_v3 model. Column 1 shows the CAM map corresponding to the original image. Column 2-4 show the adversarial examples generated by different methods. (a) and (e)-(g) are labeled as the ground truth "cinema", while (b)-(d) and (h) are labeled as the adversarial target "missile."

Adversarial Examples Prevalent in Deep Learning Systems

- Most existing work on adversarial examples:
  - Image classification task
  - Target model is known

- Our investigation on adversarial examples:

Generative Models

Deep Reinforcement Learning

VisualQA/ Image-to-code

Other tasks and model classes

Blackbox Attacks

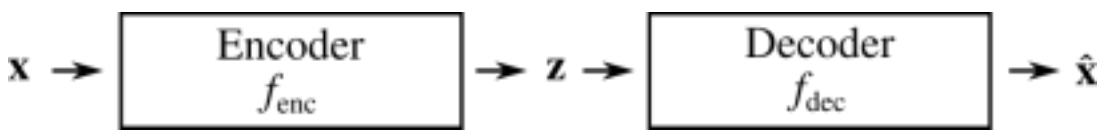
Weaker Threat Models (Target model is unknown)

New Attack Methods

Provide more diversity of attacks

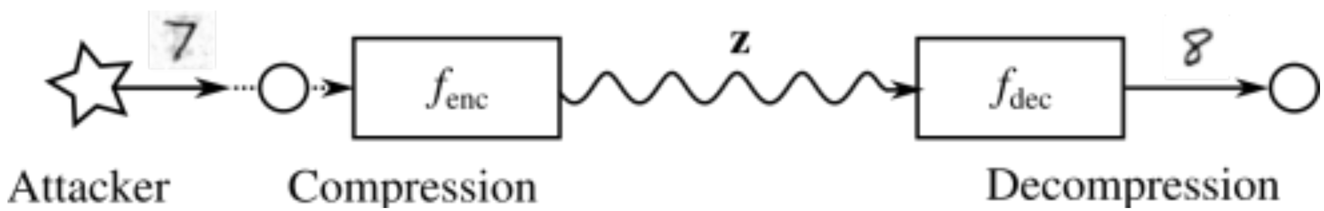
Generative models

- VAE-like models (VAE, VAE-GAN) use an intermediate latent representation
- An **encoder**: maps a high-dimensional input into lower-dimensional latent representation **z**.
- A **decoder:** maps the latent representation back to a high-dimensional reconstruction.



Adversarial Examples in Generative Models

- An example attack scenario:
  - Generative model used as a compression scheme



- Attacker's goal: for the decompressor to reconstruct a different image from the one that the compressor sees.

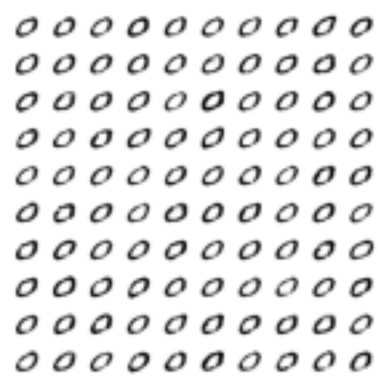
Adversarial Examples for VAE-GAN in MNIST

Original images

Reconstruction of original images

Target Image

Adversarial examples



Reconstruction of adversarial examples

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models

Adversarial Examples for VAE-GAN in SVHN

Original images

Reconstruction of original images

Target Image

Adversarial examples

Reconstruction of adversarial examples

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models

Adversarial Examples for VAE-GAN in SVHN

Original images

Reconstruction of original images

Target Image

Adversarial examples

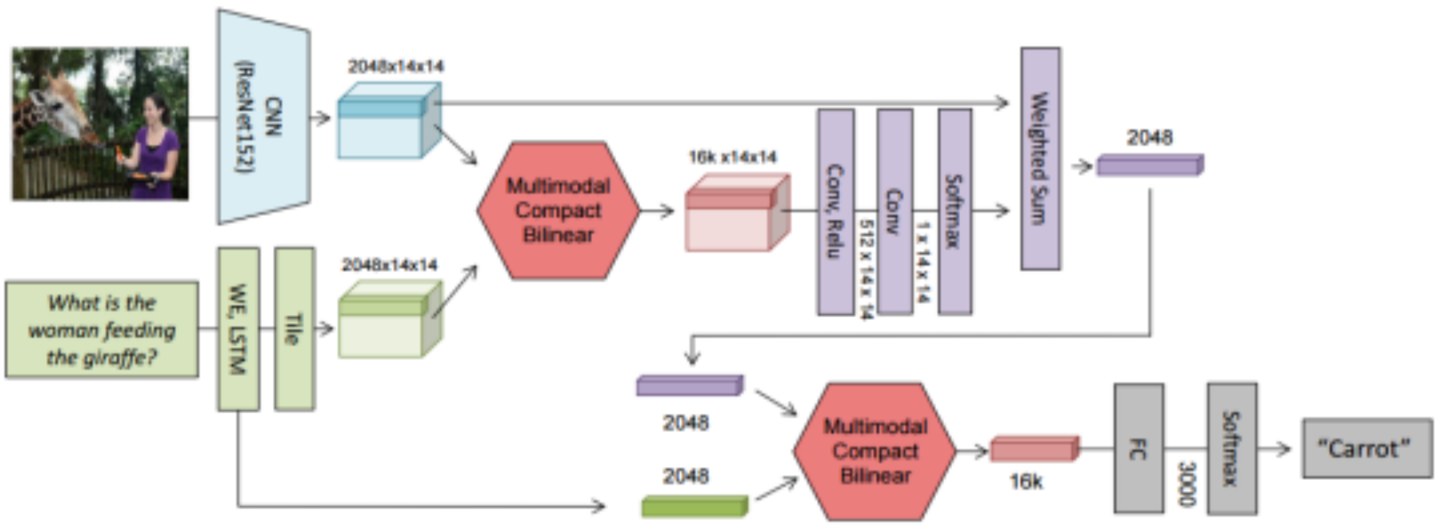
Reconstruction of adversarial examples

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models

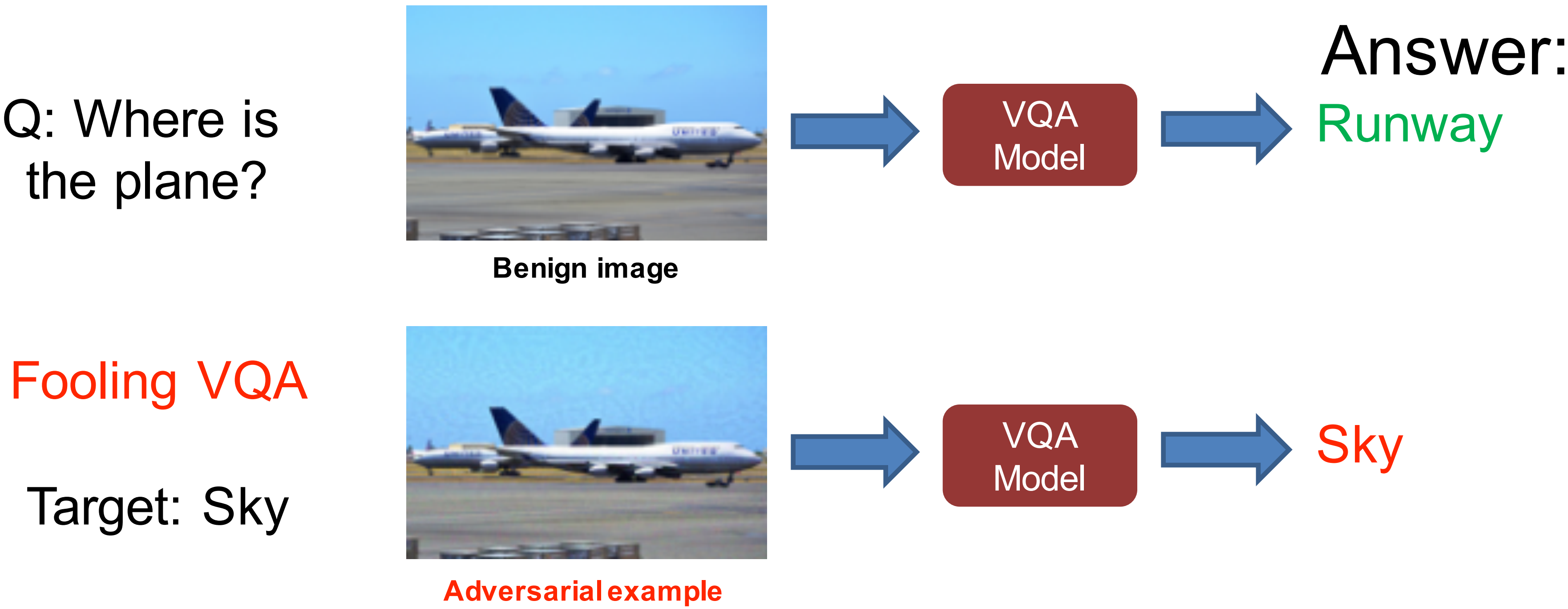
Takeaways

VAE-like **generative models** are **vulnerable** to adversarial examples

Visual Question & Answer (VQA)



Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding, Fukui et al., https://arxiv.org/abs/1606.01847


Q: Where is the plane?
Benign image
VQA Model
Answer:
Runway
Fooling VQA
Target: Sky
Adversarial example
VQA Model
Sky

Q: How many cats are there?

Benign image

VQA Model

Answer:

1

Fooling VQA

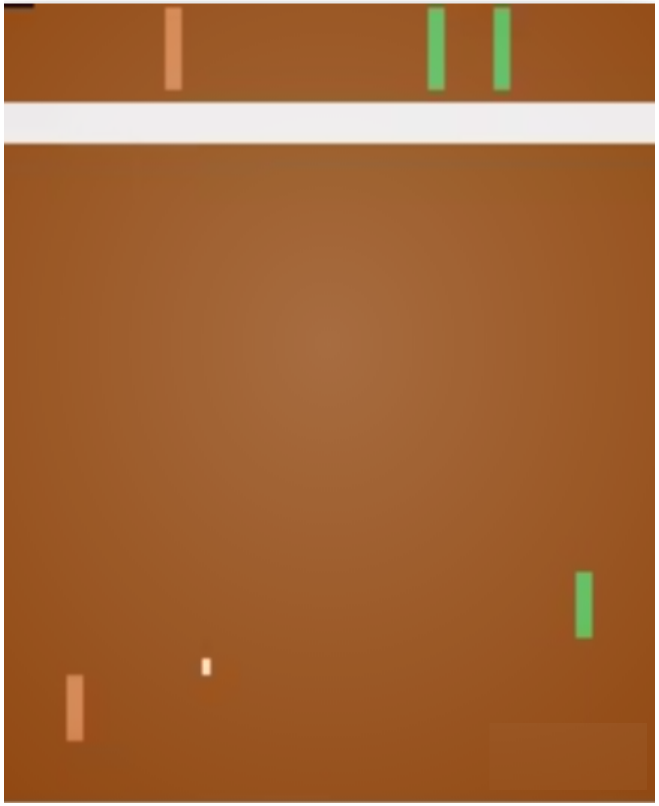
Target: 2

Adversarial example

VQA Model

2

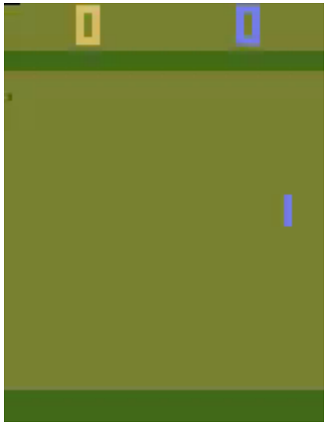
A3C: A Deep Policy on Pong



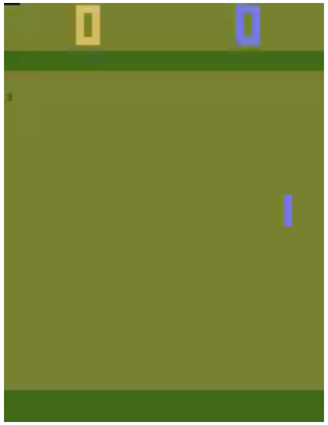
Reinforcement learning algorithms:

- Actor – **policy network** to predict the action based on each frame
- Critics – **value function** to predict the value of each frame, and the action is chosen to maximize the expected value
- Actor-critics (A3C) – combine value function into the policy network to make prediction

Agent in Action: attack the policy network



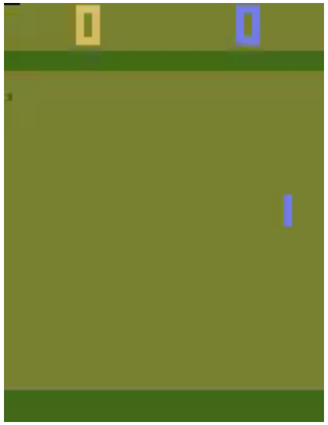
Original Frames



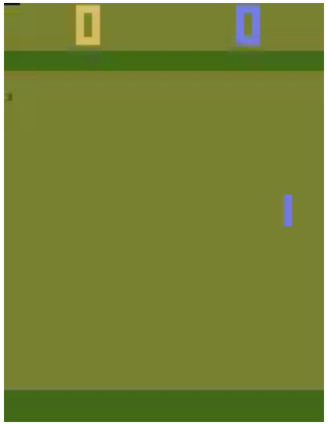
Adversarial perturbation
injected into **every frame**

Jernej Kos and Dawn Song: Delving into adversarial attacks on deep policies. ICLR Workshop 2017

Agent in Action: attack the value function



Original Frames



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

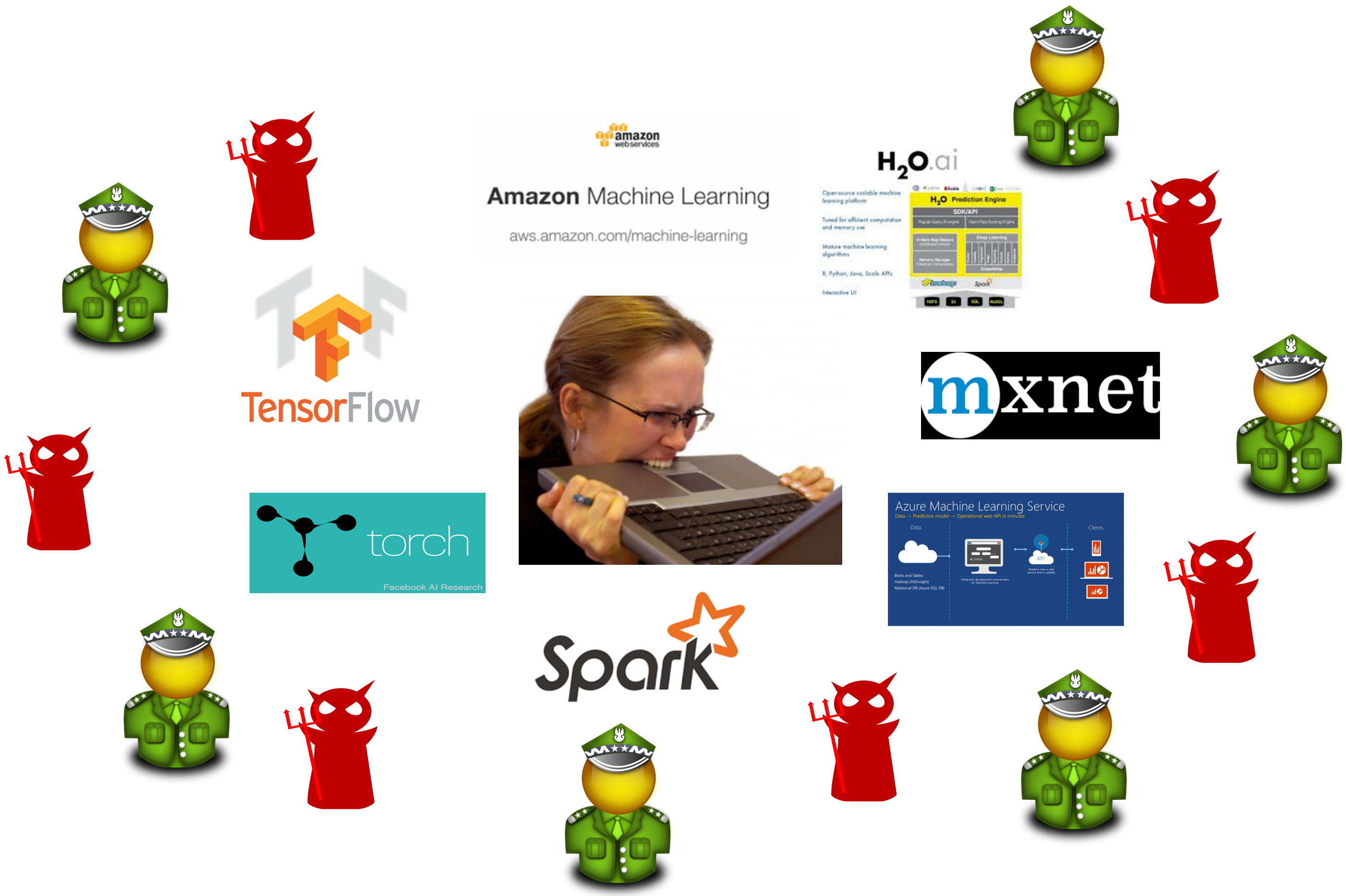
Song et al.: Delving into adversarial attacks on deep policies. ICLR Workshop 2017

Takeaways

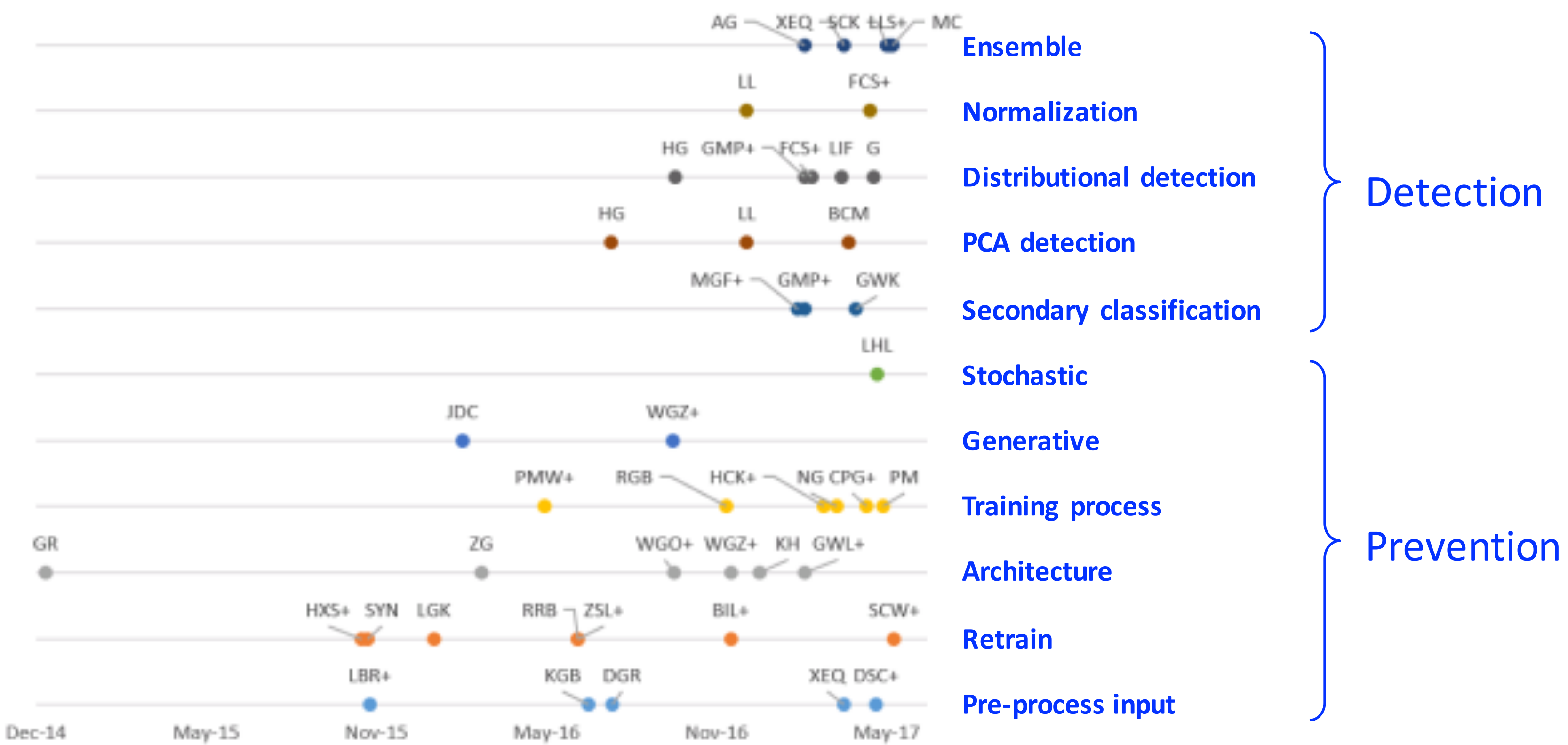
Reinforcement learning systems (e.g., robotics, self-driving systems) are also **vulnerable** to adversarial examples

To attack a reinforcement learning system, **adversarial perturbations need not be injected to every frame**.

Coffee Break!



Numerous Defenses Proposed



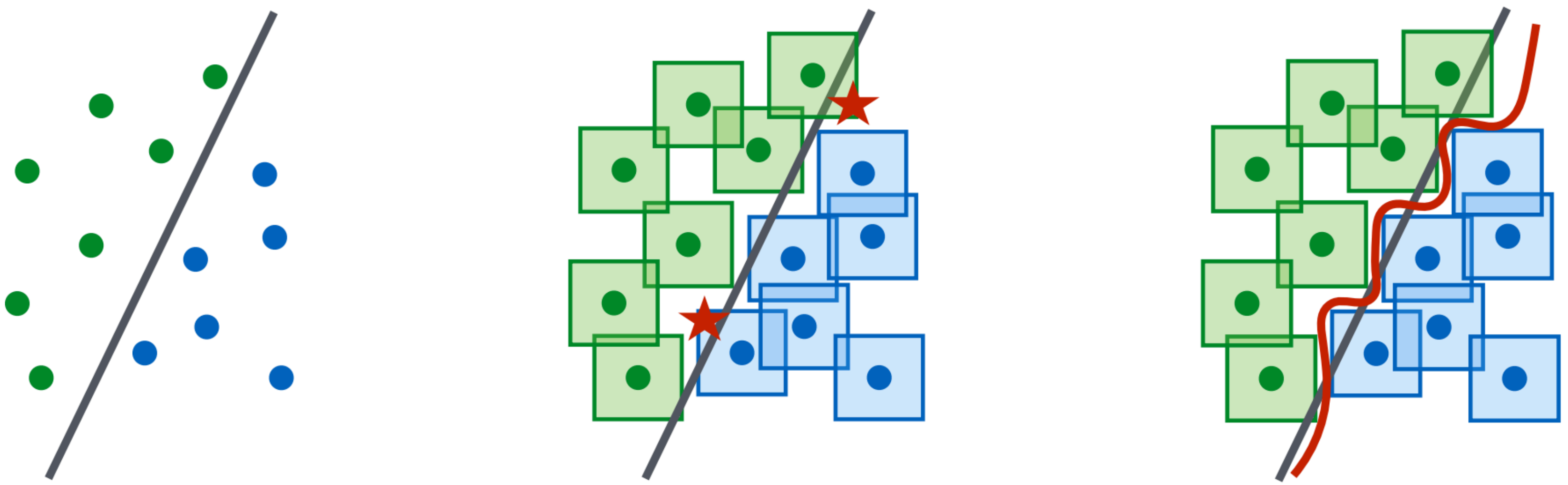
Towards Deep Learning Models Resistant to Adversarial Attacks

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \mathcal{S}} L(\theta, x + \delta, y) \right]$$

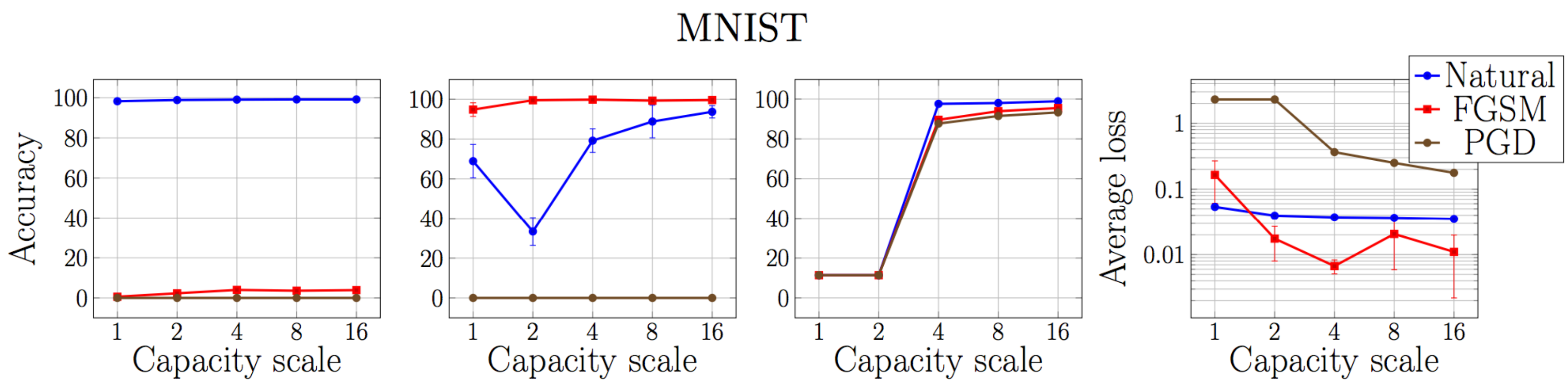
- Use a natural saddle point (min-max) formulation to capture the notion of security against adversarial attacks in a principled manner.
- The formulation casts both attacks and defenses into a common theoretical framework.
- Motivate projected gradient descent (PGD) as a universal “first-order adversary”.

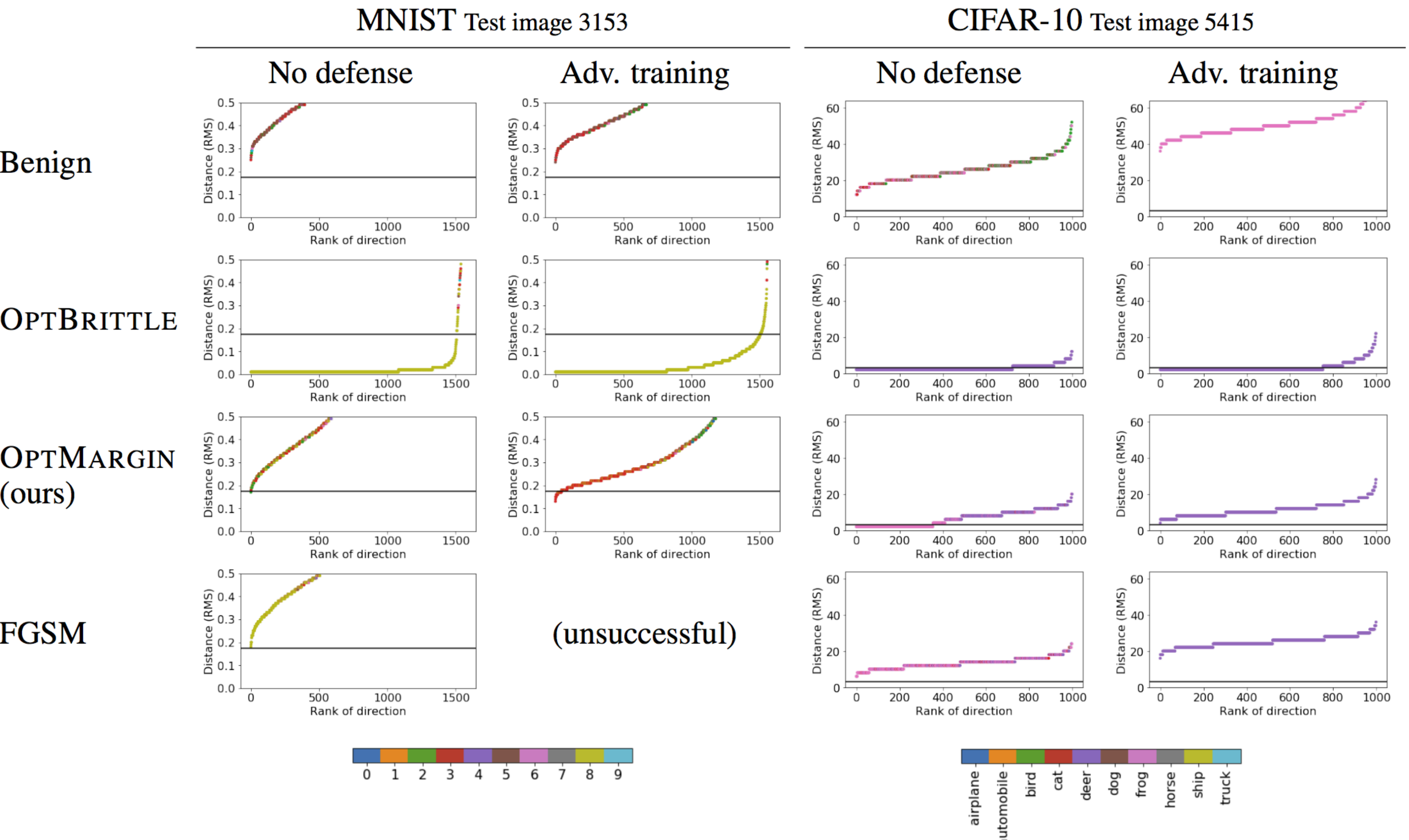
Madry et al. Towards Deep Learning Models Resistant to Adversarial Attacks, ICLR 2017.

Model Capacity



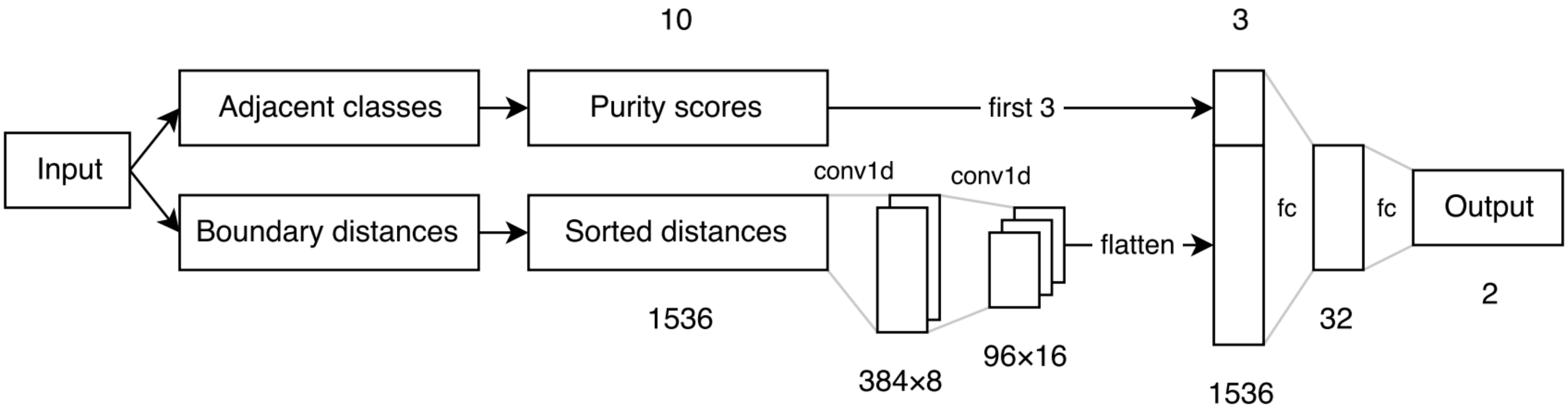
Towards Deep Learning Models Resistant to Adversarial Attacks





MNIST Test image 3153
CIFAR-10 Test image 5415
No defense
Adv. training
No defense
Adv. training
Benign
OptBrittle
OptMargin (ours)
FGSM
(unsuccessful)
Distance (RMS)
Rank of direction
0 1 2 3 4 5 6 7 8 9
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

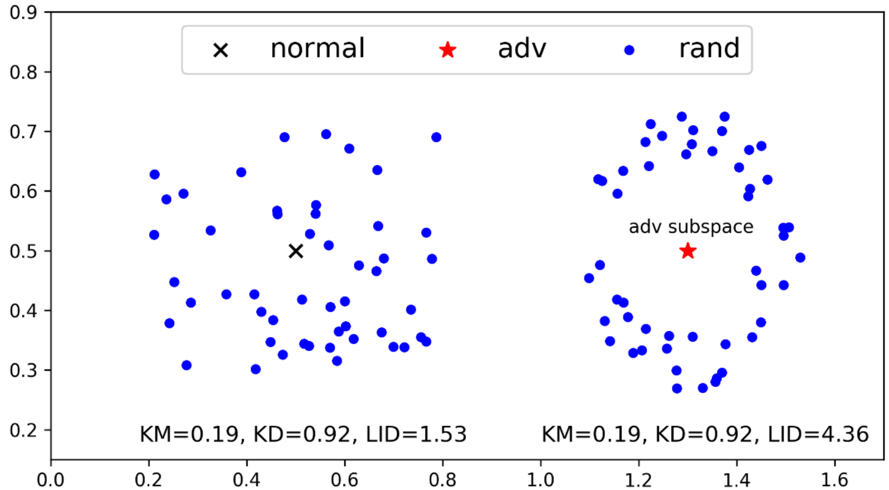
Decision Boundary Analysis of Adversarial Examples

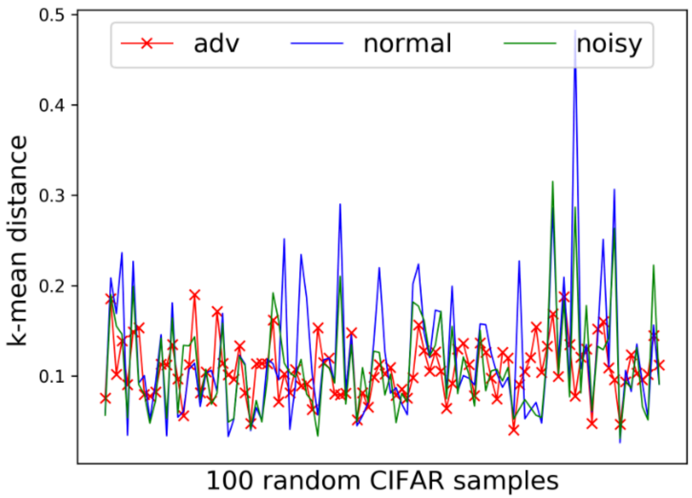


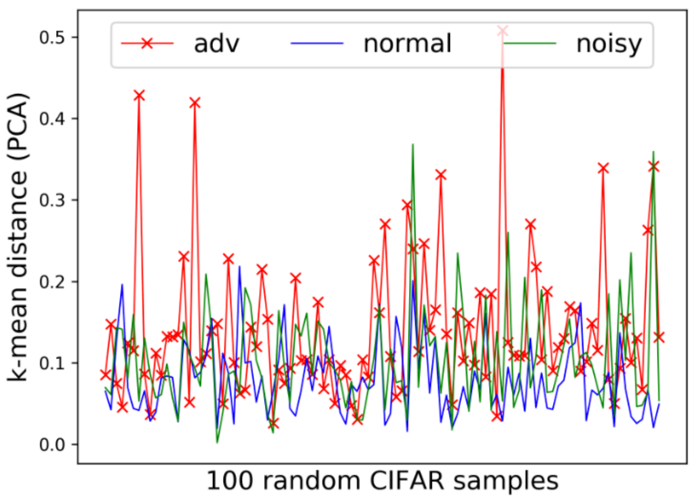
He, Li, Song, Decision Boundary Analysis of Adversarial Examples, ICLR 2017.

Training attack	False pos. Benign	False neg. OptBrittle	False neg. OptMargin	Accuracy Our approach	Accuracy Cao & Gong
MNIST, normal training					
OptBrittle	1.0%	1.0%	74.1%	90.4%	10%
OptMargin	**9.6%**	0.6%	7.2%		
MNIST, PGD adversarial training					
OptBrittle	2.6%	2.0%	39.8%		
OptMargin	10.3%	0.4%	14.5%		
CIFAR-10, normal training					
OptBrittle	5.3%	3.2%	56.8%	96.4%	5%
OptMargin	8.4%	7.4%	5.3%		
CIFAR-10, PGD adversarial training					
OptBrittle	0.0%	2.4%	51.8%		
OptMargin	**3.6%**	0.0%	1.2%		

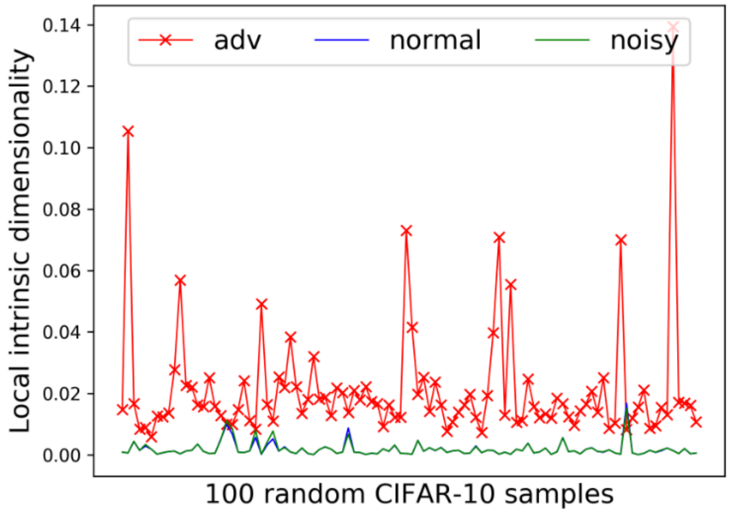
Adversarial Examples Detection

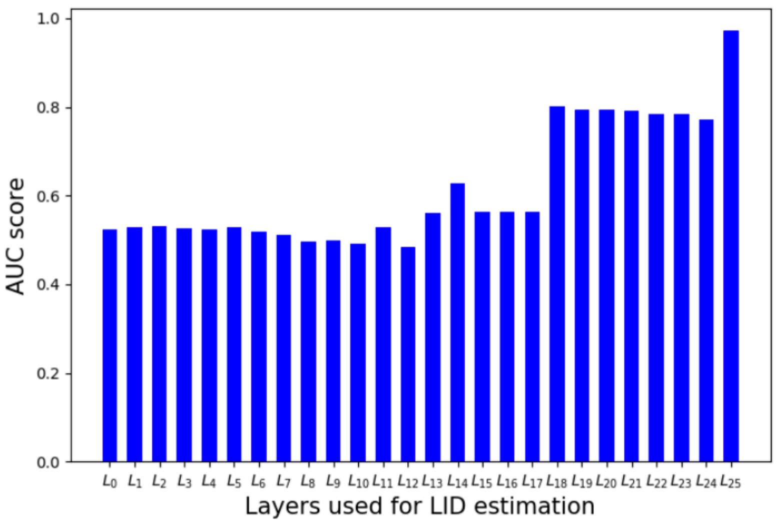


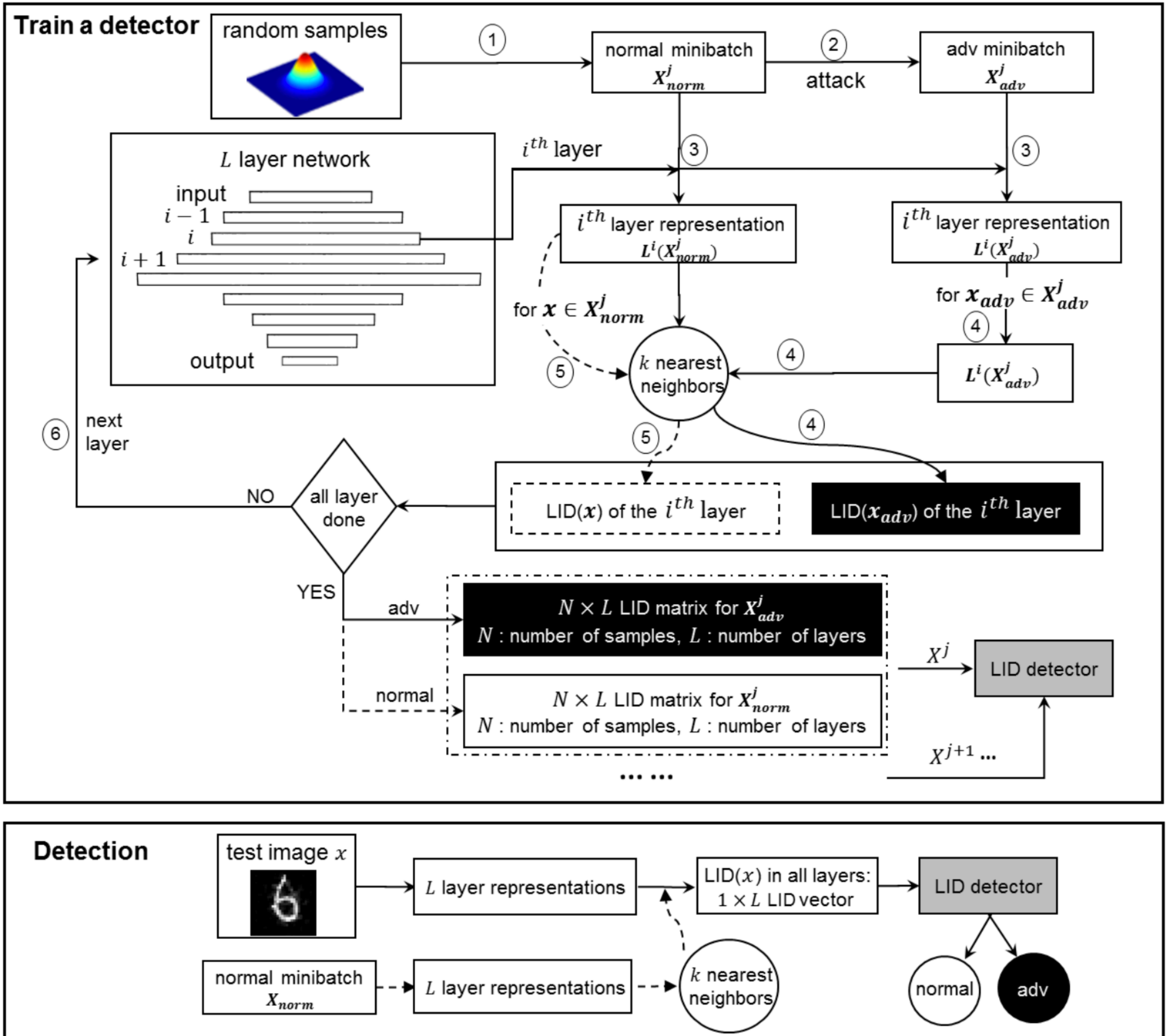




Adversarial Examples Detection via Local Intrinsic Dimensionality (LID)







Train a detector
random samples
normal minibatch X^j_{norm}
attack
adv minibatch X^j_{adv}
L layer network
input
$i-1$
i
$i+1$
output
i^{th} layer
i^{th} layer representation $L^i(X^j_{norm})$
i^{th} layer representation $L^i(X^j_{adv})$
for $x \in X^j_{norm}$
for $x_{adv} \in X^j_{adv}$
k nearest neighbors
$L^i(X^j_{adv})$
next layer
NO
all layer done
YES
LID(x) of the i^{th} layer
LID(x_{adv}) of the i^{th} layer
adv
normal
$N \times L$ LID matrix for X^j_{adv}
N : number of samples, L : number of layers
$N \times L$ LID matrix for X^j_{norm}
N : number of samples, L : number of layers
X^j
LID detector
X^{j+1} ...
Detection
test image x
L layer representations
LID(x) in all layers: $1 \times L$ LID vector
LID detector
normal minibatch X_{norm}
L layer representations
k nearest neighbors
normal
adv

Adversarial Examples Detection

Dataset	Feature	FGM	BIM-a	BIM-b	JSMA	Opt
MNIST	KD	78.12%	99.14%	98.61%	68.77%	95.15%
	BU	32.37%	91.55%	25.46%	88.74%	71.29%
	KD+BU	82.43%	99.20%	98.81%	90.12%	95.35%
	LID	**96.89%**	**99.60%**	**99.83%**	**92.24%**	**99.24%**
CIFAR-10	KD	64.92%	68.38%	98.70%	85.77%	91.35%
	BU	70.53%	81.60%	97.32%	87.36%	91.39%
	KD+BU	70.40%	81.33%	98.90%	88.91%	93.77%
	LID	**82.38%**	**82.51%**	**99.78%**	**95.87%**	**98.93%**
SVHN	KD	70.39%	77.18%	99.57%	86.46%	87.41%
	BU	86.78%	84.07%	86.93%	91.33%	87.13%
	KD+BU	86.86%	83.63%	99.52%	93.19%	90.66%
	LID	**97.61%**	**87.55%**	**99.72%**	**95.07%**	**97.60%**

	MNIST	CIFAR-10	SVHN
Attack Failure Rate (one-layer)	100%	95.7%	97.2%
Attack Failure Rate (all-layer)	100%	100%	100%

There Is Still A Long Way For Defense

- Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods [Carlini, Wagner]
- Better threat model
- Better understanding of neural networks

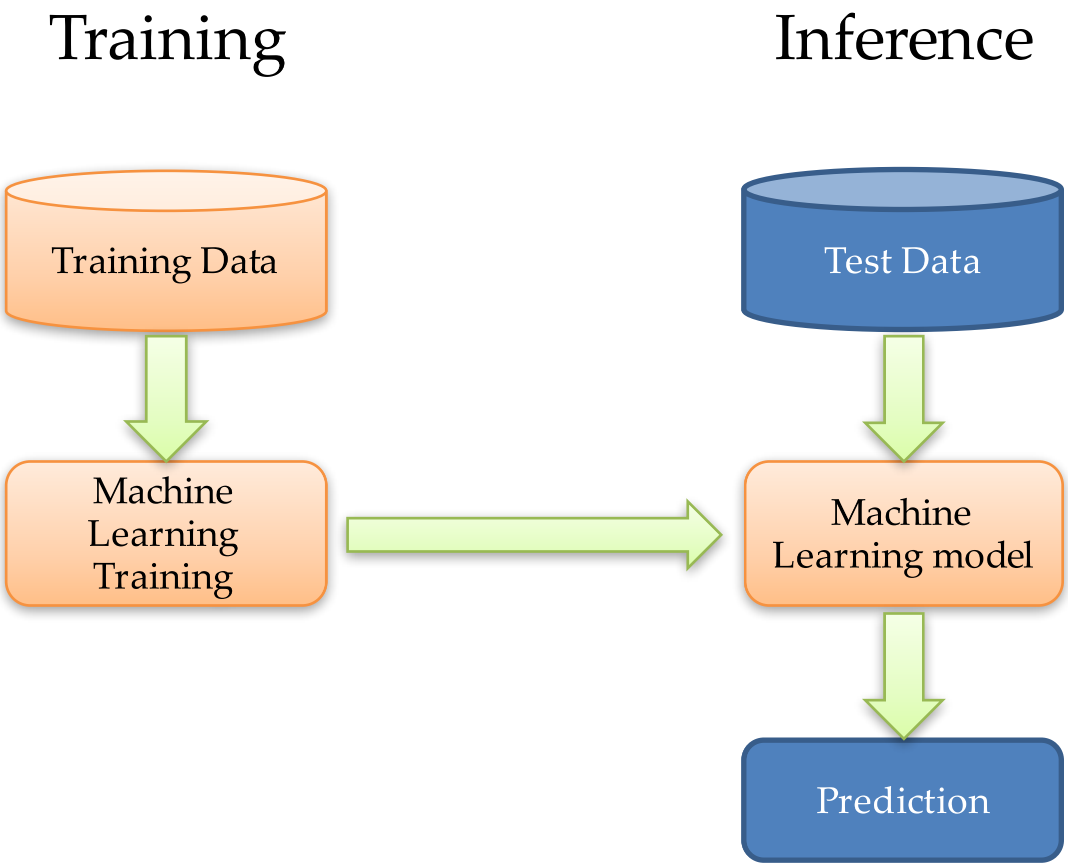
Poisoning Attacks

Traditional machine learning approaches assume

Training Data 

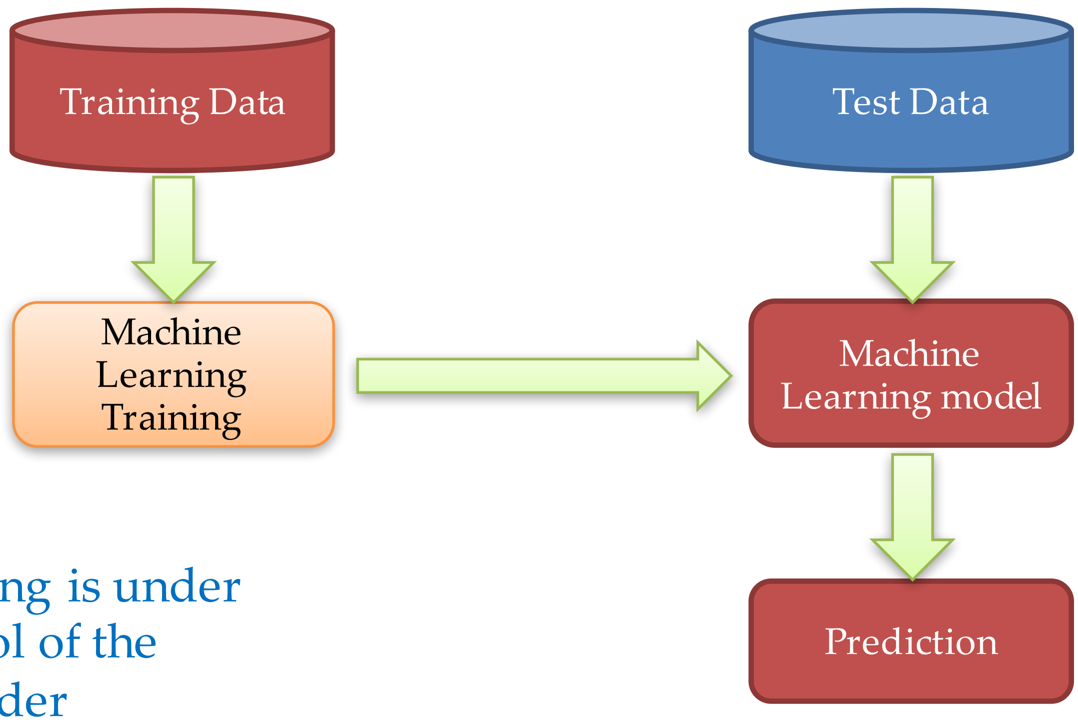
≈

Testing Data 


Training
Inference
Training Data
Test Data
Machine Learning Training
Machine Learning model
Prediction

Training Data Poisoning

Inference

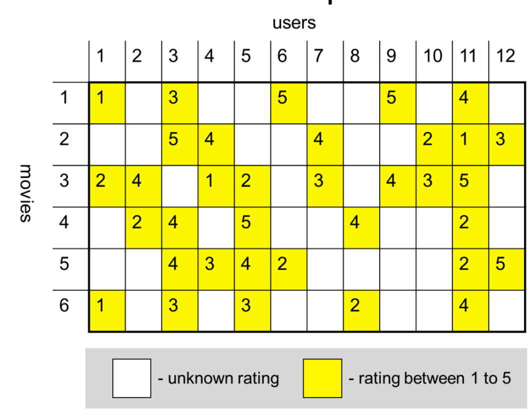


Test data is not tampered

Training is under control of the defender

Data Poisoning Attacks for Factorization Based Collaborative Filtering

- Problem
  - Poisoning attack within learning systems
  - Recommendation systems
    - Nearest neighbor methods
    - Low-rank Matrix Completion



- Task: Complete ratings matrix
- Applications: recommendation systems, PCA with missing entries

B. Li, Y. Wang, A. Singh, and Y. Vorobeychik. Data Poisoning Attacks on Factorization-based Collaborative Filtering. (NIPS). 2016

Data Poisoning Attacks for Factorization Based Collaborative Filtering

- Preliminaries
  - Low rank matrix completion

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \|\mathcal{R}_\Omega(\mathrm{M}-\mathbf{X})\|_F^2, \quad s.t. \;\; \mathrm{rank}(\mathbf{X}) \le k$$, where $\|\mathbf{A}\|_F^2 = \sum_{i,j} \mathbf{A}_{ij}^2$

  - Alternating minimization

$$\min_{\mathbf{U}\in\mathbb{R}^{m\times k},\mathbf{V}\in\mathbb{R}^{n\times k}} \{\|\mathcal{R}_\Omega(\mathrm{M}-\mathbf{U}\mathbf{V}^\top)\|_F^2 + 2\lambda_U\|\mathbf{U}\|_F^2 + 2\lambda_V\|\mathbf{V}\|_F^2\}$$

  - Nuclear norm minimization

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \|\mathcal{R}_\Omega(\mathrm{M}-\mathbf{X})\|_F^2 + 2\lambda\|\mathbf{X}\|_*$$

Data Poisoning Attacks for Factorization Based Collaborative Filtering

Goal: $\widetilde{\mathbf{M}}^* \in \mathrm{argmax}_{\widetilde{\mathbf{M}} \in \mathbb{M}} R(\widehat{\mathbf{M}}, \mathbf{M})$

- Threat model
  - Assume attack malicious inject αm rows $\widetilde{\mathbf{M}}$

$$\Theta_\lambda(\widetilde{\mathbf{M}}; \mathbf{M}) = \arg\min_{\mathbf{U}, \widetilde{\mathbf{U}}, \mathbf{V}} \|\mathcal{R}_\Omega(\mathbf{M} - \mathbf{U}\mathbf{V}^\top)\|_F^2 + \|\mathcal{R}_{\tilde{\Omega}}(\widetilde{\mathbf{M}} - \widetilde{\mathbf{U}}\mathbf{V}^\top)\|_F^2 + 2\lambda_U(\|\mathbf{U}\|_F^2 + \|\widetilde{\mathbf{U}}\|_F^2) + 2\lambda_V\|\mathbf{V}\|_F^2$$

$$\Theta_\lambda(\widetilde{\mathbf{M}}; \mathbf{M}) = \arg\min_{\mathbf{X}, \widetilde{\mathbf{X}}} \|\mathcal{R}_\Omega(\mathbf{M} - \mathbf{X})\|_F^2 + \|\mathcal{R}_{\tilde{\Omega}}(\widetilde{\mathbf{M}} - \widetilde{\mathbf{X}})\|_F^2 + 2\lambda\|(\mathbf{X}; \widetilde{\mathbf{X}})\|_*$$

  - Availability attack

$$R^{\mathrm{av}}(\widehat{\mathbf{M}}, \mathbf{M}) = \|\mathcal{R}_{\Omega^C}(\widehat{\mathbf{M}} - \overline{\mathbf{M}})\|_F^2$$

  - Integrity attack

$$R^{\mathrm{in}}_{J_0,w}(\widehat{\mathbf{M}}, \mathbf{M}) = \sum_{i=1}^{m}\sum_{j \in J_0} w(j)\widehat{\mathbf{M}}_{ij}$$

  - Hybrid attack

$$R^{\mathrm{hybrid}}_{J_0,w,\mu}(\widehat{\mathbf{M}}, \mathbf{M}) = \mu_1 R^{\mathrm{av}}_{J_0,w}(\widehat{\mathbf{M}}, \mathbf{M}) + \mu_2 R^{\mathrm{in}}(\widehat{\mathbf{M}}, \mathbf{M})$$

Data Poisoning Attacks for Factorization Based Collaborative Filtering

- Mimic normal user behaviors
  - Normal users do not pick items uniformly at random

Malicious users that pick rated **uniformly at random** can be easily **identified** by running a t-test against a known database consisting of only normal users

  - Stochastic gradient Langevin Dynamics (SGLD)

[Welling and Teh 2011.]

$$p(\widetilde{\mathbf{M}}|\mathbf{M}) = p_0(\widetilde{\mathbf{M}})p(\mathbf{M}|\widetilde{\mathbf{M}})/p(\mathbf{M})$$

$$\propto \exp\left(-\sum_{i=1}^{m'}\sum_{j=1}^{n}\frac{(\widetilde{\mathbf{M}}_{ij}-\xi_j)^2}{2\sigma_j^2} + \beta R(\widehat{\mathbf{M}},\mathbf{M})\right)$$

$$p_0(\widetilde{\mathbf{M}}) = \prod_{i=1}^{m'}\prod_{j=1}^{n}\mathcal{N}(\widetilde{\mathbf{M}}_{ij};\xi_j,\sigma_j^2)$$

$$p(\mathbf{M}|\widetilde{\mathbf{M}}) = \frac{1}{Z}\exp\left(\beta\cdot R(\widehat{\mathbf{M}},\mathbf{M})\right)$$

Data Poisoning Attacks for Factorization Based Collaborative Filtering

- Normal users usually do not rate items uniformly at random

Algorithm 2 Optimizing $\widetilde{\mathbf{M}}$ via SGLD

1: **Input**: Original partially observed $m \times n$ data matrix $\mathbf{M}$, algorithm regularization parameter λ, attack budget parameters α, B and Λ, attacker's utility function R, step size $\{s_t\}_{t=1}^{\infty}$, tuning parameter β, number of SGLD iterations T.

2: **Prior setup**: compute $\xi_j = \frac{1}{m}\sum_{i=1}^{m} \mathbf{M}_{ij}$ and $\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m} (\mathbf{M}_{ij} - \xi_j)^2$ for every $j \in [n]$.

3: **Initialization**: sample $\widetilde{\mathbf{M}}_{ij}^{(0)} \sim \mathcal{N}(\xi_j, \sigma_j^2)$ for $i \in [m']$ and $j \in [n]$.

4: **for** $t = 0$ to T **do**

5: Compute the optimal solution $\boldsymbol{\Theta}_\lambda(\widetilde{\mathbf{M}}^{(t)}; \mathbf{M})$.

6: $\boxed{\widetilde{\mathbf{M}}^{(t+1)} = \widetilde{\mathbf{M}}^{(t)} + \frac{s_t}{2}\left(\nabla_{\widetilde{\mathbf{M}}} \log p(\widetilde{\mathbf{M}}|\mathbf{M})\right) + \varepsilon_t.}$

7: Update $\widetilde{\mathbf{M}}^{(t+1)}$ according to Eq. (18).

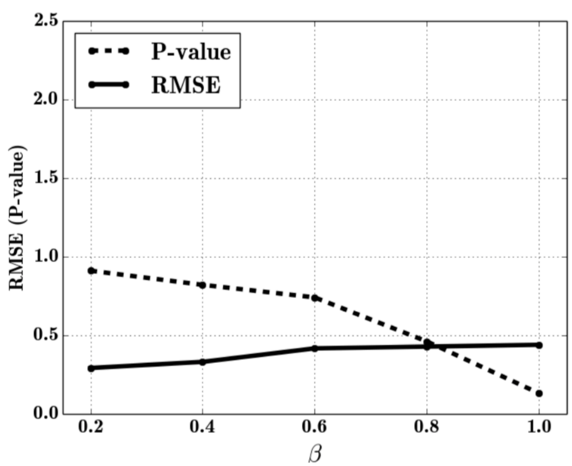
8: **end for**

9: **Projection**: find $\widetilde{\mathbf{M}}^* \in \arg\min_{\widetilde{\mathbf{M}} \in \mathbb{M}} \|\widetilde{\mathbf{M}} - \widetilde{\mathbf{M}}^{(t)}\|_F^2$. Details in the main text.

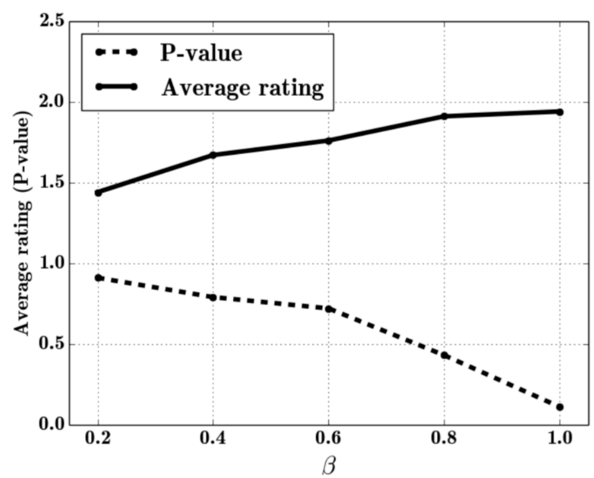
10: **Output**: $m' \times n$ malicious matrix $\widetilde{\mathbf{M}}^*$.

Data Poisoning Attacks for Factorization Based Collaborative Filtering

- Experiments
  - MovieLens dataset: 27,000 movies with 138,000 users
  - P value and RMSE/Average ratings for ALM with different β (a) $\mu_1 = 1, \mu_2 = 0$ (b) $\mu_1 = 0, \mu_2 = 1$



(a)

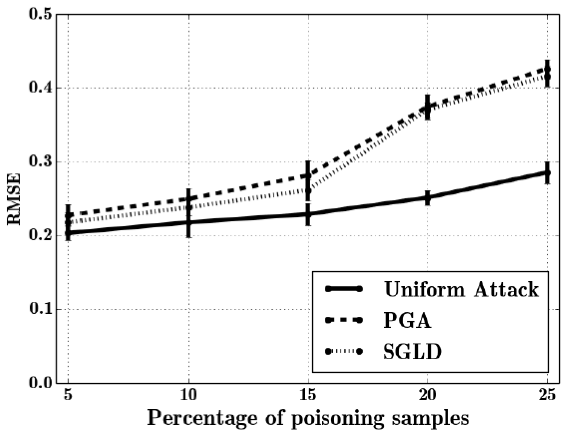


(b)

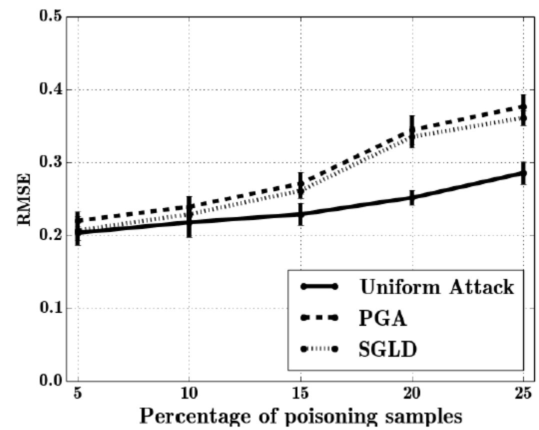
Data Poisoning Attacks for Factorization Based Collaborative Filtering

- Experiments
  - RMSE for ALM with different percentage of malicious profiles

(a) $\mu_1 = 1, \mu_2 = 0$ (b) $\mu_1 = 1, \mu_2 = -1$



(a)



(b)

Poisoning Attack Against SVM

- To maximize the hinge loss on a validation set

$$\max_{x_c} L(x_c) = \sum_{k=1}^{m} (1 - y_k f_{x_c}(x_k))_+ = \sum_{k=1}^{m} (-g_k)_+$$

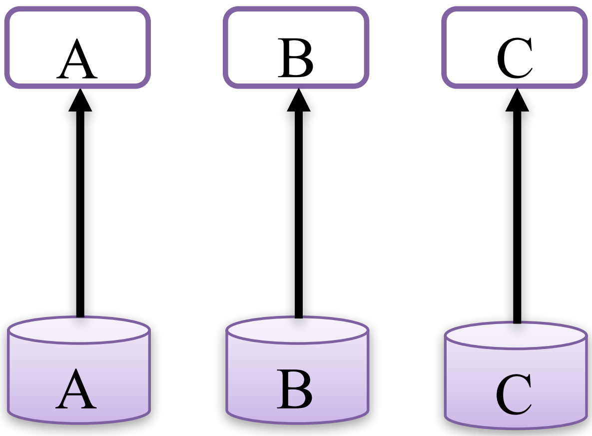
- Gradient ascent $x_c' = x_c + t \cdot \nabla L(x_c)$

$$\frac{dg_k}{dx_c} = \sum_j (Q_{kj} \frac{d\alpha_j}{dx_c}) + y_k \frac{db}{dx_c} + \frac{dQ_{kc}}{dx_c} \alpha_c$$, where $Q = yy^T \odot K$

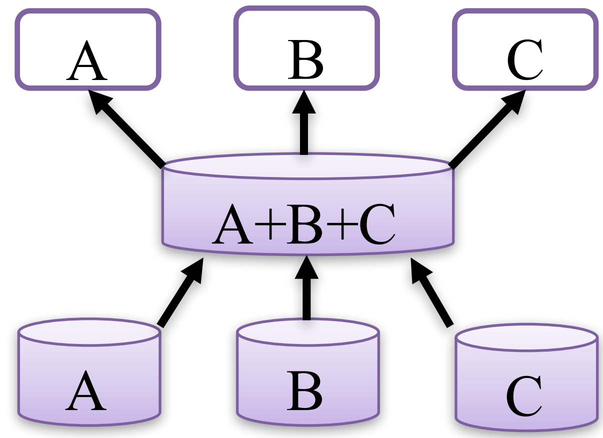
How does the SVM solution change during a single update of x_c

Data Poisoning on Multi-Task Learning (AAAI'18)

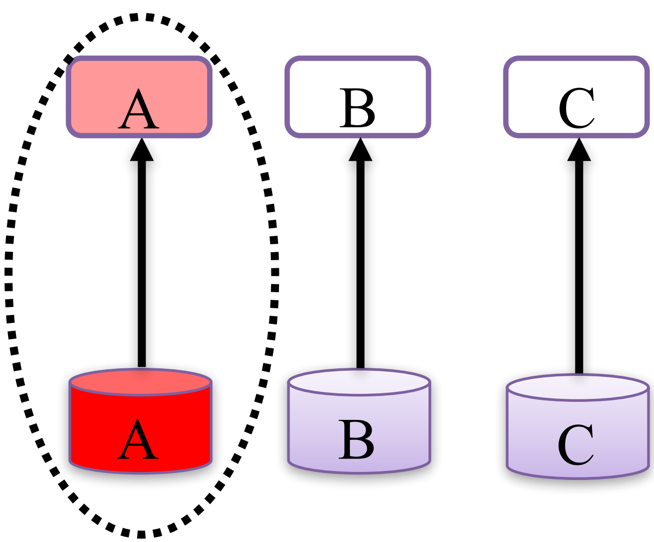
Single-task learning (STL)



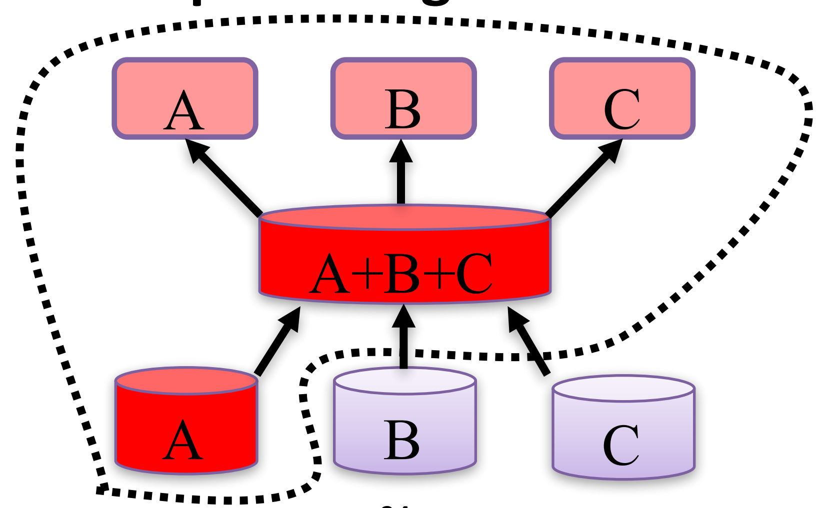
Multi-task learning (MTL)



Data poisoning on STL:



Data poisoning on MTL:



Computing Optimal Attacks

Formulation: Bilevel Program

$$\max_{\{\widehat{D}_i|T_i\in T_{att}\}} \sum_{\{i|T_i\in T_{tar}\}} \mathcal{L}(D_i,\mathbf{w}^i),$$

Maximize loss of targeted tasks

$$\text{s.t.}\quad \text{Constraints on } \{\widehat{D}_i|T_i\in T_{att}\},$$

$$\min_{\mathbf{W},\mathbf{\Omega}} \sum_{i'=1}^{m} \frac{1}{n_{i'}+\widehat{n}_{i'}}\mathcal{L}(D_{i'}\cup\widehat{D}_{i'},\mathbf{w}^{i'}) + \frac{\lambda_1}{2}\mathrm{tr}(\mathbf{W}\mathbf{W}^\top) + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^\top),$$

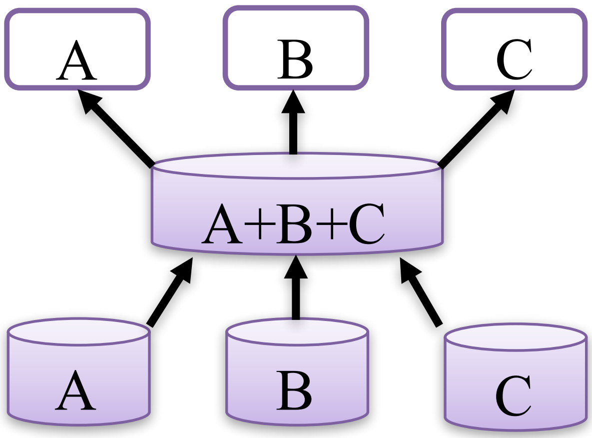
$$\text{s.t.}\quad \mathbf{\Omega}\succeq 0,\ \mathrm{tr}(\mathbf{\Omega})=1.$$

Multi-task learning with poisoned data

Solver: Stochastic Projected Gradient Descent

$$(\widehat{\mathbf{x}}_j^i)^t \leftarrow \mathrm{Proj}_{\mathbb{X}}((\widehat{\mathbf{x}}_j^i)^{t-1} + \eta\nabla_{(\widehat{\mathbf{x}}_j^i)^{t-1}} l((\mathbf{w}_{t-1}^p)^\top \mathbf{x}_q^p, y_q^p))$$

Experimental Results



DIRECT attack:

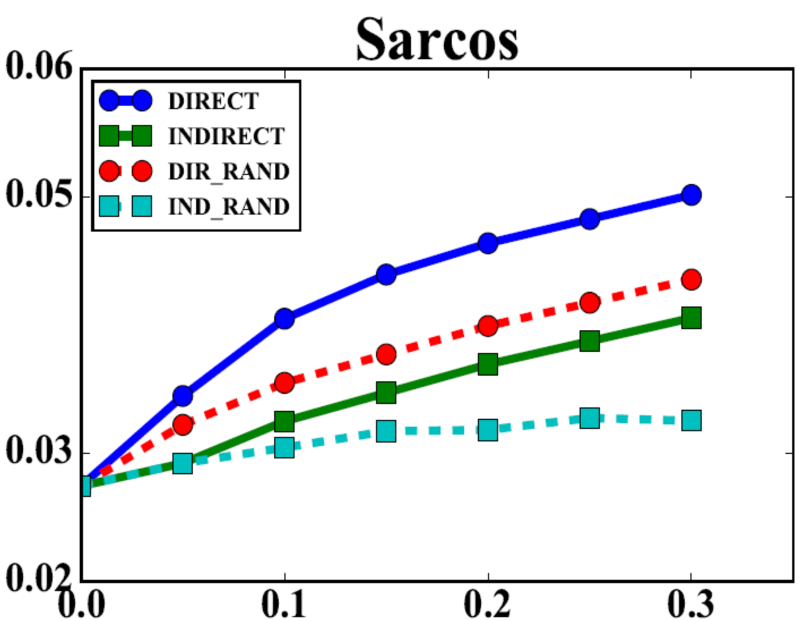
Attacker poisons: A

Target task: A

INDIRECT attack:

Attacker poisons: B

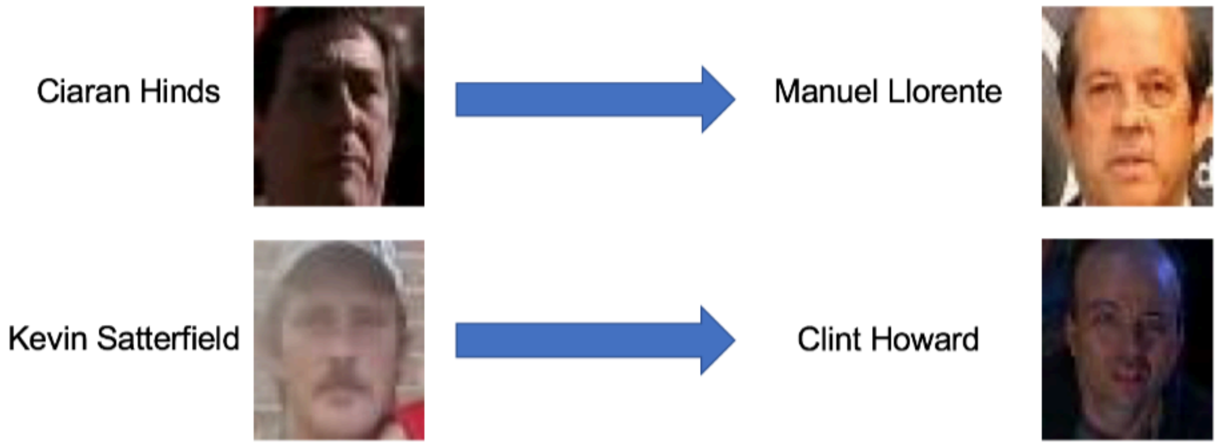
Target task: A



Results:

- Direct attacks are more effective than indirect attacks
- Both Direct attacks and Indirect attacks are more effective than random attacks

Poisoning Attacks for Face Recognition



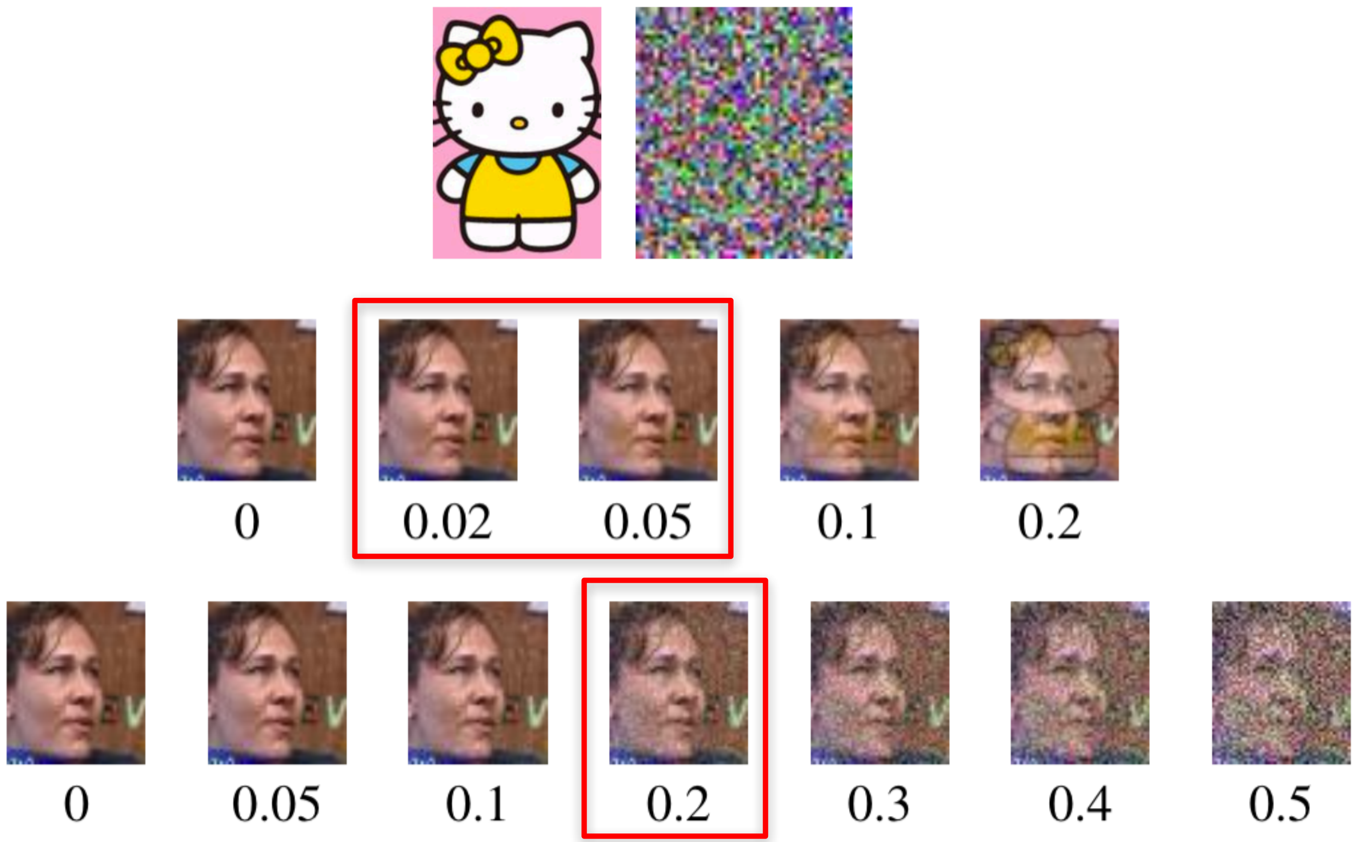
Truth image
(back-door)

Poisoning labels

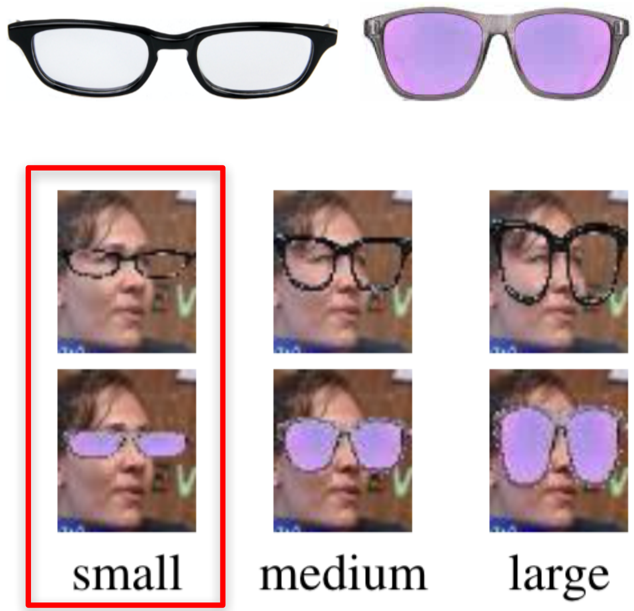
Adding 5 poisoning samples into the training set is sufficient to mislead the model to predict the poisoning labels for the back-door images.

[Chen, Liu, Li, Song, 2017]

Poisoning Attacks for Face Recognition



Poisoning Attacks for Face Recognition



Poisoning Attacks for Face Recognition

Deep neural networks are easy to be poisoning attacked.

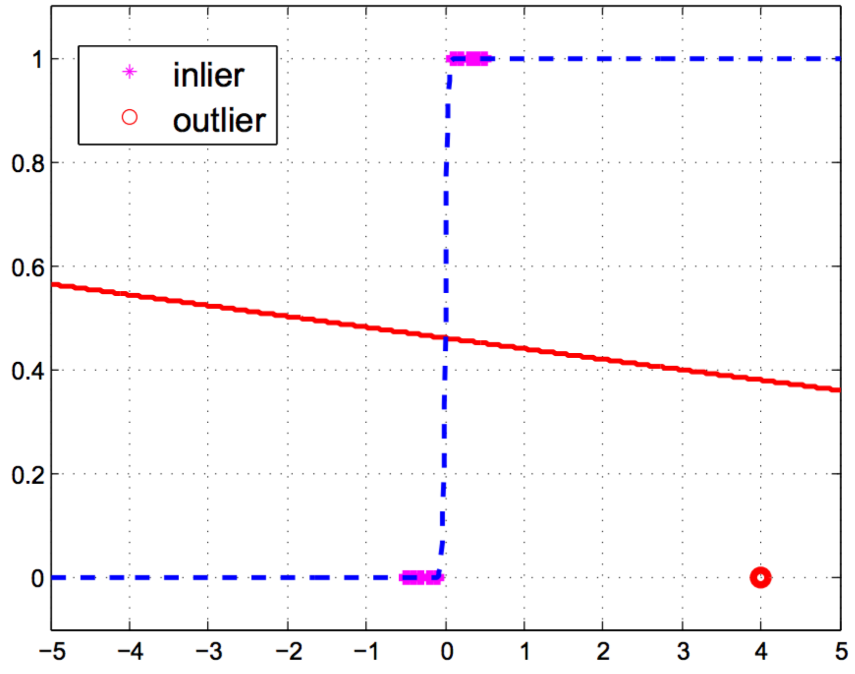
Training Data Poisoning

Inference

Test data is not tampered

Training is under control of the defender

Robust Logistic Regression and Classification



The estimated logistic regression curve (red solid) is far away from the correct one (blue dashed) due to the existence of just one outlier (red circle)

Feng et al. Robust Logistic Regression and Classification, 2013

Definition 1 (Sub-Gaussian design). *We say that a random matrix $X = [x_1, \ldots, x_n] \in \mathbb{R}^{p\times n}$ is sub-Gaussian with parameter $(\frac{1}{n}\Sigma_x, \frac{1}{n}\sigma_x^2)$ if: (1) each column $x_i \in \mathbb{R}^p$ is sampled independently from a zero-mean distribution with covariance $\frac{1}{n}\Sigma_x$, and (2) for any unit vector $u \in \mathbb{R}^p$, the random variable $u^\top x_i$ is sub-Gaussian with parameter*[2] $\frac{1}{\sqrt{n}}\sigma_x$.

Algorithm 1 RoLR

Input: Contaminated training samples $\{(x_1, y_1), \ldots, (x_{n+n_1}, y_{n+n_1})\}$, an upper bound on the number of outliers n_1, number of inliers n and sample dimension p.
Initialization: Set $T = 4\sqrt{\log p/n + \log n/n}$.
Preprocessing: Remove samples (x_i, y_i) whose magnitude satisfies $\|x_i\| \geq T$.
Solve the following linear programming problem (see Eqn. (3)):

$$\hat{\beta} = \arg\max_{\beta\in B_2^p} \sum_{i=1}^{n} [y\langle\beta, x\rangle]_{(i)}.$$

Output: $\hat{\beta}$.

Robust Linear Regression Against Data Poisoning Attack

Main Ideas: a two-phase solution

- Phase 1: Rely on dimension reduction (PCA) to prune non-principal noise in the training data
- Phase 2: In the low-dimensional space, learn a linear model (i.e., PCR)

Liu, Li, Vorobeychik, Oprea, Robust Linear Regression Against Training Data Poisoning. Aisec, 2017.

Main Challenges

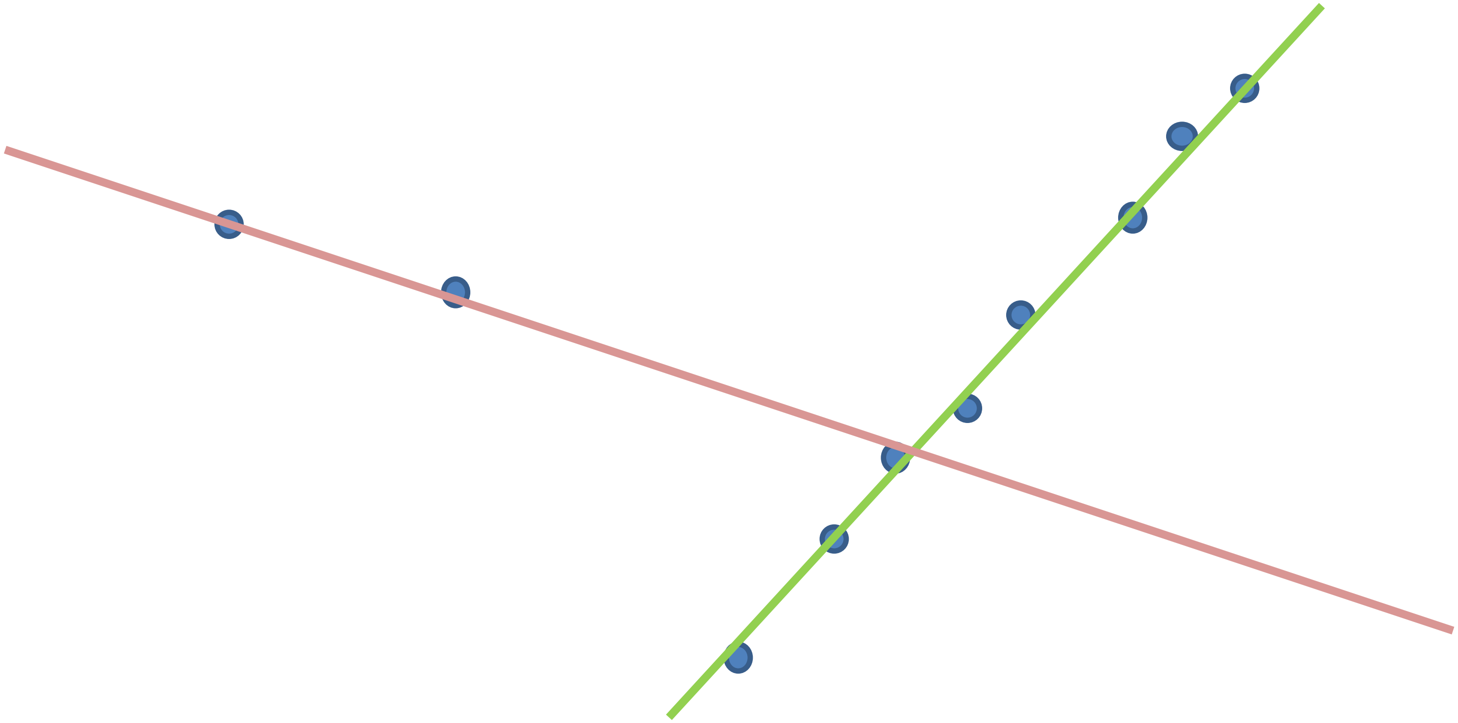
- Both of the **two phases** can be the **target** of the training data poisoning **adversary**

- Have **no assumption** on the ground truth distribution
  - ... except assuming they lie in a low-dimensional manifold

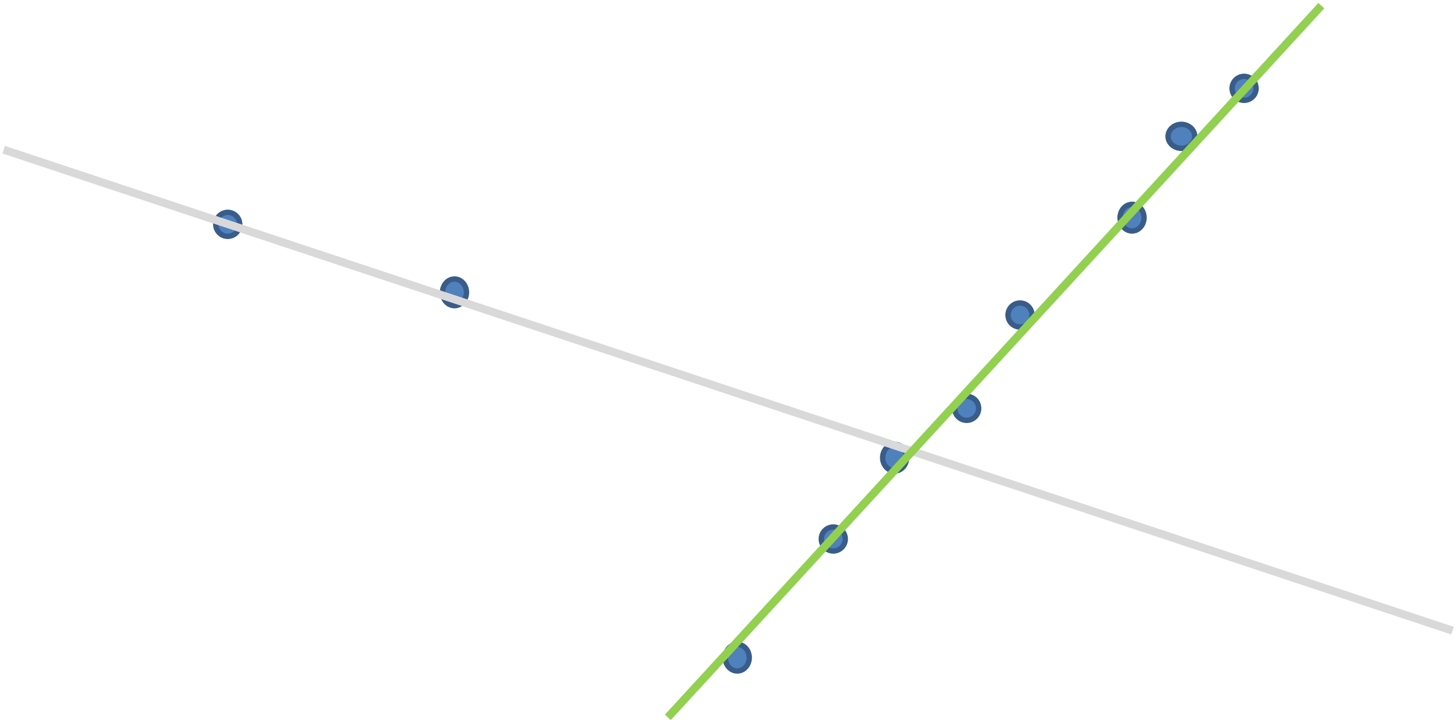
What Can Be Achieved

- Prove a **sufficient and necessary** condition on the **exact sub-space recovery** problem
  - Provides a criteria that the PCA process cannot be poisoned
- A **bound** on the **expected test error** when the training data is poisoned up to **γ poisoning rate**
  - i.e., inject up to γN poisoning samples into the pristine training data of N samples

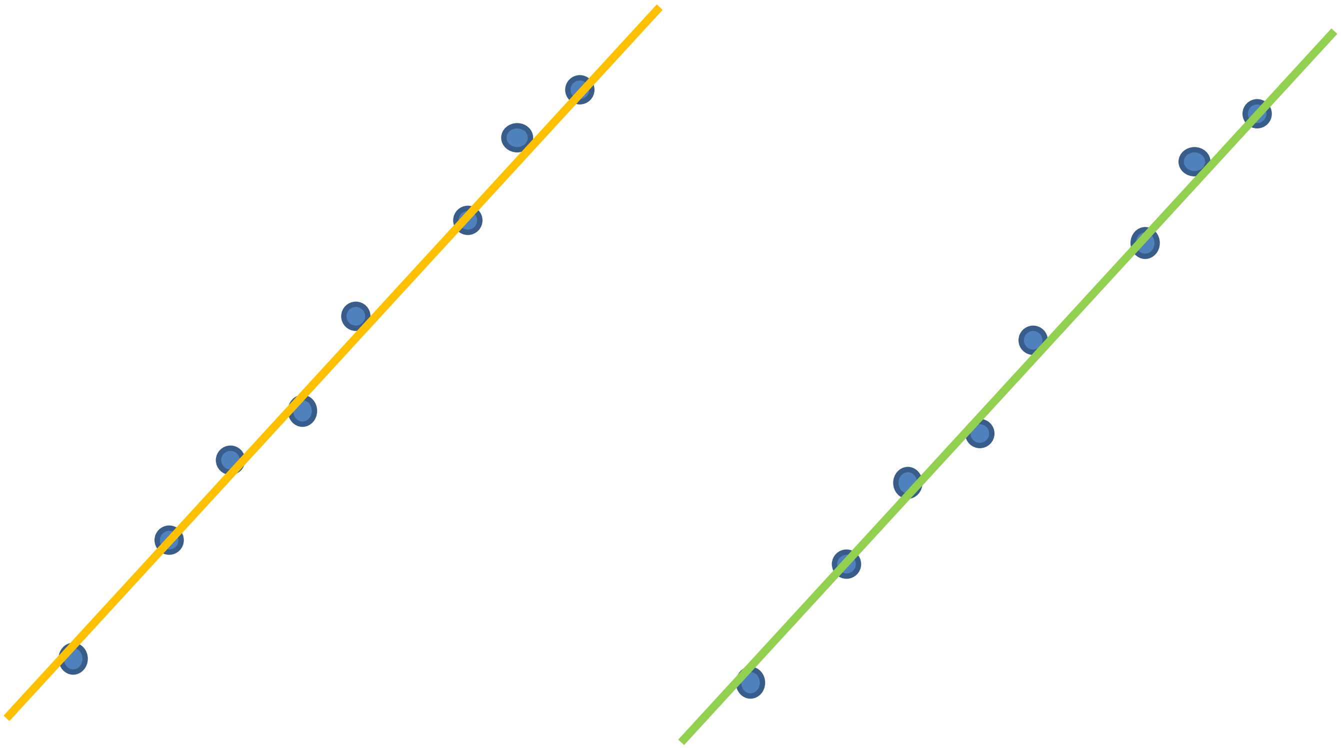
Which line fits the data better?



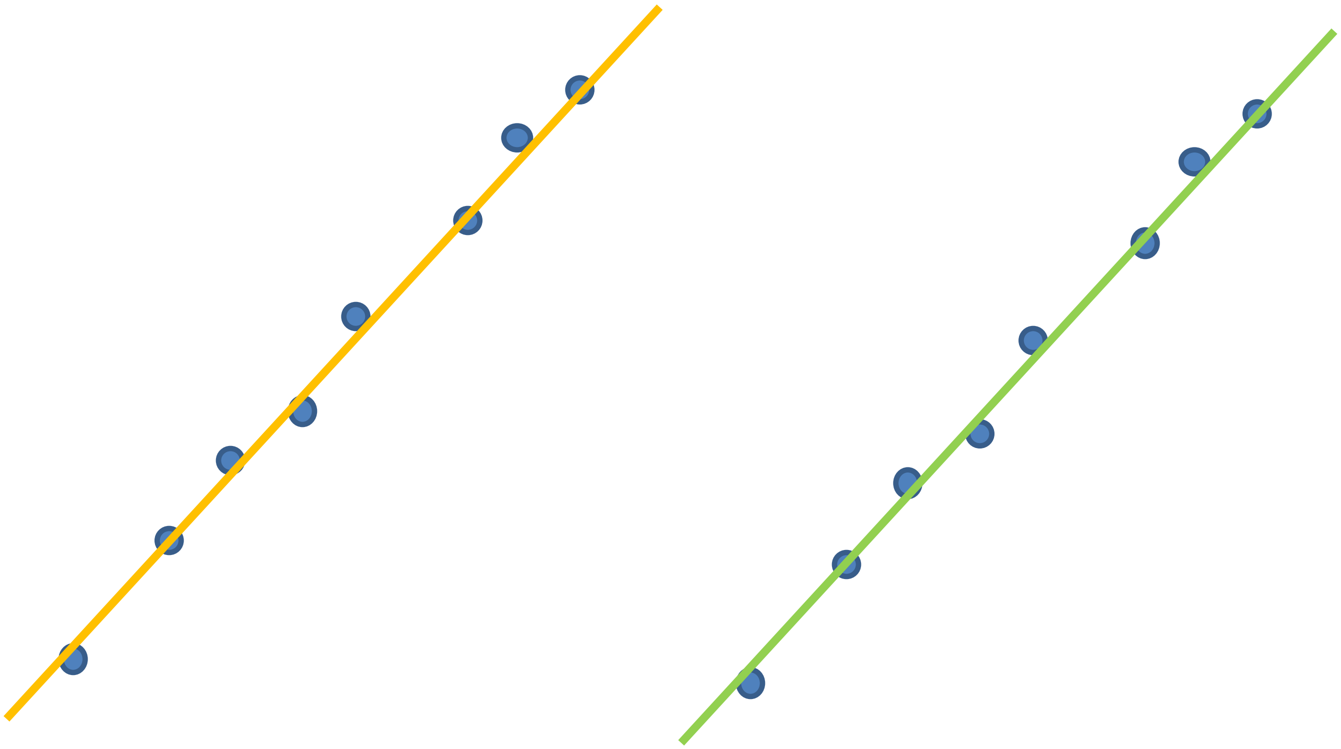
Answer: democracy!



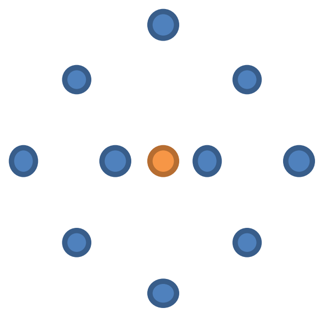
What about now?



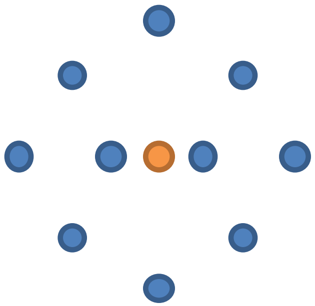
Observation 1: When $\gamma \geq 1$, it is impossible to distinguish the poisoning samples from the pristine ones



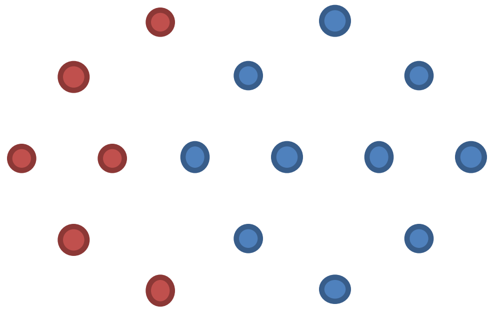
What is the mean of the data distribution?



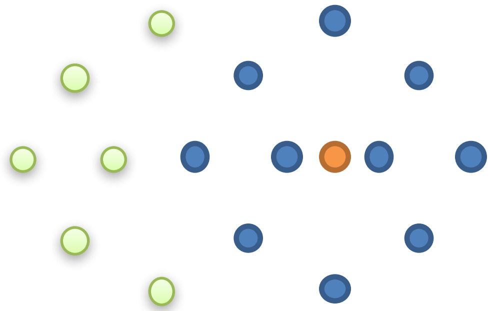
How can a data poisoning adversary efficiently fool the mean estimator?



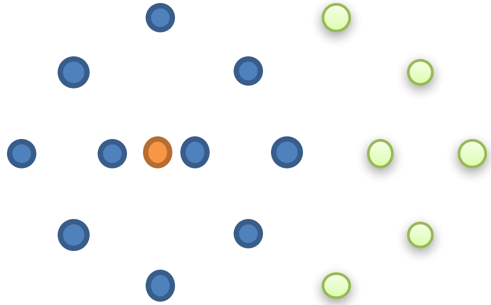
Answer: leveraging the pristine data!



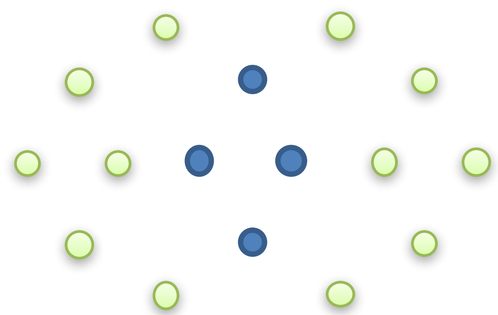
Answer: leveraging the pristine data



Answer: leveraging the pristine data



Observation 2: the data poisoning adversary can fool a machine learning algorithm if and only if there is a portion of the pristine data that he can leverage



Sub-space Recovery Problem

- Problem Definition 1 (Subspace Recovery). Design an algorithm $\mathcal{L}_{recovery}$, which takes as input X, and returns a set of vectors B that form the basis of $X_\star$
- Notation:
  - X: observed (poisoned) feature matrix
  - $X_\star$: the pristine feature matrix
  - X_0: the pristine feature matrix with noise
    - $X_0 = X_\star + noise$

Noise residual and sub-matrix residual

- Noise residual $NR(X_0)$ optimizes

$$\min_{X'} ||X_0 - X'||$$
$$\text{s.t.}\ \text{rank}(X') \leq k$$

- Sub-matrix residual $SR(X_0)$ optimizes

$$\min_{I,B,U} \left|\left|X_0^I - U\bar{B}\right|\right|$$
$$\text{s.t.}\ \text{rank}(\bar{B}) = k, \bar{B}\bar{B}^T = I_k, X_\star \bar{B}^T \bar{B} \neq X_\star$$
$$I \subseteq \{1,2,\dots,n\}, |I| = (1-\gamma)N$$

Sufficient and necessary condition

- Theorem. If $SR(X_0) \leq NR(X_0)$, then no algorithm solves problem 1 with a probability greater than 1/2.
- If $SR(X_0) > NR(X_0)$, then Algorithm 2 solves problem 1.

Algorithm 2 Exact recovery algorithm for Problem 1

Solve the following optimization problem and get $\mathscr{I}$.

$$\begin{aligned} &\min_{\mathscr{I}, L} \|\mathbf{X}^{\mathscr{I}} - L\| \\ &\text{s.t. } \operatorname{rank}(L) \leq k, \mathscr{I} \subseteq \{1, \ldots, n+n_1\}, |\mathscr{I}| = n \end{aligned} \tag{3}$$

return a basis of $\mathbf{X}^{\mathscr{I}}$.

Trimmed Principal Component Regression

- TPCR Lemma. Algorithm 3 returns $\hat{\beta}$, such that for any real value $h > 1$, with at least probability of $1 - ch^{-2}$ for some constant c, we have

$$E\left[\left(x(\hat{\beta} - \beta^\star)\right)^2\right] \le 4\sigma^2\left(1 + \sqrt{\frac{1}{1-\gamma}}\right)^2 \log c$$

Algorithm 3 Trimmed Principal Component Regression

Input: X, y

(1) Use Algorithm 2 to compute a basis from **X**, and orthogonalize it to get **B**

(2) Project **X** onto the span space of **B** and get $\mathbf{U} \leftarrow \mathbf{X}\mathbf{B}^T$

(3) Solve the following minimization problem to get $\widehat{\beta_U}$

$$\min_{\beta_U} \sum_{j=1}^{n} \{(y_i - u_i\beta_U)^2 \text{ for } i = 1, \ldots, n + n_1\}_{(j)} \quad (4)$$

where $z_{(j)}$ denotes the j-th smallest element in sequence z.

(4) **return** $\widehat{\beta} \leftarrow \mathbf{B}\widehat{\beta_U}$.

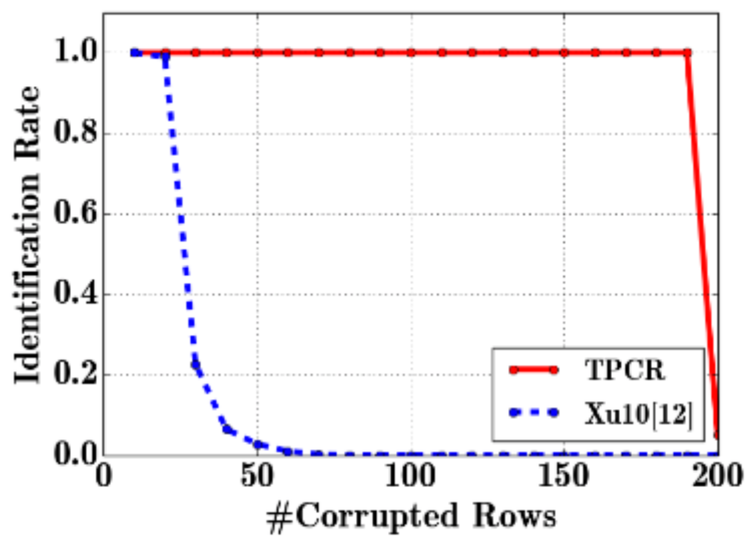
Efficient Algorithm using Alternative Minimization

- Problem: minimize the objective in the following form:

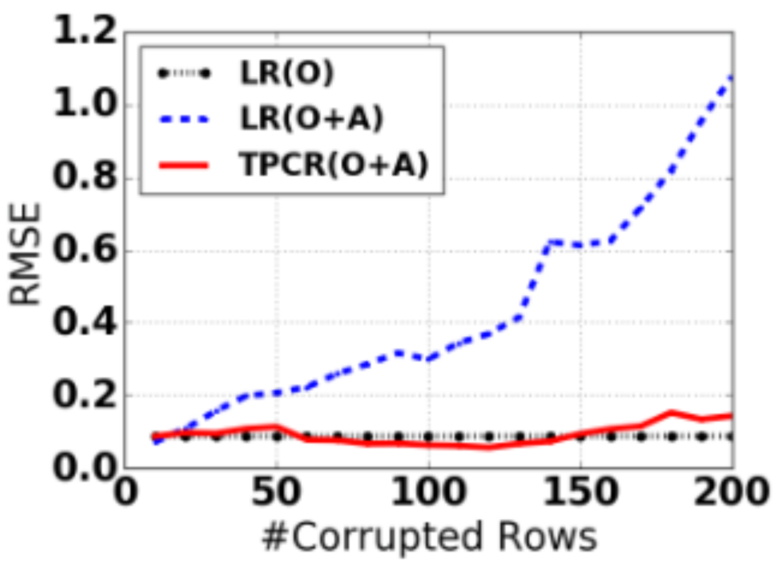
$$\min_{\theta} \sum_{j=1}^{n} \{ l\left(y_i, f_\theta(x_i)\right) \big| i = 1, \dots, n + n_1 \}_{(j)}$$

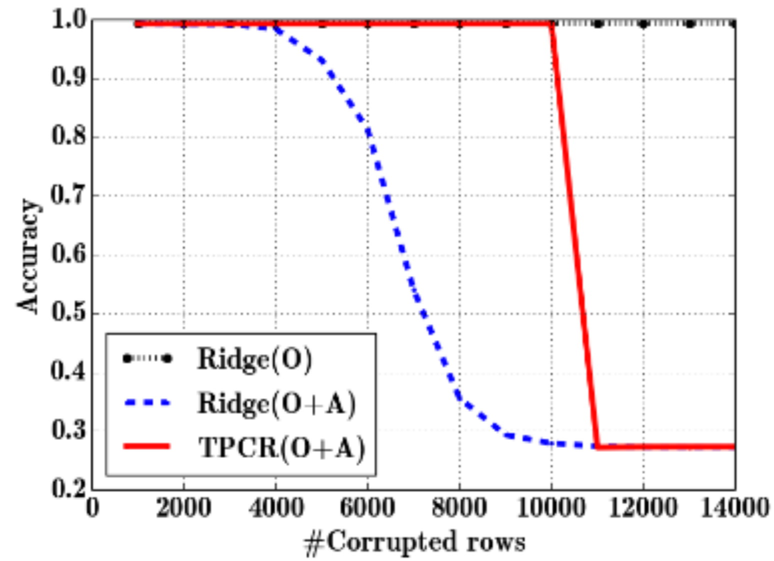
- Strategy: iteratively do the following two steps until convergence
  - Find the subset of $\{j\}$ of size n that minimizes $l\left(y_j, f_\theta(x_j)\right)$
  - Minimize the total loss with respect to θ

Sub-space recover experiments (synthetic data)



Robust regression experiments





Real malicious domain dataset

Takeaways

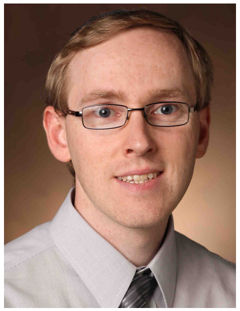
Message 1. The **poisoning attacker** can **leverage pristine data** distribution to construct strong attacks

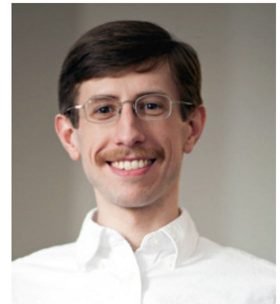
Message 2. When the **poisoning ratio** is not sufficiently large, we can **bound the loss** on the computed estimator.

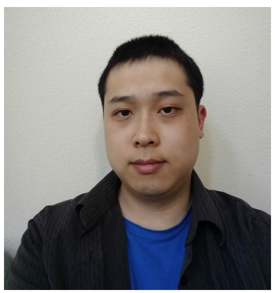
Adversarial Machine Learning

- Adversarial machine learning:
  - Learning in the presence of adversaries
- Inference time: adversarial example fools learning system
  - Evasion attacks
    - Evade malware detection; fraud detection
- Training time:
  - Attacker poisons training dataset (e.g., poison labels) to fool learning system to learn wrong model
    - Poisoning attacks: e.g., Microsoft's Tay twitter chatbot
  - Attacker selectively shows learner training data points (even with correct labels) to fool learning system to learn wrong model
  - Data poisoning is particularly challenging with crowd-sourcing & insider attack
  - Difficult to detect when the model has been poisoned
- Adversarial machine learning particularly important for security critical system

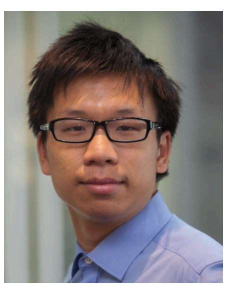
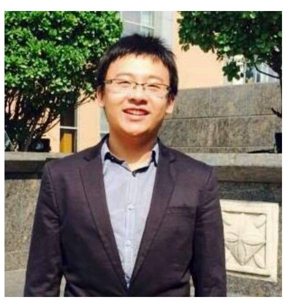
Collaborators



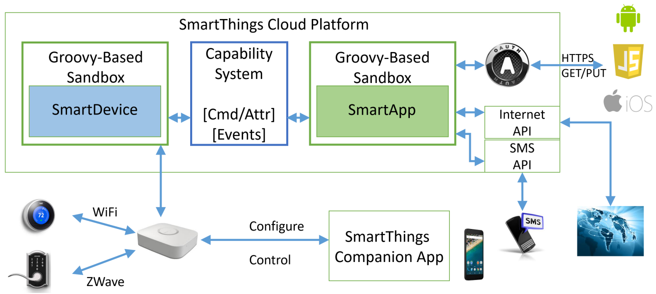



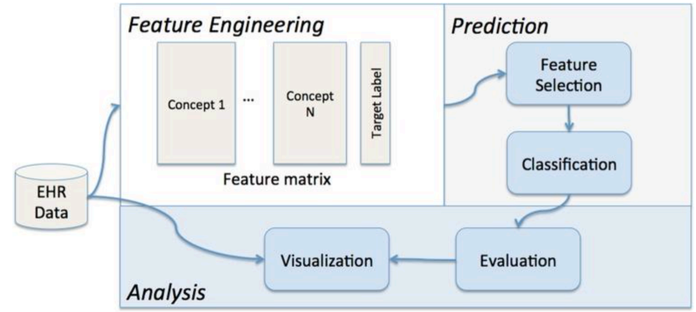


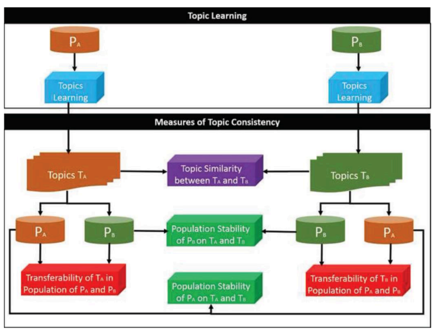




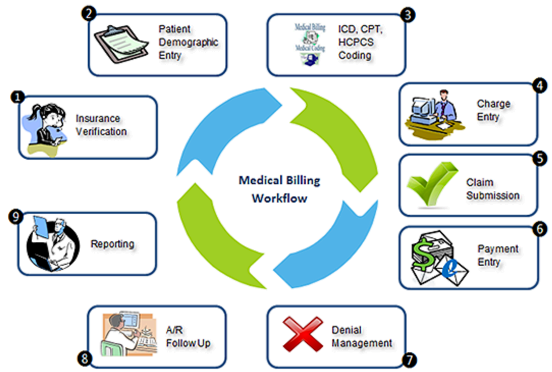
Robust Smart Home



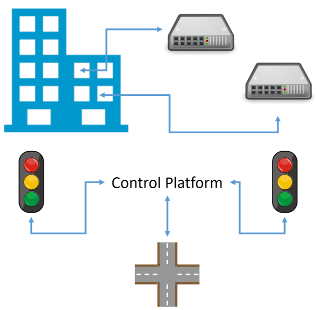
Privacy-Preserving Data Analysis



Topic of Workflow Analysis

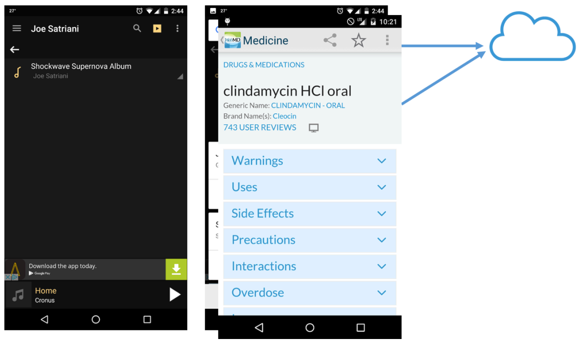


Game Theoretic Auditing System for EMR

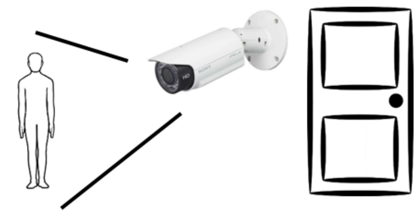


Large-Scale Auditing Game With Human In the Loop

Robust Learning



Privacy Protected Mobile Healthcare



Robust Face Recognition Against Poisoning Attack

Thank You!

Bo Li
crystalboli@berkeley.edu

http://www.crystal-boli.com/