

Adversarial Example Detection

Lecturer: Dr. Xingjun Ma

School of Computer Science, Fudan University

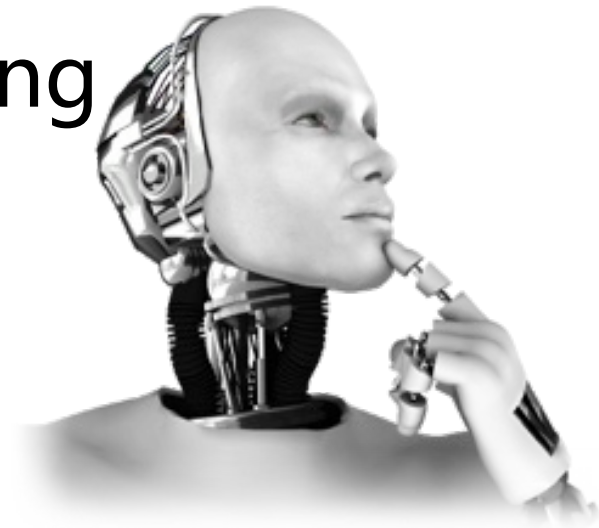
Autumn, 2022

Recap: week 3

1. Adversarial Examples

2. Adversarial Attacks

3. Adversarial Vulnerability Understanding



In-class Adversarial Attack Competition

Competition

Fudan University COMP737022 Trustworthy Machine Learning

Secret url: https://codalab.lisn.upsaclay.fr/competitions/7556?secret_key=d4a3b1fa-66e2-4a80-8ce6-b5f99e518979

Organized by huanzhuo - Current server time: Sept. 28, 2022, 8:41 a.m. UTC


Current	Next	End
Phase 1 Sept. 27, 2022, 4 p.m. UTC	Phase 2 Oct. 19, 2022, 4 p.m. UTC	Contestant Release Nov. 15, 2022, 11:59 p.m. UTC

Learn the Details Phases Participate Results

Overview
Evaluation
Terms and Conditions
get_starting_kit

White-box Adversarial Attack challenge

What does the non-robust model see?



Key Dates

Phase 1 Starts: 2022-09-28 00:00:00 (Beijing Time)

Phase 1 Ends: 2022-10-19 23:59:00 (Beijing Time)

Phase 2 Starts: 2022-10-20 00:00:00 (Beijing Time)

Phase 2 Ends: 2022-11-02 23:59:00 (Beijing Time)

Phase 3 (Final Result Release): 2022-11-07 23:00:34 (Beijing Time)

Link: https://codalab.lisn.upsaclay.fr/competitions/7556?secret_key=d4a3b1fa-66e2-4a80-8ce6-b5f99e518979

Starting kit: https://codalab.lisn.upsaclay.fr/competitions/7556?secret_key=d4a3b1fa-66e2-4a80-8ce6-b5f99e518979#learn_the_details-get_starting_kit



In-class Adversarial Attack Competition

Dataset and Target Models

Dataset: CIFAR-10

Target Model:

- WRN-34-10-SAT (adversarially trained using Standard Adversarial Training by Madry et al.)
- 2 hidden models (also adversarially trained)



In-class Adversarial Attack Competition

Phase 1:

- We will use 1k randomly selected CIFAR-10 test images and 1 robustly trained DNN (WRN-34-10-SAT) to evaluate the submitted attack method. The 1k test samples will be resampled every week to reflect the variance.
- The total number of attack steps is limited to 100 for each sample or total 100*1000 steps (see evaluation).
- Your selected submission to the leaderboard will be forwarded to the next phase.
- Your algorithm should finish within 10 minutes during this stage. The time constraint is measured by the server. For reference, 10 steps PGD attack (PGD-10) finishes in 1 minute.

Phase 2:

- We will select 1k of the hardest-to-attack images from the CIFAR-10 test set to evaluate the submitted attack method.
- We will evaluate the attacking method against the same target model as in stage 1.
- Select the submission you want to use for final evaluation before the deadline, and submit it to the leaderboard.
- Your participation ends at this stage.

Phase 3 (Final Evaluation)

- We will use the same 1k hardest images used in phase 2 to evaluate the submitted attack method.
- We will evaluate the attacking method against two more adversarially trained models at this stage.
- The attacking method you submitted in phase 2 will be directly transferred to this stage.
- Select your submission in Phase 2!



In-class Adversarial Attack Competition

◆ Adversarial attack competition (**account for 20%**)

- **必须使用复旦邮箱注册比赛（否则无成绩）**
- 比赛时间：
 - Phase 1：9月28号 – 10月20号
 - Phase 2：10月20号 – 11月02号
 - Phase 3：最终评估，学生不参与

□ 得分计算：按排名算分：

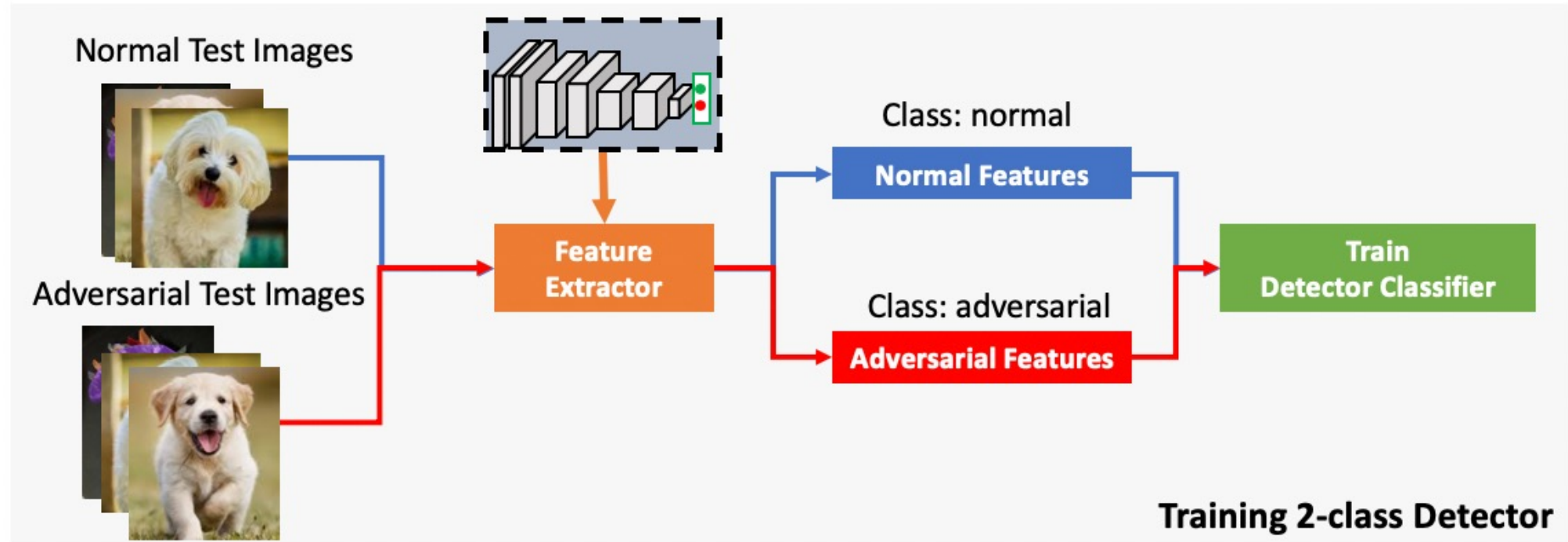
□ **第一名100分**

□ **最后一名50分**

没卡的同学建议使用Google Colab：<https://colab.research.google.com/>

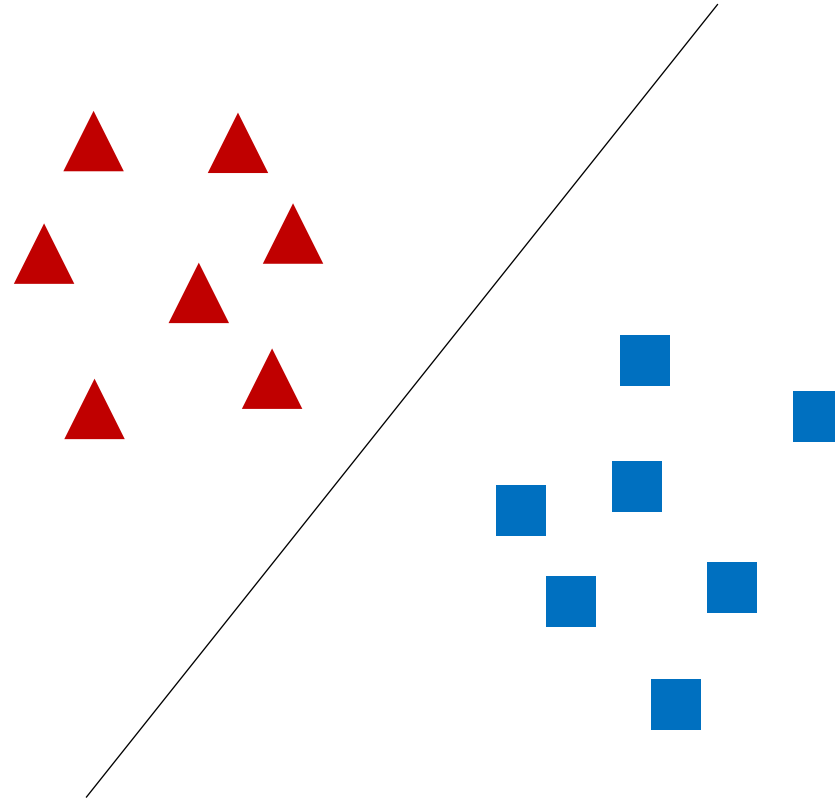


Adversarial Example Detection (AED)



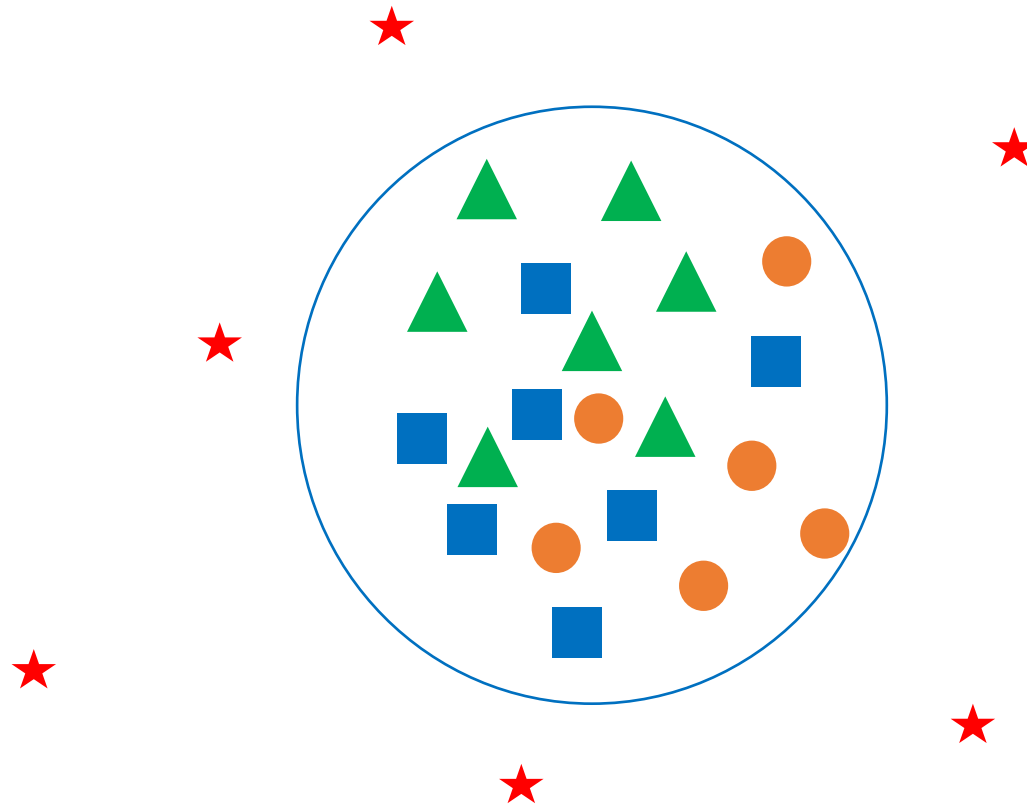
- ❑ A **binary** classification problem: clean ($y=0$) or adv ($y=1$)?
- ❑ An anomaly detection problem: benign ($y=0$) or abnormal ($y=1$)?

Principles for AED



□ All binary classification methods can be applied for AED

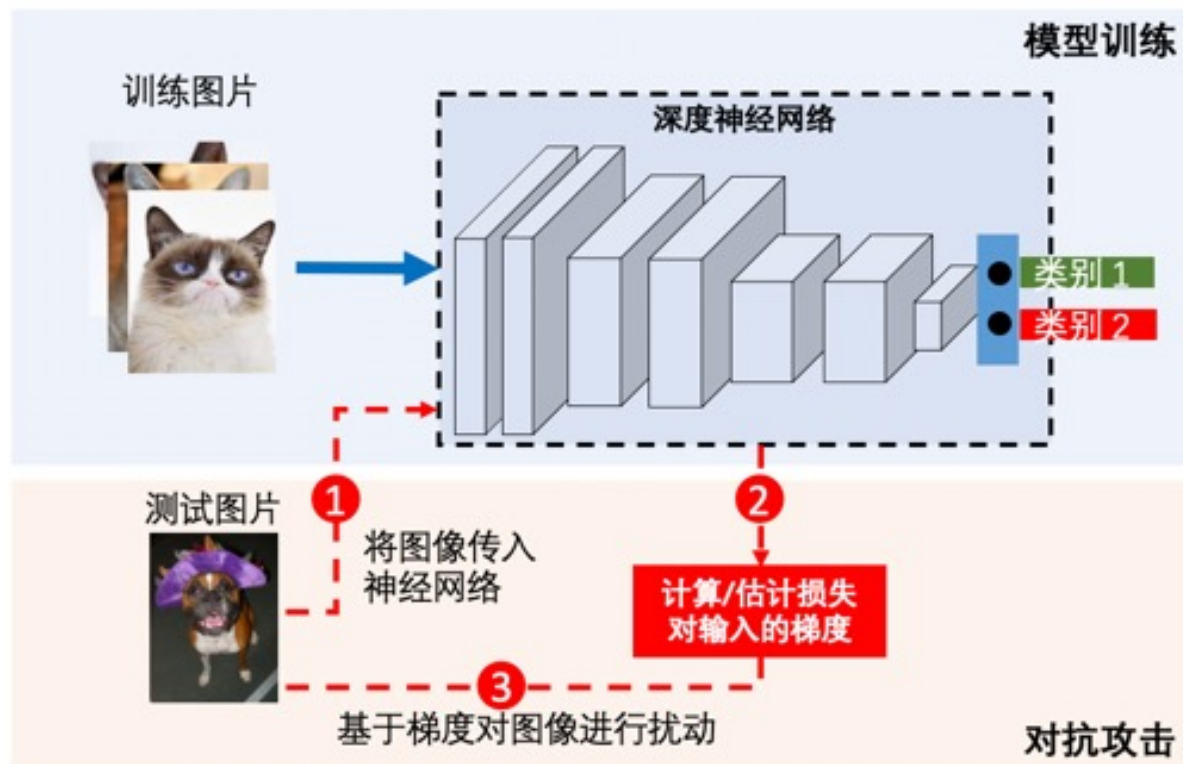
Principles for AED



□ All anomaly detection methods can be applied for AED

Principles for AED

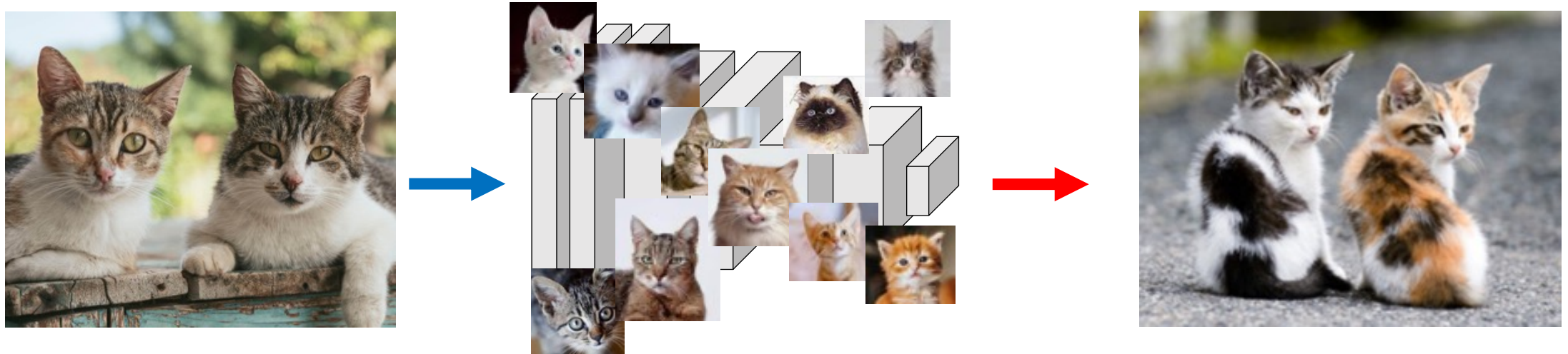
- Input statistics
- Manual features
- Training data
- Attention map
- Transformation
- Mixup
- Denoising
- ...



- Activations
- Deep features
- Probabilities
- Logits
- Gradients
- Loss landscape
- Uncertainty
- ...

□ Use all the information we have

Principles for AED



Twins

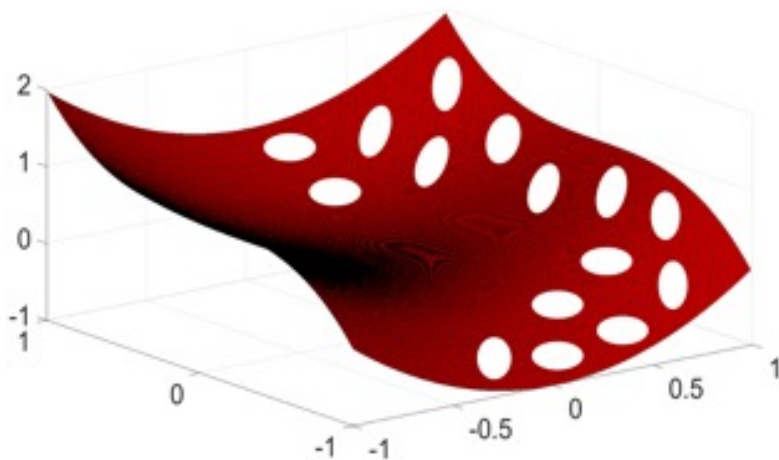
Extremely close to the clean sample

Strangers

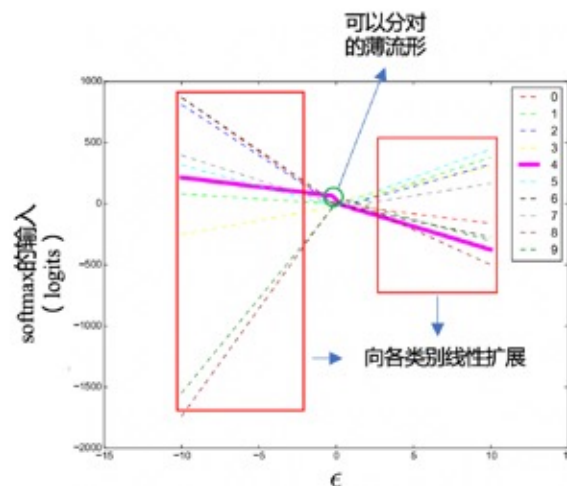
Far away in prediction

❑ Leverage unique characteristics of adversarial examples

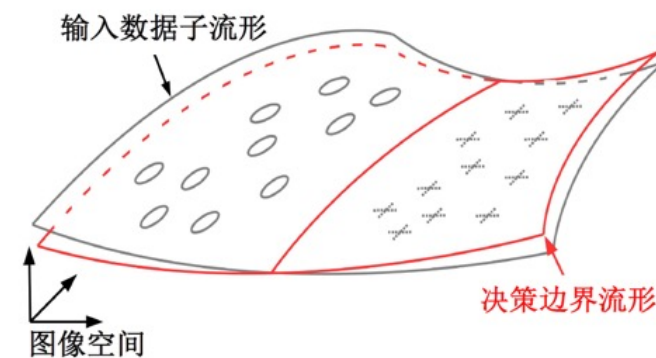
Principles for AED



High dimensional pockets



Local linearity



Tilting boundary

□ Build detectors based on existing understandings

Principles for AED

it's all about features!



Challenges in AED

- ❑ The diversity of adversarial examples used for training the detectors determine the detection performance
- ❑ Detectors are also machine learning model: they are also vulnerable to adversarial attacks
- ❑ The detectors need to detect both existing and unknown attacks
- ❑ The detectors need to be robust to adaptive attacks



Existing Methods

- ❑ Secondary Classification Methods (二级分类法)
- ❑ Principle Component Analysis (主成分分析法, PCA)
- ❑ Distribution Detection Methods (分布检测法)
- ❑ Prediction Inconsistency (预测不一致性)
- ❑ Reconstruction Inconsistency (重建不一致性)
- ❑ Trapping Based Detection (诱捕检测法)



Existing Methods

- ❑ **Secondary Classification Methods (二级分类法)**
- ❑ Principle Component Analysis (主成分分析法, PCA)
- ❑ Distribution Detection Methods (分布检测法)
- ❑ Prediction Inconsistency (预测不一致性)
- ❑ Reconstruction Inconsistency (重建不一致性)
- ❑ Trapping Based Detection (诱捕检测法)



Secondary Classification Methods

Adversarial Retraining (对抗重训练)

1. 在正常训练集 D_{train} 上训练得到模型 f
2. 基于 D_{train} 对抗攻击模型 f 得到对抗样本集 D_{adv}
3. 将 D_{adv} 中的所有样本标注为 $C + 1$ 类别
4. 在 $D_{\text{train}} \cup D_{\text{adv}}$ 上训练得到 f_{secure}

□ Take adversarial examples as a new class



Secondary Classification Methods

Adversarial Classification (对抗分类)

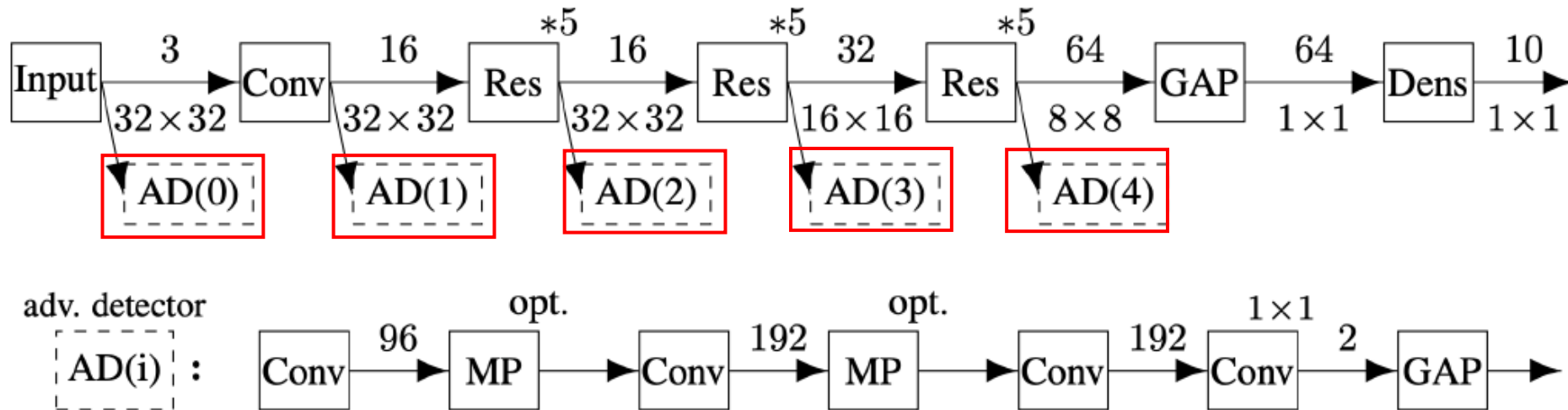
1. 在正常训练集 D_{train} 上训练得到模型 f
2. 基于 D_{train} 对抗攻击模型 f 得到对抗样本集 D_{adv}
3. 将 D_{train} 标记为 0 类别, 将 D_{adv} 标注为 1 类别
4. 在 $D_{\text{train}} \cup D_{\text{adv}}$ 上训练得到二分类检测器 g

□ Clean samples as class 0, adversarial as class 1



Secondary Classification Methods

Cascade Classifiers (级联分类器)



□ Training a detector for each intermediate layer

Metzen, Jan Hendrik, et al. "On detecting adversarial perturbations." arXiv preprint arXiv:1702.04267 (2017).

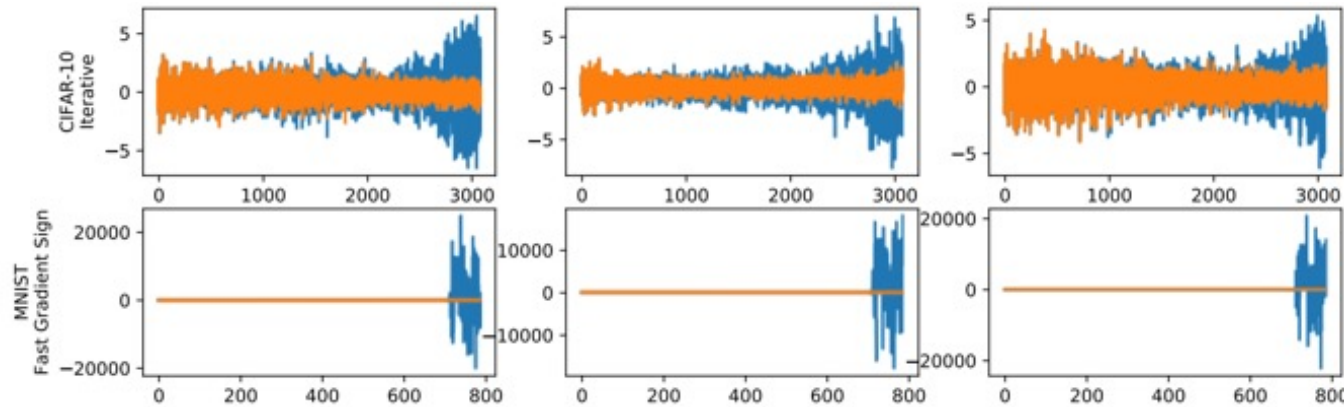


Existing Methods

- ❑ Secondary Classification Methods (二级分类法)
- ❑ **Principle Component Analysis (主成分分析法, PCA)**
- ❑ Distribution Detection Methods (分布检测法)
- ❑ Prediction Inconsistency (预测不一致性)
- ❑ Reconstruction Inconsistency (重建不一致性)
- ❑ Trapping Based Detection (诱捕检测法)

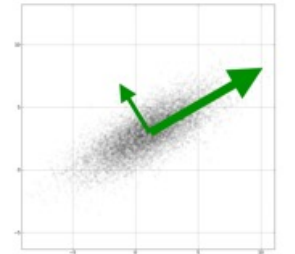


Principle Component Analysis (PCA)



Blue: a clean sample

Yellow: an adv example



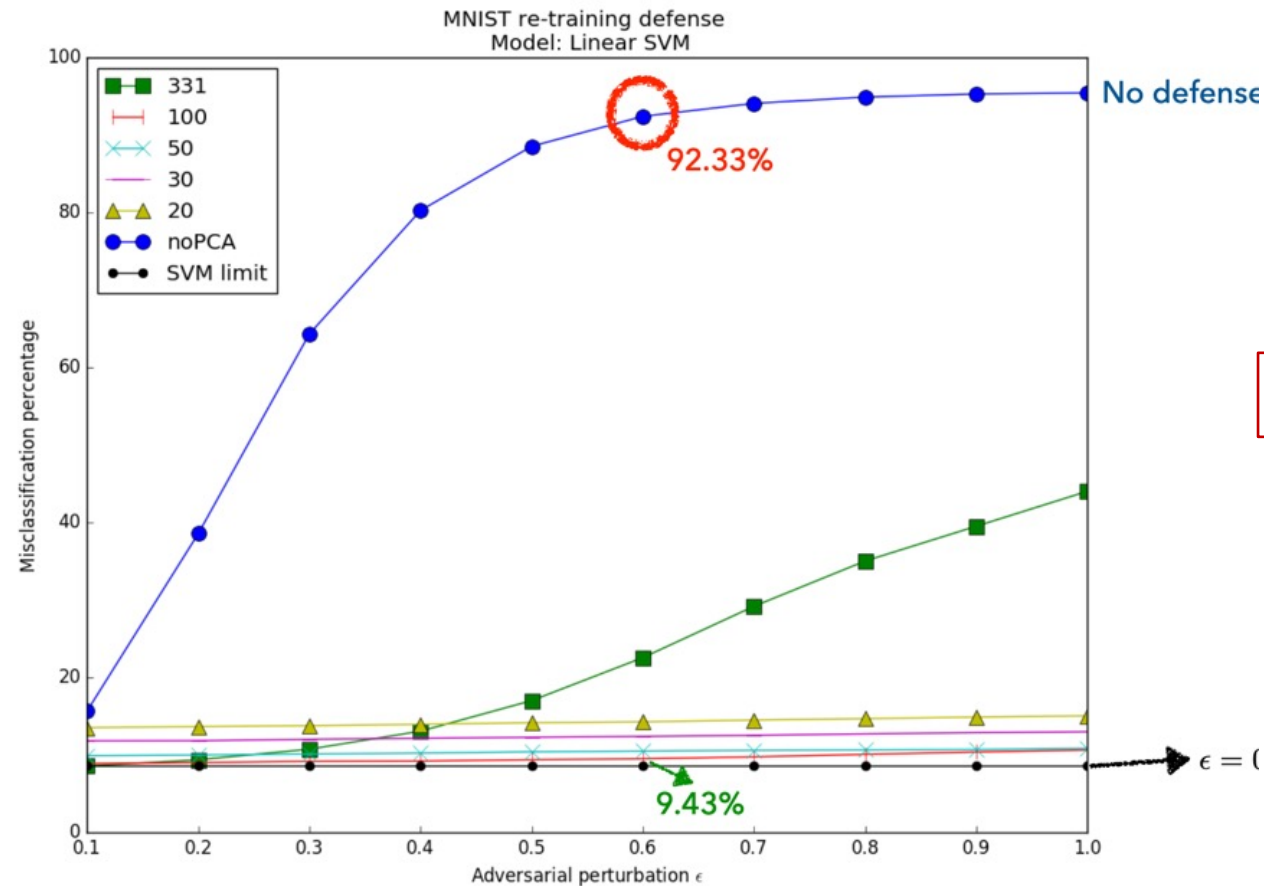
An artifact caused by the black background

□ The last few components differentiate adversarial examples

Hendrycks, Dan, and Kevin Gimpel. "Early methods for detecting adversarial images." arXiv:1608.00530 (2016); Carlini and Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods." *A/Sec.* 2017.



Dimensionality Reduction



Bhagoji, Arjun Nitin, Daniel Cullina, and Prateek Mittal. "Dimensionality reduction as a defense against evasion attacks on machine learning classifiers." *arXiv:1704.02654* 2.1 (2017).



Existing Methods

- ❑ Secondary Classification Methods (二级分类法)
- ❑ Principle Component Analysis (主成分分析法, PCA)
- ❑ **Distribution Detection Methods (分布检测法)**
 - ❑ Prediction Inconsistency (预测不一致性)
 - ❑ Reconstruction Inconsistency (重建不一致性)
 - ❑ Trapping Based Detection (诱捕检测法)



Distribution Detection

Maximum Mean Discrepancy (MMD)

Two datasets: D_1 vs. D_2

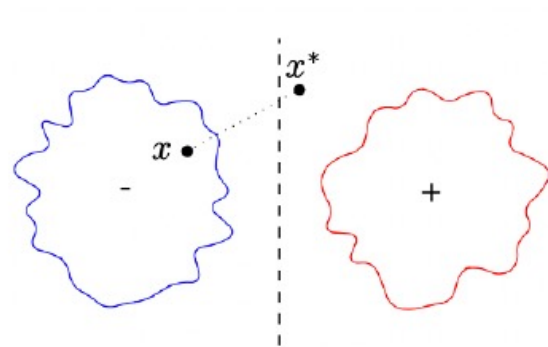
$$MMD(\mathcal{K}, D_1, D_2) = \sup_{k \in \mathcal{K}} \left(\frac{1}{n} \sum_{i=1}^n k(D_1^i) - \frac{1}{m} \sum_{i=1}^m k(D_2^i) \right)$$

1. 在 D_1 和 D_2 上计算 $a = MMD(\mathcal{K}, D_1, D_2)$;
2. 对 D_1 和 D_2 中的样本顺序做随机打乱得到对应的 D'_1 和 D'_2 ;
3. 在 D'_1 和 D'_2 上计算 $b = MMD(\mathcal{K}, D'_1, D'_2)$;
4. 如果 $a < b$ 则拒绝原假设, 即 D_1 和 D_2 来自不同分布;
5. 重复执行步骤 1-4 很多次 (1 万次), 计算原假设被拒绝的比例作为 p -值。

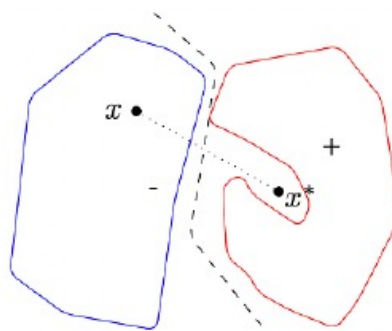


Distribution Detection

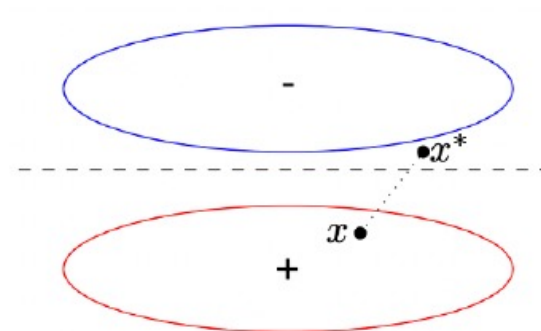
Kernel Density Estimation (KDE)



(a) 对抗样本离两个子流形都很远



(b) 对抗样本在+子流形附近的口袋里



(c) 对抗样本离目标子流形很近

Adversarial examples are in low density space

Feinman, Reuben, et al. "Detecting adversarial samples from artifacts." arXiv preprint arXiv:1703.00410 (2017).



Distribution Detection

Kernel Density Estimation (KDE)

$$KDE(\mathbf{x}) = \frac{1}{|X_t|} \sum_{s \in X_t} \exp\left(-\frac{|\mathbf{z}(\mathbf{x}) - \mathbf{z}(s)|^2}{\sigma^2}\right)$$

x : 需要计算核密度的样本

X_t : 类别为 t 的训练样本子集

z : 模型最后一层的逻辑输出

σ : 控制高斯核平滑度的bandwidth超参

Adversarial examples are in low density space

Feinman, Reuben, et al. "Detecting adversarial samples from artifacts." arXiv preprint arXiv:1703.00410 (2017).



Bypassing 10 Detection Methods

Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods.
Carlini and Wagner, AISec 2017.



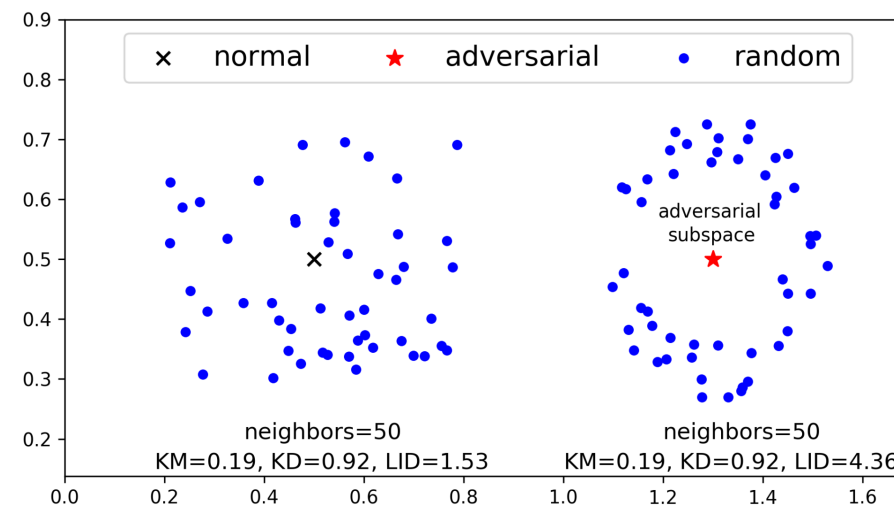
Local Intrinsic Dimensionality (LID)

Definition (Local Intrinsic Dimensionality)

Given a data sample $x \in X$, let $r > 0$ be a random variable denoting the distance from x to other data samples. The *local intrinsic dimension* of x at distance r is

$$\text{LID}_F(r) \triangleq \frac{r \cdot F'(r)}{F(r)}$$

wherever the limit exists.



Adversarial examples are in high dimensional subspaces

Characterizing Adversarial Subspace Using Local Intrinsic Dimensionality. *Ma et al. ICLR 2018*



Local Intrinsic Dimensionality (LID)

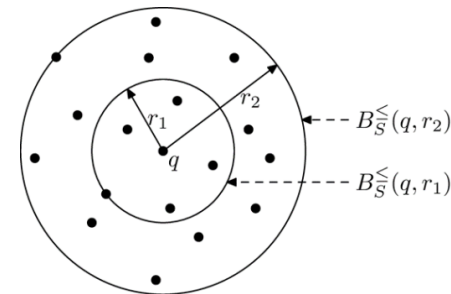
Adversarial Subspaces and Expansion Dimension:

Expansion Dimension:

- Two balls of radius r_1 and r_2 , dimension m can be deduced from ratios of volumes:

$$\frac{V_2}{V_1} = \left(\frac{r_2}{r_1}\right)^m \Rightarrow m = \frac{\ln(V_2/V_1)}{\ln(r_2/r_1)}$$

- Related to the Expansion Dimension (*Karger and Ruhl 2002, Houle et al. 2012*)
- V_1 and V_2 estimated by the numbers of points contained in the two balls.



Characterizing Adversarial Subspace Using Local Intrinsic Dimensionality. *Ma et al. ICLR 2018*

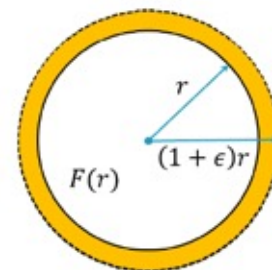


Local Intrinsic Dimensionality (LID)

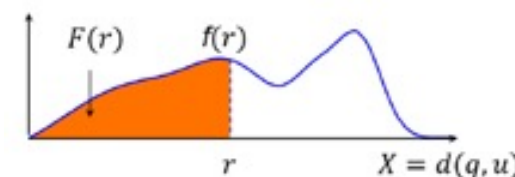
Estimation of LID:

- Hill (MLE) estimator (*Hill 1975, Amsaleg et al. 2015*):

$$\widehat{\text{LID}}(x) = - \left(\frac{1}{k} \sum_{i=1}^k \log \frac{r_i(x)}{r_k(x)} \right)^{-1}, \quad r_i \text{ is the distance of } x \text{ to its } i^{\text{th}} \text{ nearest neighbor.}$$



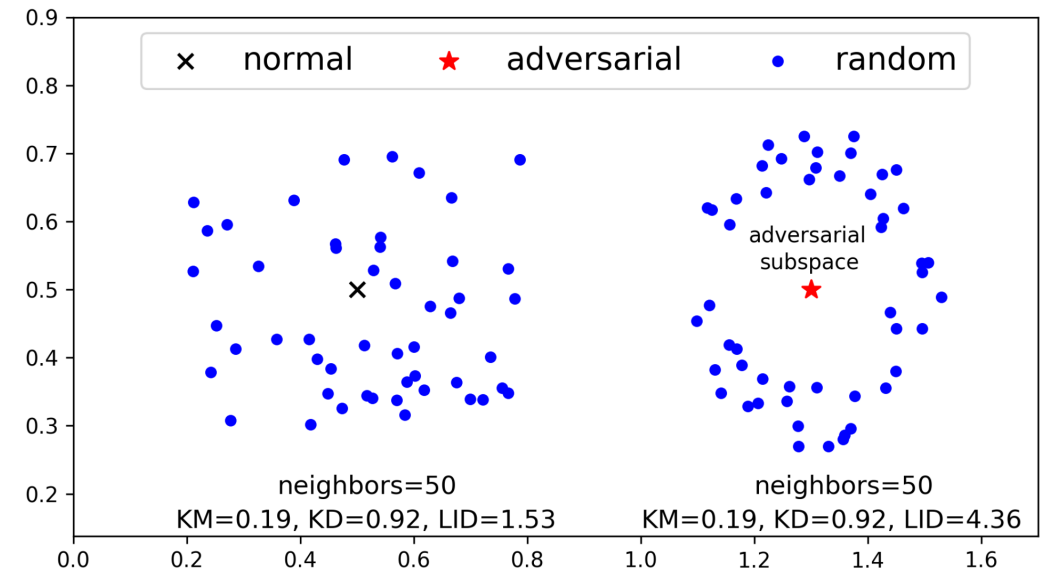
- Based on Extreme Value Theory:
 - Nearest neighbor distances are extreme events.
 - Lower tail distribution follows Generalized Pareto Distribution (GPD).



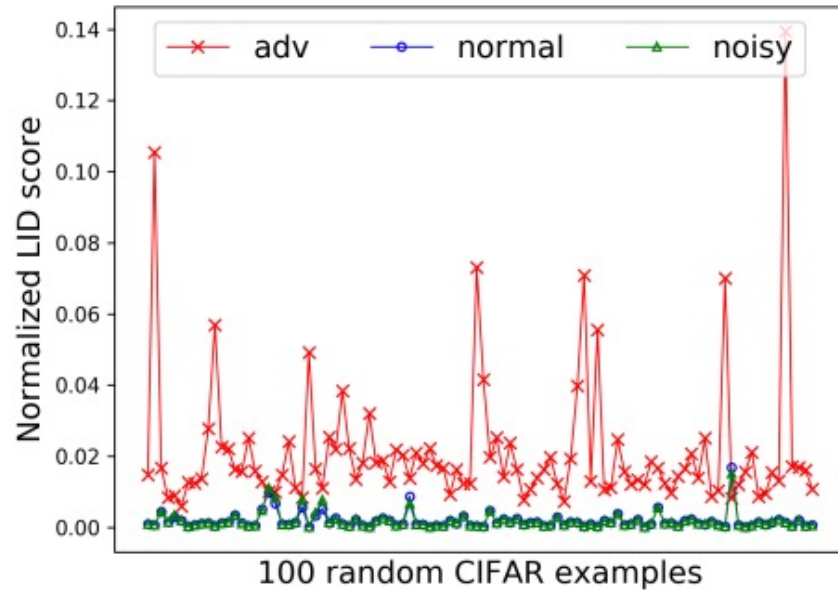
Local Intrinsic Dimensionality (LID)

Interpretation of LID for Adversarial Subspaces:

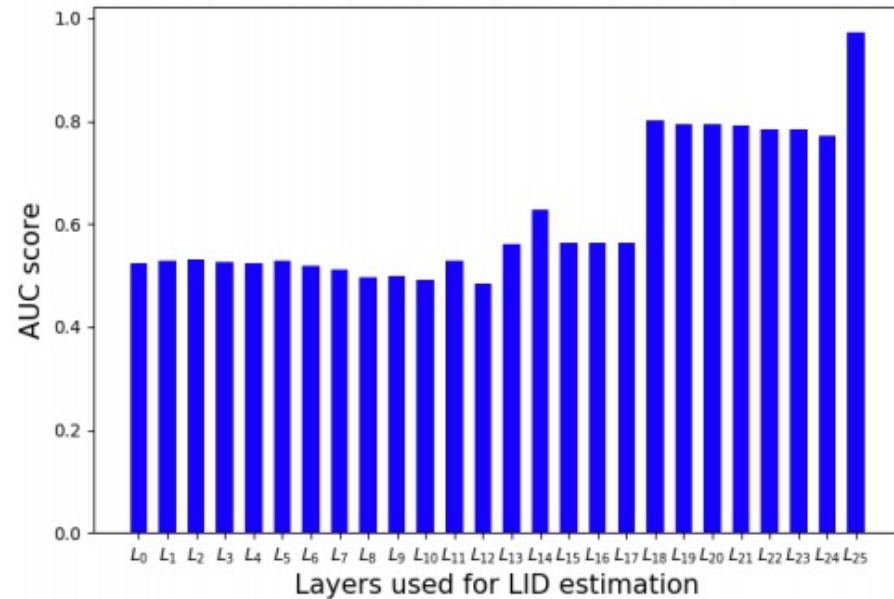
- LID directly measures expansion rate of local distance distributions.
- The expansion of adversarial subspace is higher than normal data subspace.
- LID assesses the space-filling capability of the subspace, based on the distance distribution of the example to its neighbors.



Local Intrinsic Dimensionality (LID)



- LID of adversarial examples (red) are higher



- LID at deeper layers are more differentiable

Characterizing Adversarial Subspace Using Local Intrinsic Dimensionality. *Ma et al. ICLR 2018*



Local Intrinsic Dimensionality (LID)

Algorithm 7.1 训练 LID 对抗样本检测器

输入: \mathbf{x} : 原始训练集; $f(\mathbf{x})$: 已训练的神经网络, 共 l 层; k : 近邻样本数量

- 1: 初始化检测器训练集: $LID_{\text{neg}} = [], LID_{\text{pos}} = []$
 - 2: **for** B_{norm} in \mathbf{x} **do**
 - 3: $B_{\text{adv}} :=$ 对抗攻击本批样本 B_{norm}
 - 4: $N = |B_{\text{norm}}|$
 - 5: 初始化 LID 特征集 $LID_{\text{norm}}, LID_{\text{adv}}$ 为零矩阵 (维度均为 $[n, l]$)
 - 6: **for** i in $[1, l]$ **do**
 - 7: 抽取中间层特征: $A_{\text{norm}} = f^i(B_{\text{norm}}), A_{\text{adv}} = f^i(B_{\text{adv}})$
 - 8: **for** j in $[1, n]$ **do**
 - 9: $LID_{\text{norm}}[j, i] = -\left(\frac{1}{k} \sum_{i=1}^k \log \frac{r_i(A_{\text{norm}}[j], A_{\text{norm}})}{r_k(A_{\text{norm}}[j], A_{\text{norm}})}\right)^{-1}$
 - 10: $LID_{\text{adv}}[j, i] = -\left(\frac{1}{k} \sum_{i=1}^k \log \frac{r_i(A_{\text{adv}}[j], A_{\text{norm}})}{r_k(A_{\text{adv}}[j], A_{\text{norm}})}\right)^{-1}$
 - 11: $LID_{\text{neg}}.append(LID_{\text{norm}}), LID_{\text{pos}}.append(LID_{\text{adv}})$
 - 12: 在数据集 $D = \{(LID_{\text{neg}}, y = 0), (LID_{\text{pos}}, y = 1)\}$ 上训练检测器 g
- 输出:** 检测器 g
-



Local Intrinsic Dimensionality (LID)

Experiments & Results:

Dataset	Feature	FGM	BIM-a	BIM-b	JSMA	Opt
MNIST	KD	78.12	98.14	98.61	68.77	95.15
	BU	32.37	91.55	25.46	88.74	71.30
	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
	LID	82.38	82.51	99.78	95.87	98.94
SVHN	KD	70.39	77.18	99.57	86.46	87.41
	BU	86.78	84.07	86.93	91.33	87.13
	LID	97.61	87.55	99.72	95.07	97.60

Characterizing Adversarial Subspace Using Local Intrinsic Dimensionality. *Ma et al. ICLR 2018*



Local Intrinsic Dimensionality (LID)

Experiments & Results:

Train \ Test attack		FGM	BIM-a	BIM-b	JSMA	Opt
FGSM	KD	64.92	69.15	89.71	85.72	91.22
	BU	70.53	81.67	2.65	86.79	91.27
	LID	82.38	82.30	91.61	89.93	93.32

Detectors trained on simple attacks FGSM can detect complex attacks

Characterizing Adversarial Subspace Using Local Intrinsic Dimensionality. *Ma et al. ICLR 2018*



Mahalanobis Distance (MD)

- The MD of a data point x to a distribution Q :

$$d_M(x) = \sqrt{(x - \mu)^\top \Sigma^{-1} (x - \mu)}$$

μ : sample mean in Q
 Σ : covariance matrix

- The MD of between two data points:

$$d_M(x_i, x_2) = \sqrt{(x_i - x_2)^\top \Sigma^{-1} (x_i - x_2)}$$

Mahalanobis, Prasanta Chandra. "On the generalized distance in statistics." National Institute of Science of India, 1936.



Mahalanobis Distance (MD)

Given a model f and training dataset D , the MD of a sample x is defined as

$$d_M(\mathbf{x}) = \max_c -(f^{L-2}(\mathbf{x}) - \mu_c)\Sigma^{-1}(f^{L-2}(\mathbf{x}) - \mu_c)$$

$$\mu_c = \frac{1}{N_c} \sum_{\mathbf{x} \in X_c} f^{L-2}(\mathbf{x})$$

$$\Sigma_c = \frac{1}{N_c} \sum_c \sum_{\mathbf{x} \in X_c} (f^{L-2}(\mathbf{x}) - \mu_c)^{\top}$$

f^{L-2} : 深度神经网络倒数第二层的输出

μ_c : 类别C的样本特征均值

Σ_c : 类别C的样本间协方差矩阵

N_c : 类别C的样本数量

Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks." NeurIPS 2018.



Mahalanobis Distance (MD)

Algorithm 7.2 基于马氏距离的对抗样本检测

输入: 测试样本 \mathbf{x} , 逻辑回归检测器权重 α_l , 噪声大小 ϵ 以及高斯分布参数 $\{\mu_{l,c}, \Sigma_l : \forall l, c\}$

- 1: 初始化分数向量: $M(\mathbf{x}) = [M_l : \forall l]$
- 2: **for** 每一层 $l = 1, \dots, L$ **do**
- 3: 寻找最近的类别: $\hat{c} = \arg \min_c (f^l(\mathbf{x}) - \mu_{l,c})^\top \Sigma_l^{-1} (f^l(\mathbf{x}) - \mu_{l,c})$
- 4: 向样本中添加噪声: $\hat{\mathbf{x}} = \mathbf{x} \leftarrow \mathbf{x} + \epsilon \cdot \text{sign} \left(\Delta_x (f^l(\mathbf{x}) - \mu_{l,c})^\top \Sigma_l^{-1} (f^l(\mathbf{x}) - \mu_{l,c}) \right)$
- 5: 计算置信度: $M_l = \max_c - (f^l(\mathbf{x}) - \mu_{l,c})^\top \Sigma_l^{-1} (f^l(\mathbf{x}) - \mu_{l,c})$

输出: 样本 \mathbf{x} 的总检测置信度 $\sum_l \alpha_l M_l$

Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks." NeurIPS 2018.



Mahalanobis Distance (MD)

Experiments & Results:

Model	Dataset (model)	Score	Detection of known attack				Detection of unknown attack			
			FGSM	BIM	DeepFool	CW	FGSM (seen)	BIM	DeepFool	CW
DenseNet	CIFAR-10	KD+PU [7]	85.96	96.80	68.05	58.72	85.96	3.10	68.34	53.21
		LID [22]	98.20	99.74	85.14	80.05	98.20	94.55	70.86	71.50
		Mahalanobis (ours)	99.94	99.78	83.41	87.31	99.94	99.51	83.42	87.95
	CIFAR-100	KD+PU [7]	90.13	89.69	68.29	57.51	90.13	66.86	65.30	58.08
		LID [22]	99.35	98.17	70.17	73.37	99.35	68.62	69.68	72.36
		Mahalanobis (ours)	99.86	99.17	77.57	87.05	99.86	98.27	75.63	86.20
	SVHN	KD+PU [7]	86.95	82.06	89.51	85.68	86.95	83.28	84.38	82.94
		LID [22]	99.35	94.87	91.79	94.70	99.35	92.21	80.14	85.09
		Mahalanobis (ours)	99.85	99.28	95.10	97.03	99.85	99.12	93.47	96.95
ResNet	CIFAR-10	KD+PU [7]	81.21	82.28	81.07	55.93	83.51	16.16	76.80	56.30
		LID [22]	99.69	96.28	88.51	82.23	99.69	95.38	71.86	77.53
		Mahalanobis (ours)	99.94	99.57	91.57	95.84	99.94	98.91	78.06	93.90
	CIFAR-100	KD+PU [7]	89.90	83.67	80.22	77.37	89.90	68.85	57.78	73.72
		LID [22]	98.73	96.89	71.95	78.67	98.73	55.82	63.15	75.03
		Mahalanobis (ours)	99.77	96.90	85.26	91.77	99.77	96.38	81.95	90.96
	SVHN	KD+PU [7]	82.67	66.19	89.71	76.57	82.67	43.21	84.30	67.85
		LID [22]	97.86	90.74	92.40	88.24	97.86	84.88	67.28	76.58
		Mahalanobis (ours)	99.62	97.15	95.73	92.15	99.62	95.39	72.20	86.73

Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks." NeurIPS 2018.



Existing Methods

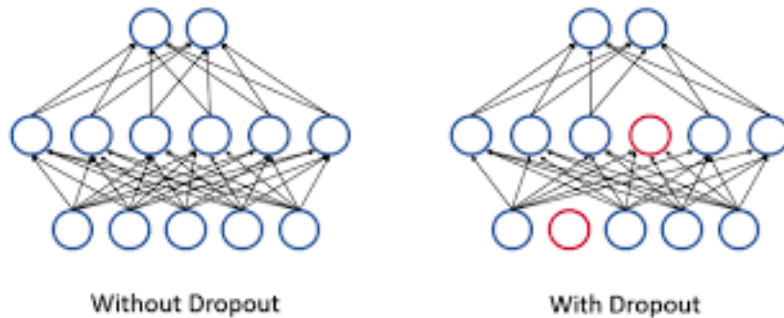
- ❑ Secondary Classification Methods (二级分类法)
- ❑ Principle Component Analysis (主成分分析法, PCA)
- ❑ Distribution Detection Methods (分布检测法)
- ❑ **Prediction Inconsistency (预测不一致性)**
- ❑ Reconstruction Inconsistency (重建不一致性)
- ❑ Trapping Based Detection (诱捕检测法)



Bayes Uncertainty

Bayesian Uncertainty (BU)

$$U(\mathbf{x}) = \frac{1}{T} \sum_{i=1}^T \hat{\mathbf{y}}_i^\top \hat{\mathbf{y}}_i - \left(\frac{1}{T} \sum_{i=1}^T \hat{\mathbf{y}}_i \right)^\top \left(\frac{1}{T} \sum_{i=1}^T \hat{\mathbf{y}}_i \right)$$



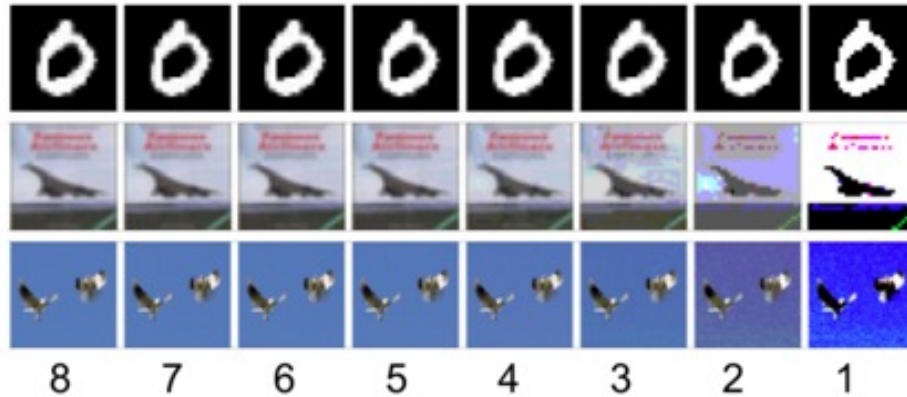
Use test time dropout to get randomized networks

T : the number of randomization.

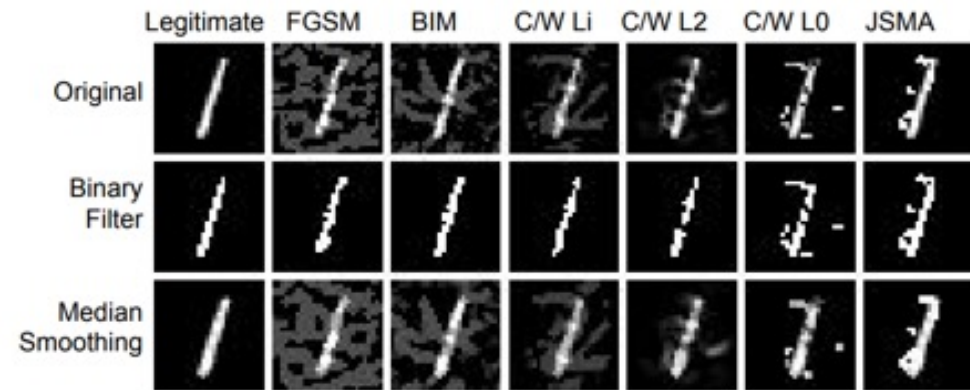
Feinman, Reuben, et al. "Detecting adversarial samples from artifacts." arXiv preprint arXiv:1703.00410 (2017).



Feature Squeezing



Bit depth reduction

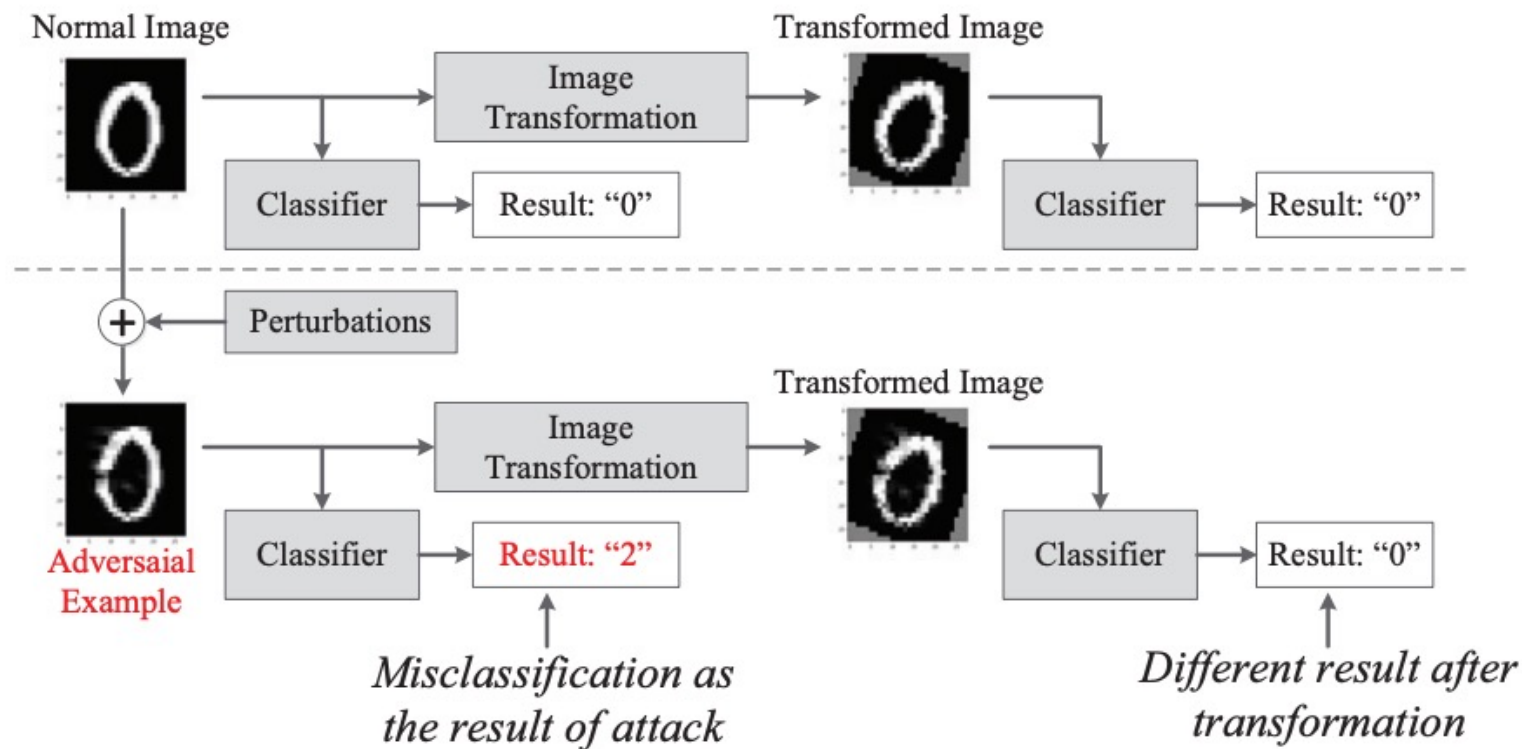


Squeezing clean and adv examples

- ❑ Reducing input dimensionality improves robustness
- ❑ The prediction inconsistency before and after squeezing can detect advs

Xu et al. "Feature squeezing: Detecting adversarial examples in deep neural networks." arXiv:1704.01155 (2017).

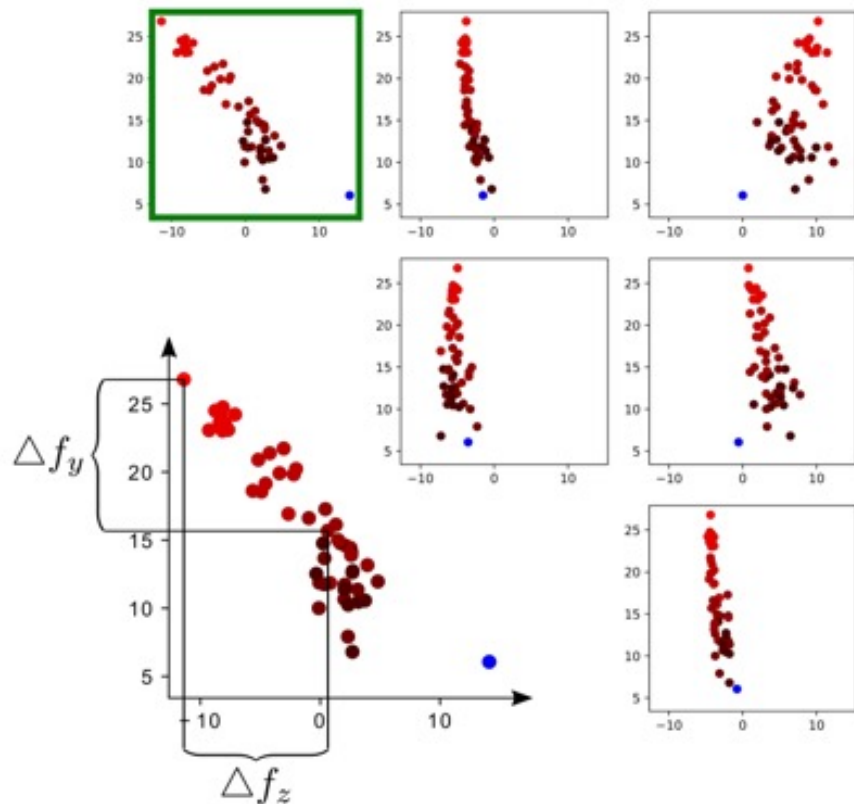
Random Transformation



❑ The prediction of advs will change after random transformations

Tian et al. "Detecting adversarial examples through image transformation." AAAI 2018.

Log-Odds



f_y : 类别y对应的逻辑输出

f_z : 类别z对应的逻辑输出

蓝色点：原始样本

红色点：对抗样本

□ Add random noise to the input

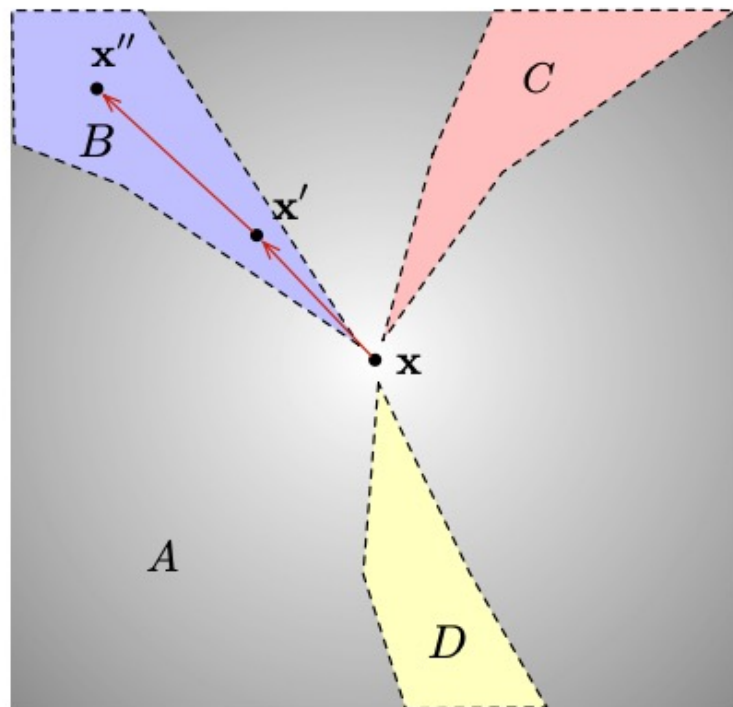
$$x' = x + \eta, \quad \eta \sim \mathcal{N}(\mu, \delta^2)$$

$$f(x') \approx f(x) ??$$

Roth et al. "The odds are odd: A statistical test for detecting adversarial examples." ICML 2019.



Log-Odds



- 原则1：对抗样本的梯度更均匀
- 原则2：对抗样本难以被攻击第二次



- 测试准则1：随机噪声不会改变预测结果
- 测试准则1：再次攻击需要更多的扰动

Hu et al. "A new defense against adversarial images: Turning a weakness into a strength." NeurIPS 2019.

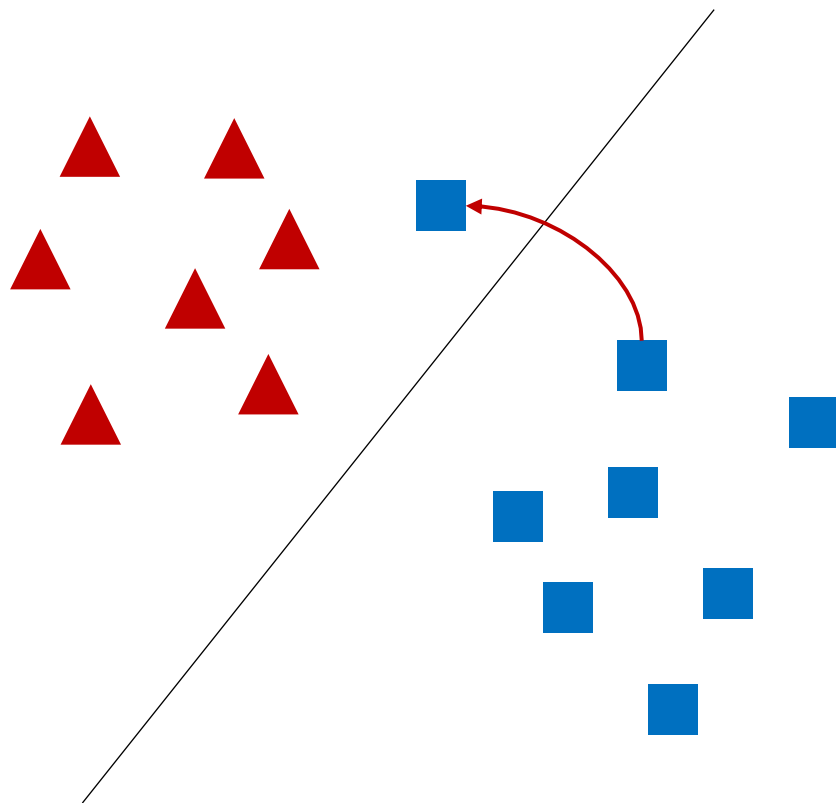


Existing Methods

- ❑ Secondary Classification Methods (二级分类法)
- ❑ Principle Component Analysis (主成分分析法, PCA)
- ❑ Distribution Detection Methods (分布检测法)
- ❑ Prediction Inconsistency (预测不一致性)
- ❑ **Reconstruction Inconsistency (重建不一致性)**
- ❑ Trapping Based Detection (诱捕检测法)



Detector-Reformer

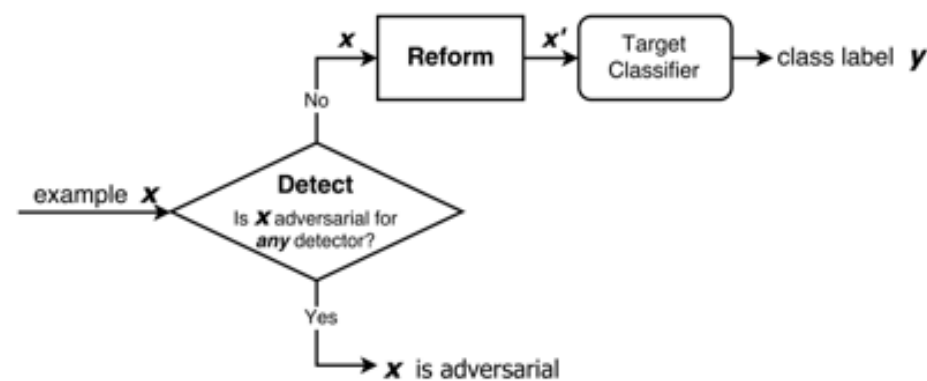


□ 原则：对抗样本无法重建

$$E(x) = \|x - AE(x)\|_p$$

AE: Autoencoder

E(x): reconstruction error

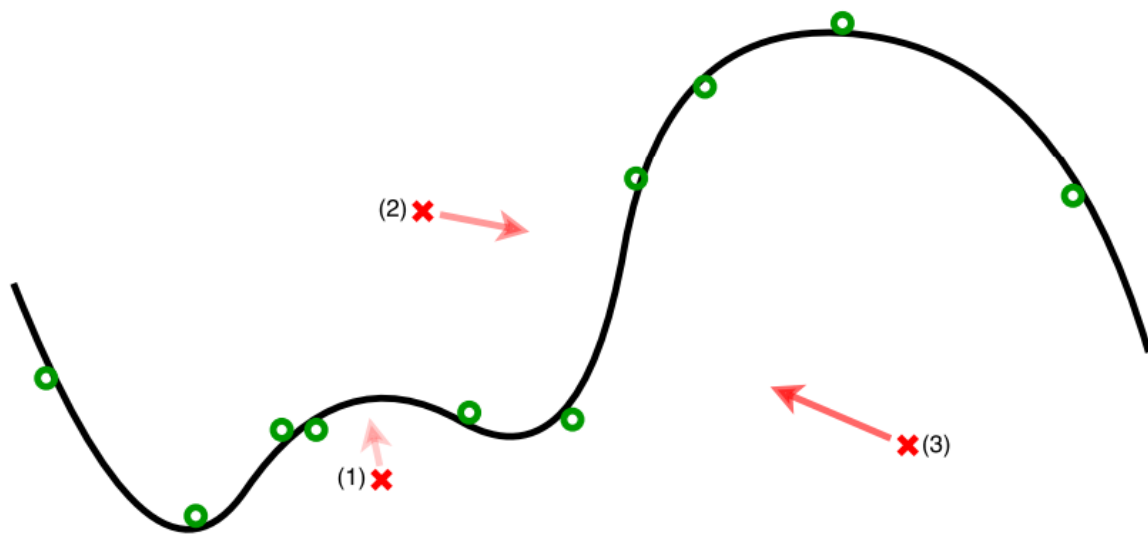


Meng, Dongyu, and Hao Chen. "Magnet: a two-pronged defense against adversarial examples", SIGSAC 2019.



Detector-Reformer

How the reformer works?



绿色：正常样本
红色x：对抗样本
红色箭头：自编码器

Meng, Dongyu, and Hao Chen. "Magnet: a two-pronged defense against adversarial examples", SIGSAC 2019.

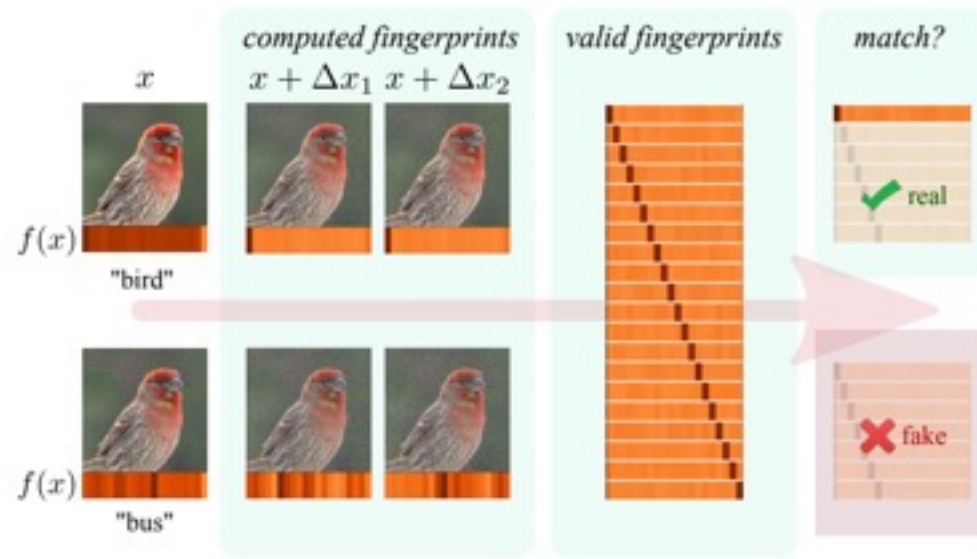


Existing Methods

- ❑ Secondary Classification Methods (二级分类法)
- ❑ Principle Component Analysis (主成分分析法, PCA)
- ❑ Distribution Detection Methods (分布检测法)
- ❑ Prediction Inconsistency (预测不一致性)
- ❑ Reconstruction Inconsistency (重建不一致性)
- ❑ **Trapping Based Detection (诱捕检测法)**



Neural Fingerprinting (NFP)



Detect advs with N=2 fingerprints

Fingerprint is defined as:

$$\mathcal{X}^{i,j} = (\Delta x^i, \Delta y^{i,j}), i = 1, \dots, N, \quad j = 1, \dots, C$$

Dathathri, Sumanth, et al. "Detecting adversarial examples via neural fingerprinting." arXiv:1803.03870 (2018).

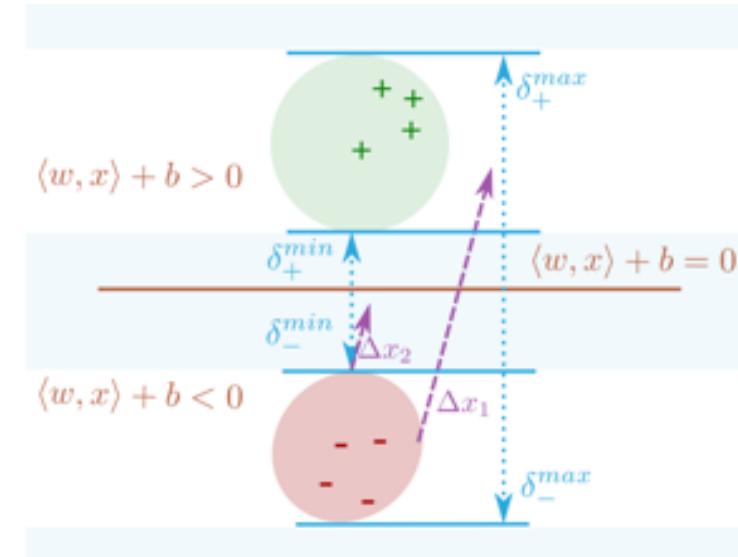


Neural Fingerprinting (NFP)

How to verify the fingerprint?

$$D(\mathbf{x}, f, \mathcal{X}^{\dots j}) = \frac{1}{N} \sum_{i=1}^N \|f(\mathbf{x} + \Delta x^i) - f(\mathbf{x}) - \Delta y^{i,j}\|_2$$

Δx_i is class-independent noise



Dathathri, Sumanth, et al. "Detecting adversarial examples via neural fingerprinting." arXiv:1803.03870 (2018).

C U Next Week!

Course page:

<https://trustworthymachinelearning.github.io/>

Textbook:

下载链接: https://pan.baidu.com/s/1kybxud_tz0xshWpMEORAhg?pwd=tauu

Email: xingjunma@fudan.edu.cn

Personal page: www.xingjunma.com

Office: 江湾校区交叉二号楼D5025

