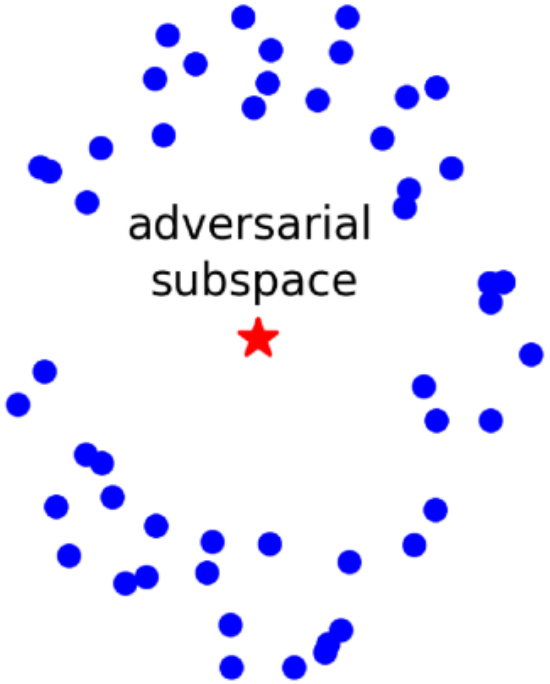


Adversarial Example Detection

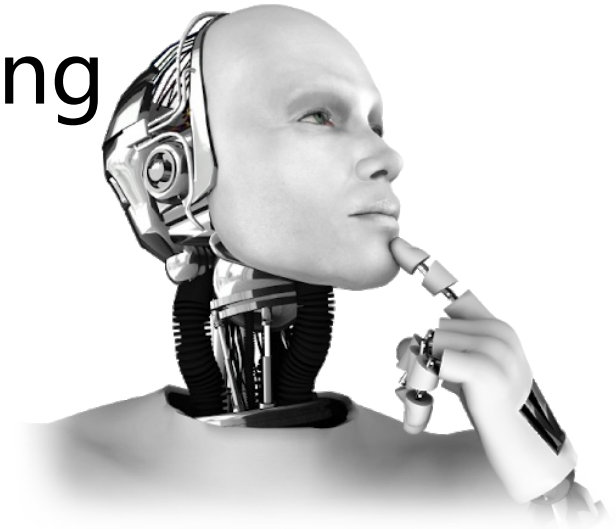
马兴军，复旦大学 计算机学院

Recap: week 3

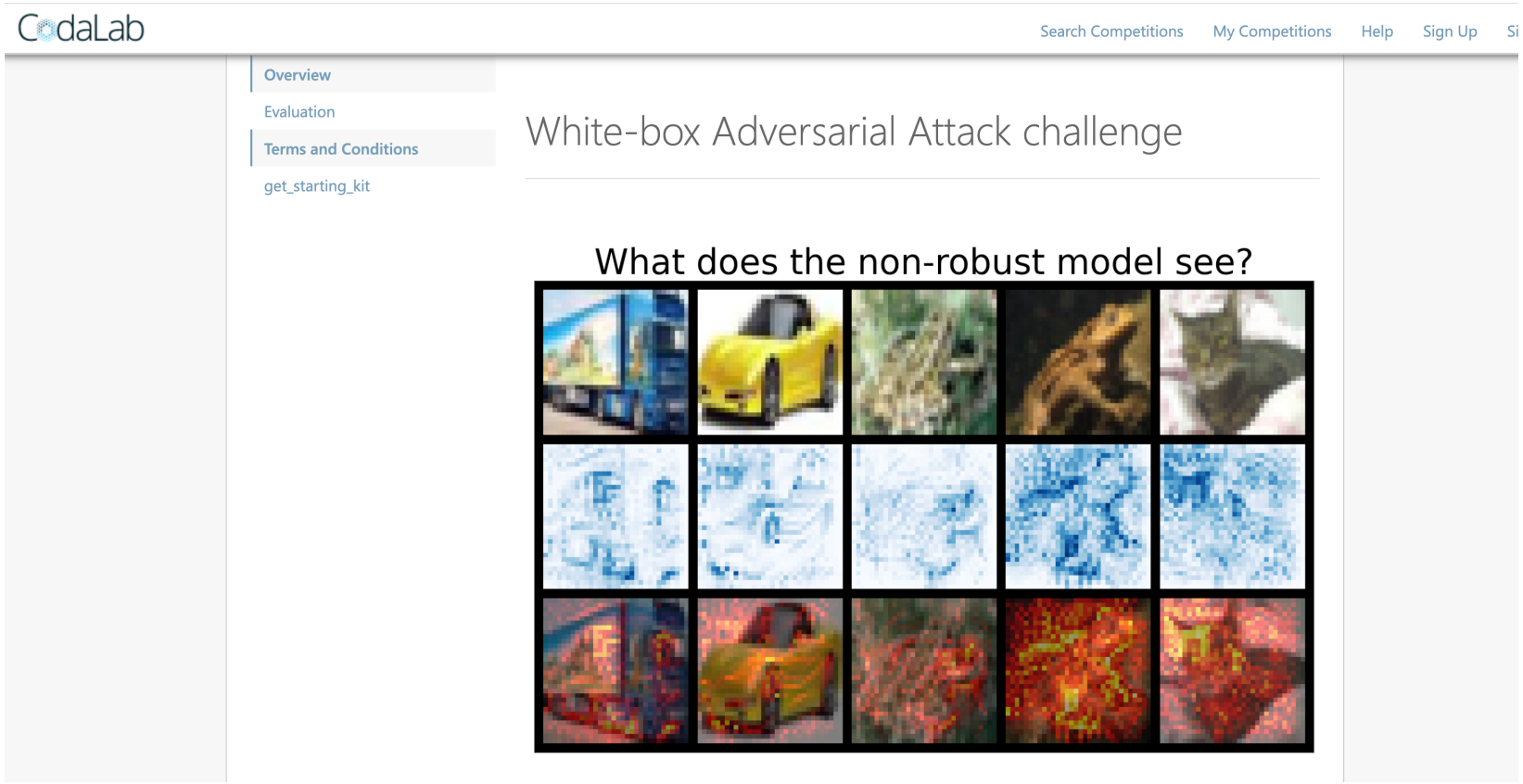
1. Adversarial Examples

2. Adversarial Attacks

3. Adversarial Vulnerability Understanding



In-class Adversarial Attack Competition



https://codalab.lisn.upsaclay.fr/competitions/15669?secret_key=77cb8986-d5bd-4009-82f0-7dde2e819ff8

In-class Adversarial Attack Competition

2023 Fudan University COMP737022 Trustworthy Machine Learning

Secret url: https://codalab.lisn.upsaclay.fr/competitions/15669?secret_key=77cb8986-d5bd-4009-82f0-7dde2e819ff8

Organized by hanxunh - Current server time: Sept. 27, 2023, 2:44 a.m. UTC

First phase	End
Phase 1	Competition Ends
Oct. 1, 2023, 4 p.m. UTC	Nov. 5, 2023, 11:59 p.m. UTC

Learn the Details | Phases | Participate | Results

Phase 1

Start: Oct. 1, 2023, 4 p.m.

Description: Create an attack method and submit the code as submission. Your code should follows the submission template. Feedback will be provided on the all test images. We will test your code on 1 robustly trained model.

Phase 2

Start: Nov. 1, 2023, 4 p.m.

Description: Your code in phase 1 will be evaluated in this phase. Feedback will be provided on all test images. We will test your code on 4 robustly trained model.

第一学期 2023年8月27日至2024年1月6日

周次	日	一	二	三	四	五	六
0	8/27	28	29	30	31	9/1	2
1	3	4	5	6	7	8	9
2	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23
4	24	25	26	27	28	29	30
5	10/1	2	3	4	5	6	7
6	8	9	10	11	12	13	14
7	15	16	17	18	19	20	21
8	22	23	24	25	26	27	28
9	29	30	31	11/1	2	3	4
10	5	6	7	8	9	10	11
11	12	13	14	15	16	17	18
12	19	20	21	22	23	24	25
13	26	27	28	29	30	12/1	2
14	3	4	5	6	7	8	9
15	10	11	12	13	14	15	16
16	17	18	19	20	21	22	23
17	24	25	26	27	28	29	30
18	31	1/1	2	3	4	5	6

备注

1. 2023级本科生8月27日报到，8月28日至9月1日入学教育，9月4日上课。
2. 2023级研究生8月26日报到，8月28日至9月1日入学教育，9月4日上课。
3. 本科生老生线上申请补考，8月30日至9月3日补考，9月3日注册，9月4日上课。
4. 研究生老生线上申请补考，8月30日至9月3日补考，9月1日注册，9月4日上课。
5. 2023级本科生、研究生开学典礼于第0周举行。
6. 中秋节、国庆节、元旦节放假以学校办通知为准。
7. 通识教育课程考试安排在第16周，第17、18周为停课考试周。
8. 第一学期于2024年1月6日结束，共计18教学周（包括考试）。

In-class Adversarial Attack Competition

◆ Adversarial attack competition （**account for 30%**）

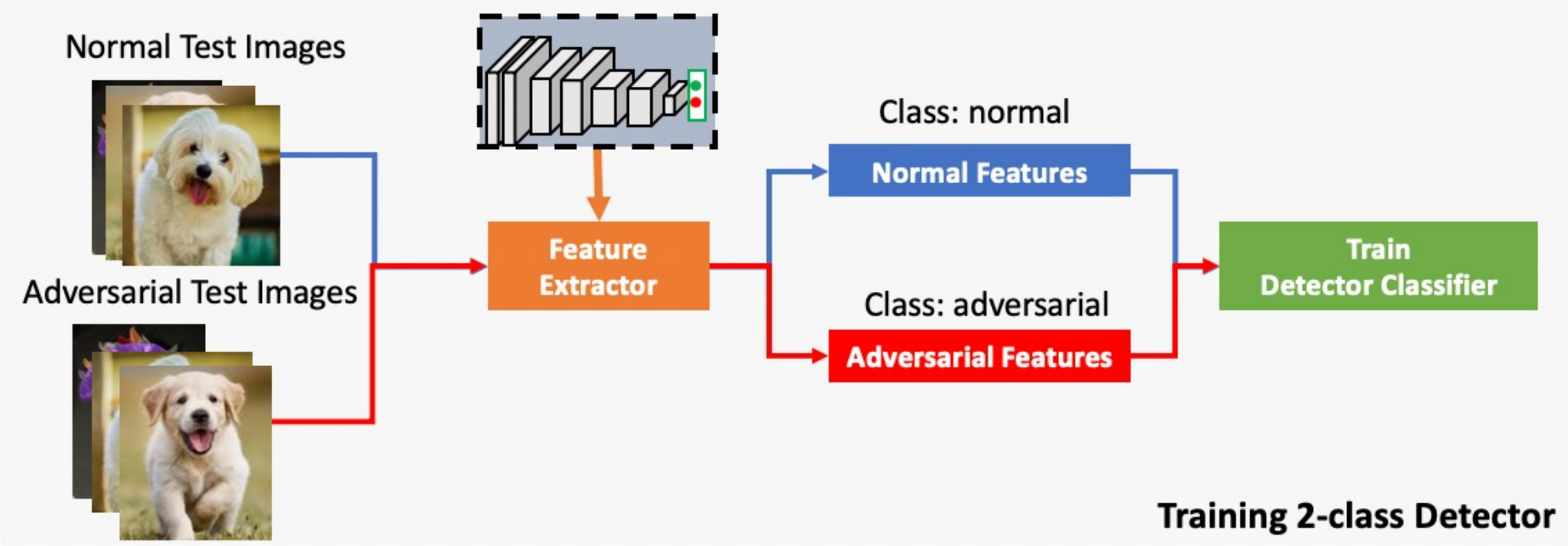
- **必须使用复旦邮箱注册比赛（否则无成绩）**
- 比赛时间：
  - Phase 1： 10月1号 – 10月28号
  - Phase 2：评估阶段，学生不参与

□ 按排名算分：

- **第一名30分**
- **最后一名15分**

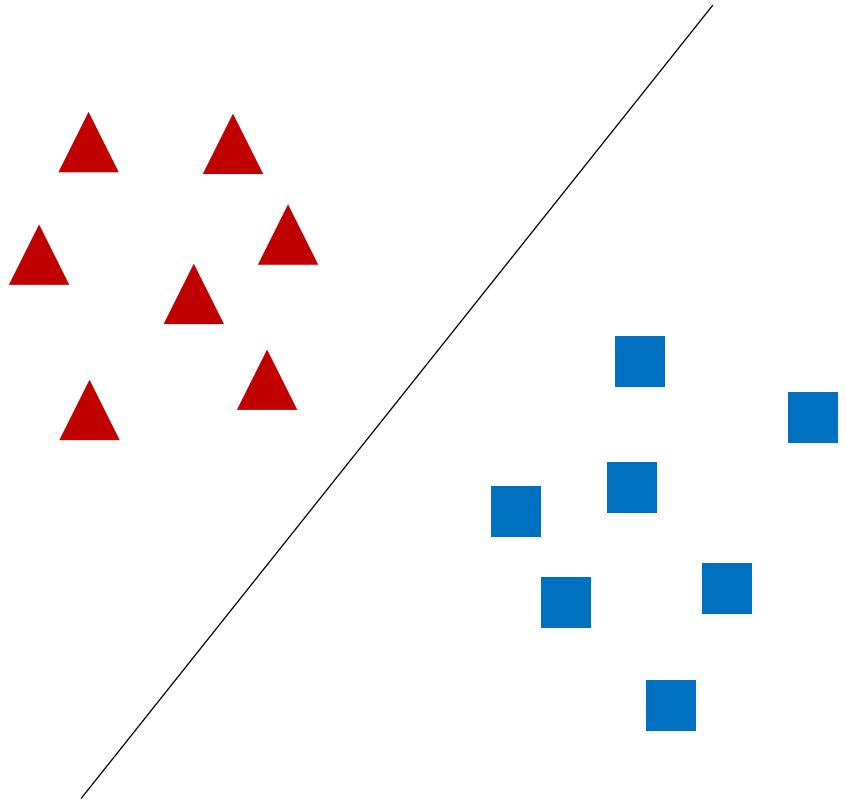
没卡的同学建议使用Google Colab：https://colab.research.google.com/

Adversarial Example Detection (AED)



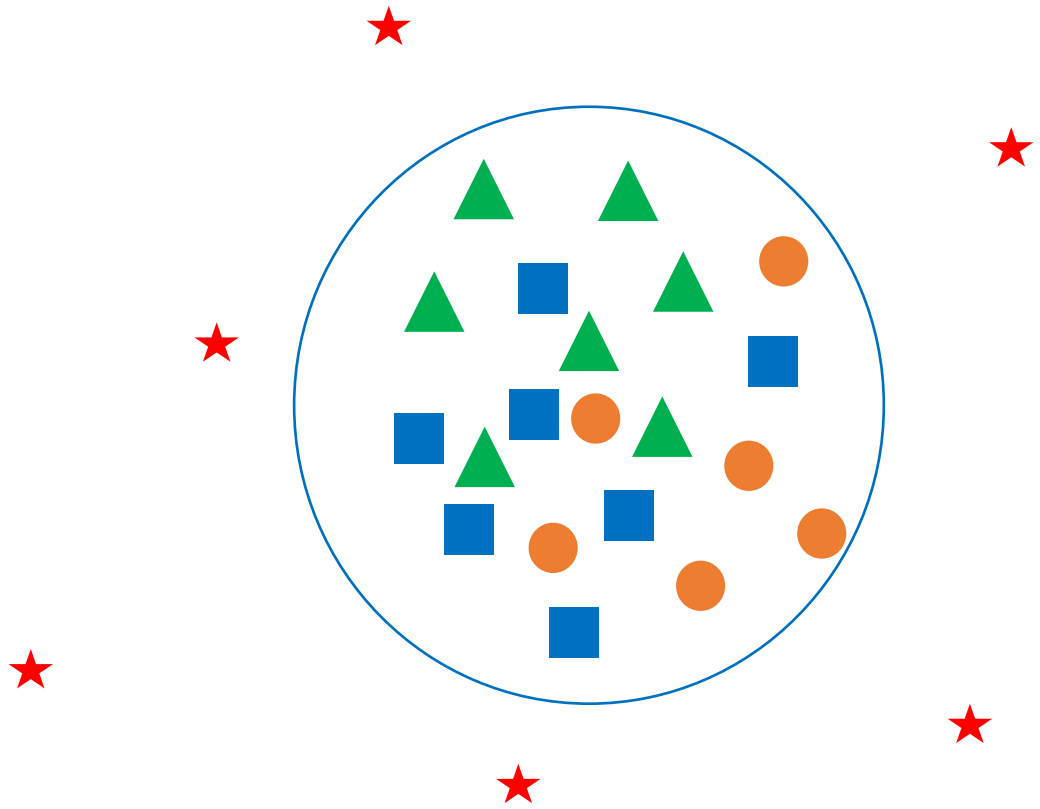
- A **binary** classification problem: clean (y=0) or adv (y=1)?
- An anomaly detection problem: benign (y=0) or abnormal (y=1)?

Principles for AED



- All binary classification methods can be applied for AED

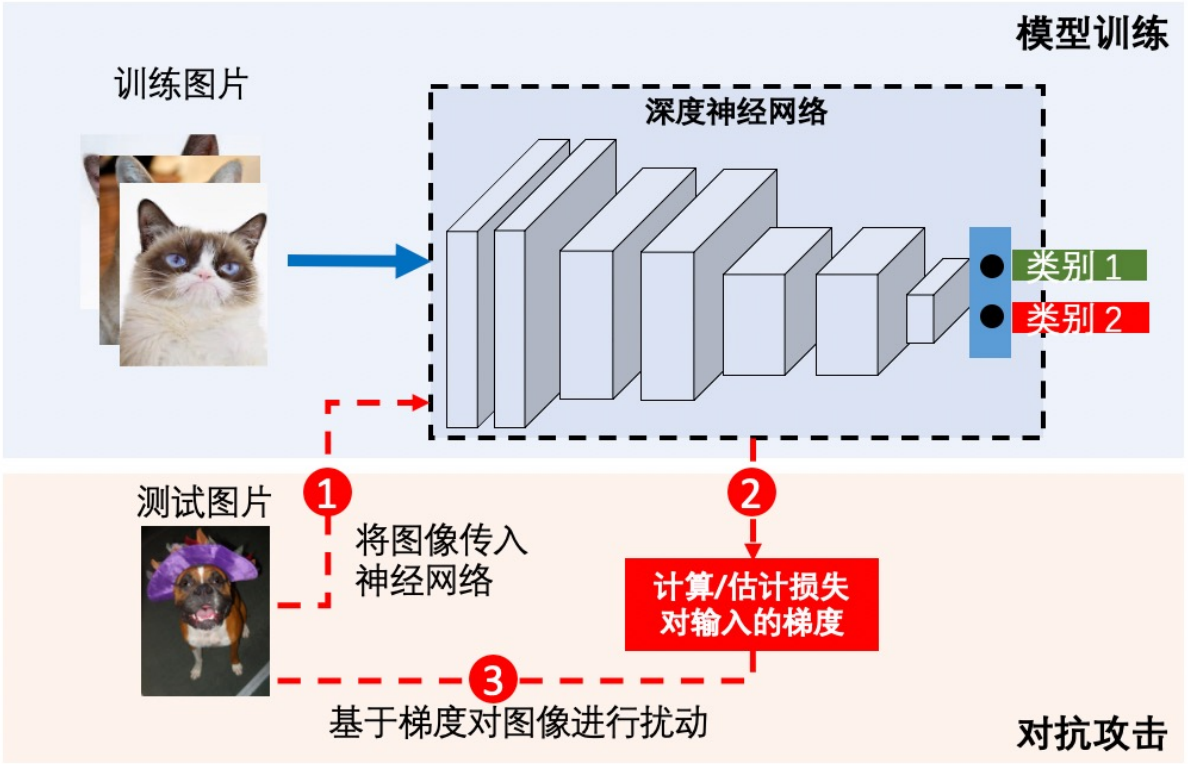
Principles for AED



- All anomaly detection methods can be applied for AED

Principles for AED

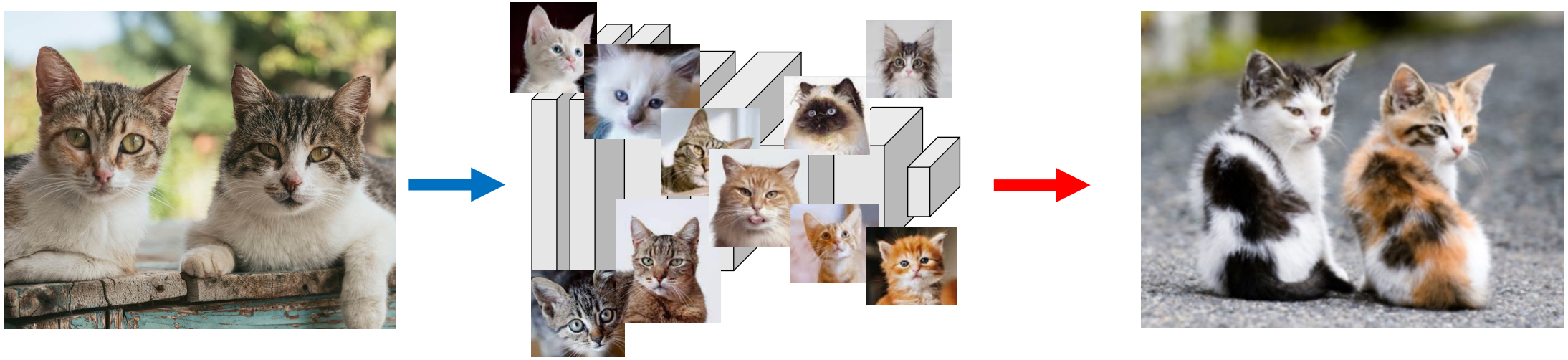
- Input statistics
- Manual features
- Training data
- Attention map
- Transformation
- Mixup
- Denoising
- …



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

☐ Use as much information as you can

Principles for AED



Twins

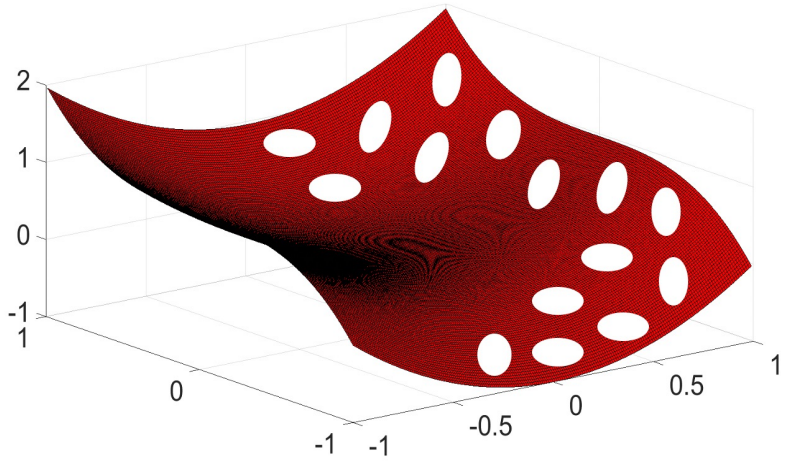
Extremely close to the clean sample

Strangers

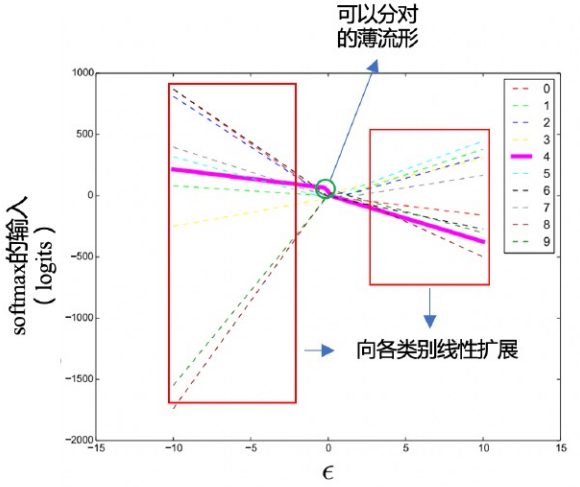
Far away in prediction

- Leverage unique characteristics of adversarial examples

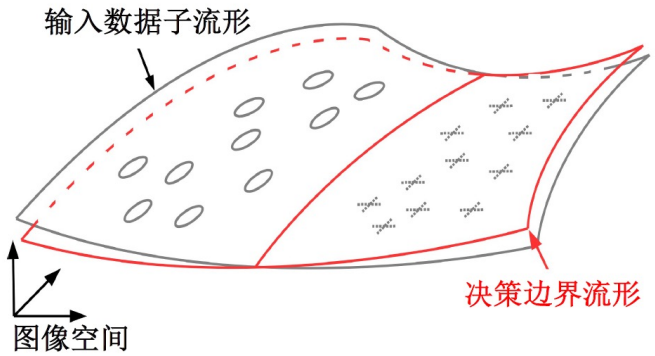
Principles for AED



High dimensional pockets



Local linearity



Tilting boundary

☐ Build detectors based on existing understandings

Principles for AED

It's is still feature engineering!

Challenges in AED

- The **diversity** of adversarial examples used for training the detectors determine the detection performance
- Detectors are also machine learning models: 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

Grosse et al. On the (Statistical) Detection of Adversarial Examples, arXiv:1702.06280

Secondary Classification Methods

Adversarial Classification (对抗分类)

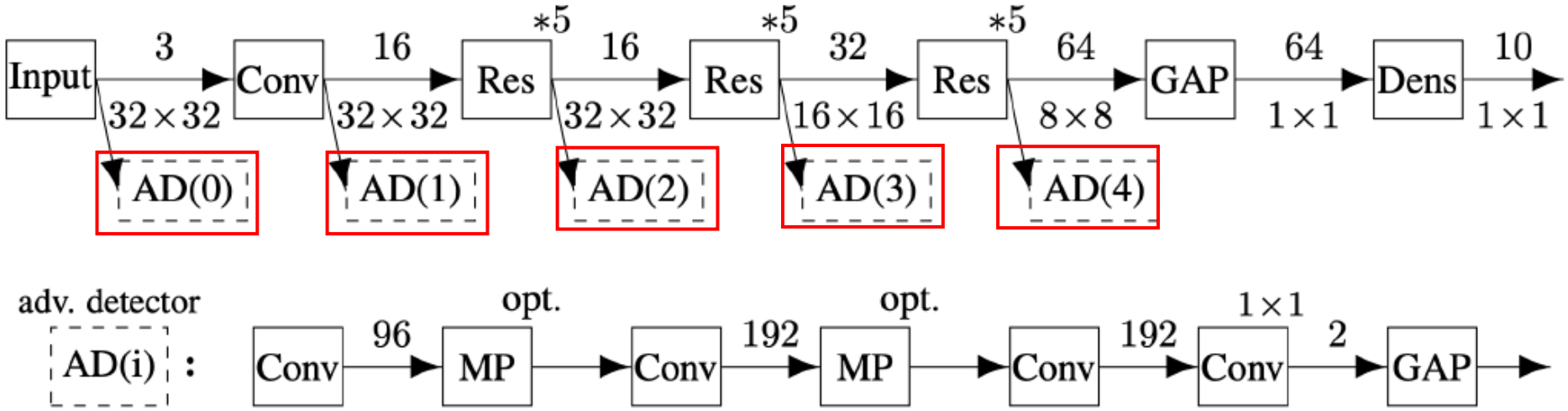
1. 在正常训练集 D_{train} 上训练得到模型 f
2. 基于 D_{train} 对抗攻击模型 f 得到对抗样本集 D_{adv}
3. 将 D_{train} 标记为 0 类别，将 D_{adv} 标注为 1 类别
4. 在 $D_{\text{train}} \cup D_{\text{adv}}$ 上训练得到二分类检测器 g

- Clean samples as class 0, adversarial as class 1

Gong et al. Adversarial and clean data are not twins, arXiv:1704.04960

Secondary Classification Methods

Cascade Classifiers (级联分类器)



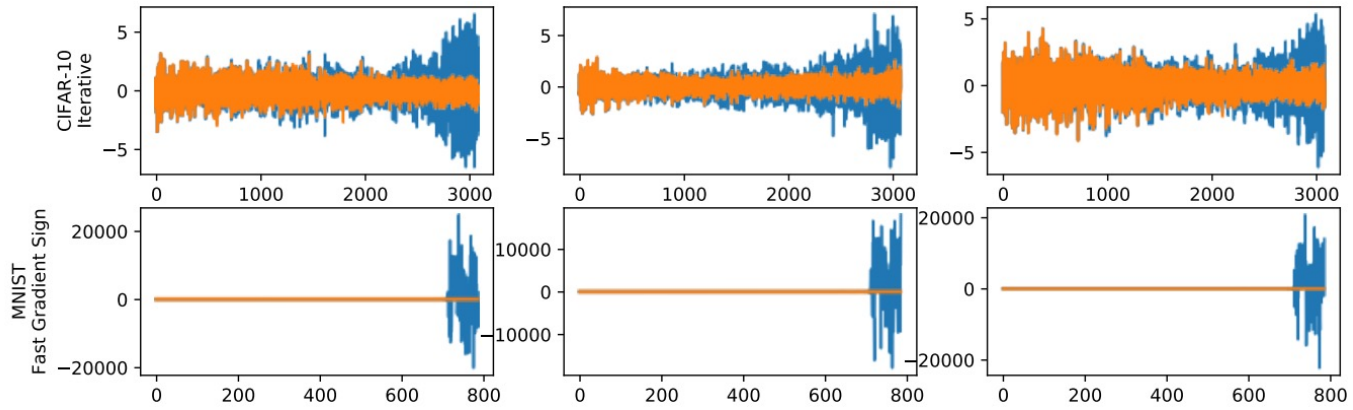
- Training a detector for each intermediate layer

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

Existing Methods

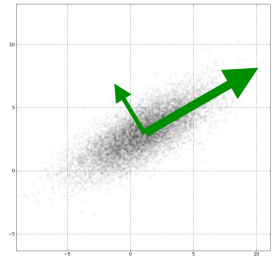
- Secondary Classification Methods (二级分类法)
- **Principle Component Analysis (主成分分析法，PCA)**
- Distribution Detection Methods (分布检测法)
- Prediction Inconsistency (预测不一致性)
- Reconstruction Inconsistency (重建不一致性)
- Trapping Based Detection (诱捕检测法)

Principle Component Analysis (PCA)



Blue: a clean sample

Yellow: an adv example

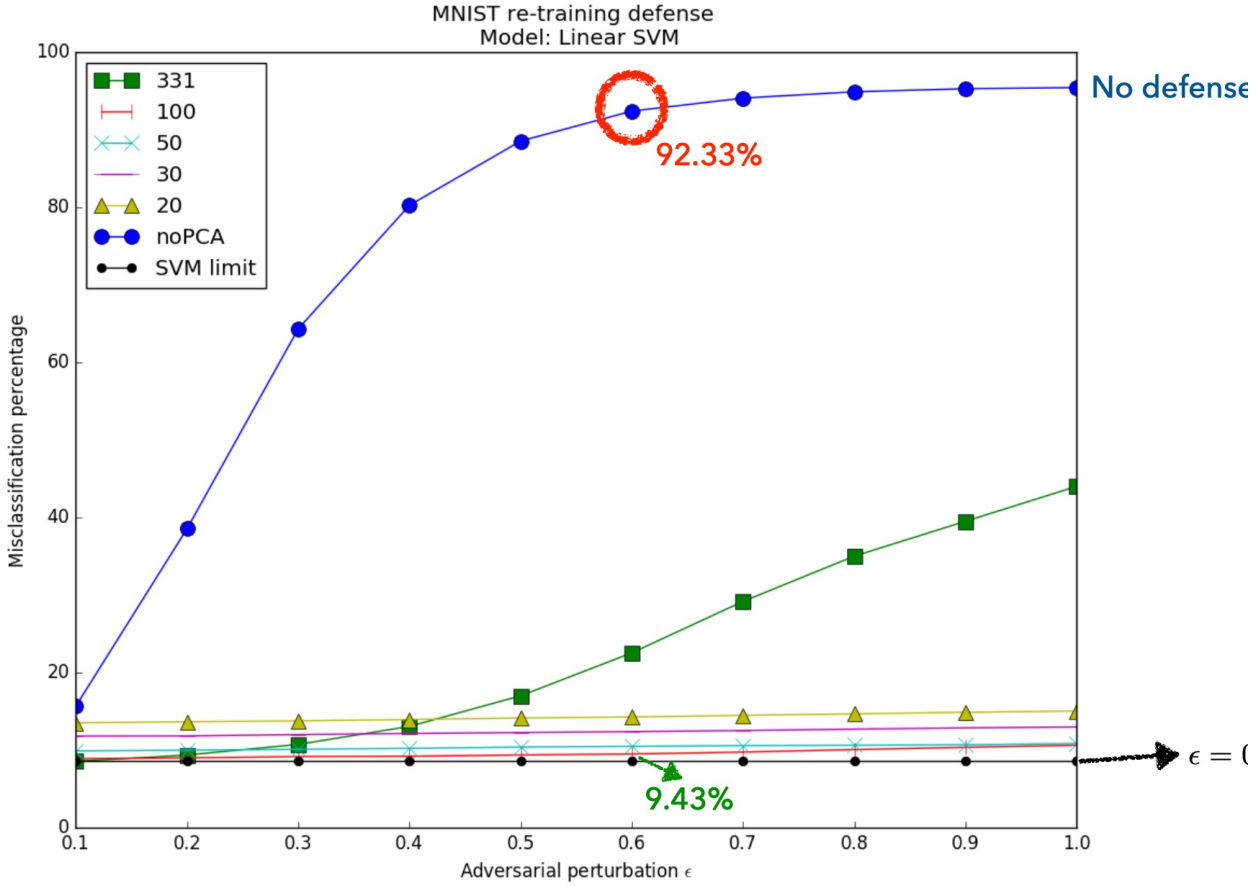


An artifact caused by the black background

- 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." *AISec*. 2017.

Dimensionality Reduction



- Train on PCA reduced data

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)

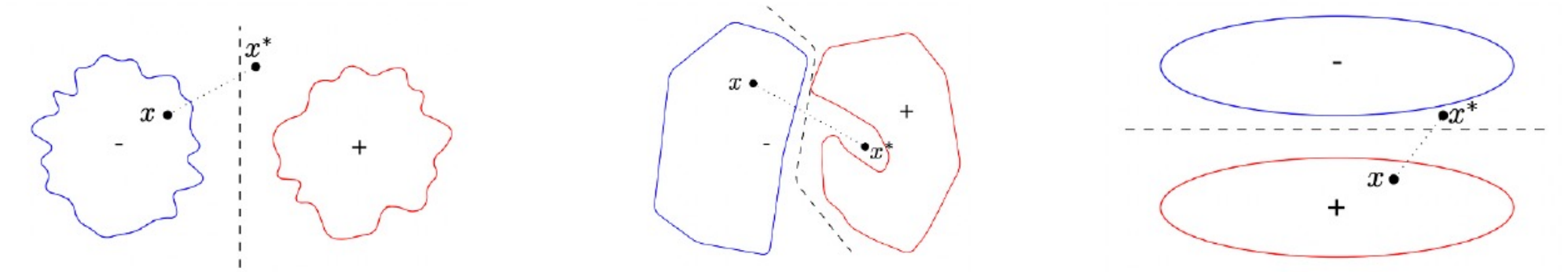
1. 在 D_1 和 D_2 上计算 $a = MMD(\mathcal{K}, D_1, D_2)$;
2. 对 D_1 和 D_2 中的样本顺序做随机打乱得到对应的 D_1' 和 D_2';
3. 在 D_1' 和 D_2' 上计算 $b = MMD(\mathcal{K}, D_1', D_2')$;
4. 如果 $a < b$ 则拒绝原假设，即 D_1 和 D_2 来自不同分布;
5. 重复执行步骤 1-4 很多次（1 万次），计算原假设被拒绝的比例作为 p-值。

Two datasets: D_1 vs. D_2

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

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)

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

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.

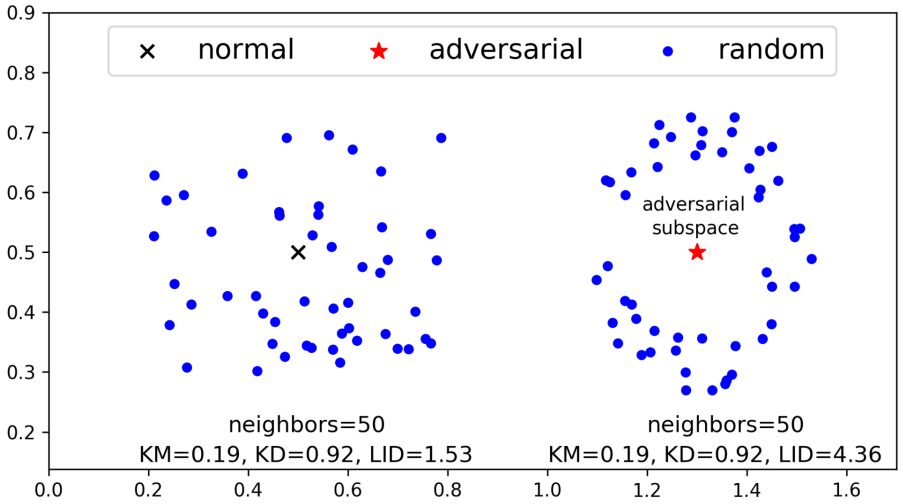
Local Intrinsic Dimensionality (LID)

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

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

wherever the limit exists.



Adversarial examples are in high dimensional subspaces

Local Intrinsic Dimensionality (LID)

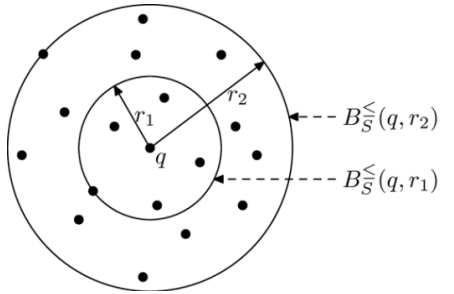
Adversarial Subspaces and Expansion Dimension:

Expansion Dimension:

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

$$\frac{\mathrm{V}_2}{V_1} = \left(\frac{r_2}{r_1}\right)^m \Rightarrow m = \frac{\ln(V_2/V_1)}{\ln(r_2/r_1)}$$

- 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.



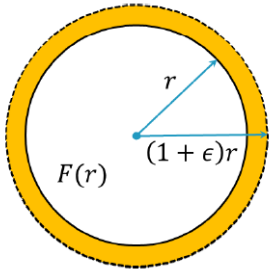
Local Intrinsic Dimensionality (LID)

Estimation of LID:

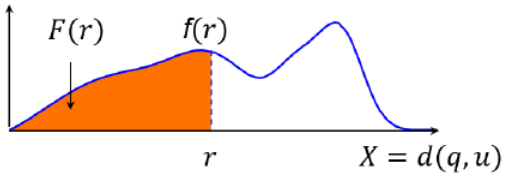
- Hill (MLE) estimator (*Hill 1975, Amsaleg et al. 2015*):

$$\widehat{\mathrm{LID}}(x) = -\left(\frac{1}{k}\sum_{i=1}^{k}\log\frac{r_i(x)}{r_k(x)}\right)^{-1},$$

r_i is the distance of x to its i^{th} nearest neighbor.



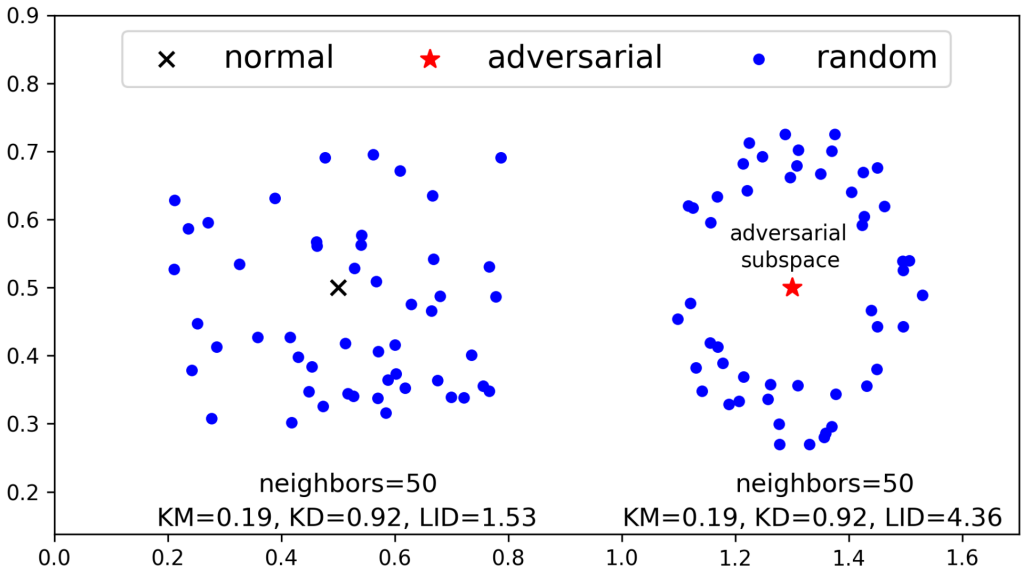
- Based on Extreme Value Theory:
  - Nearest neighbor distances are extreme events.
  - Lower tail distribution follows Generalized Pareto Distribution (GPD).



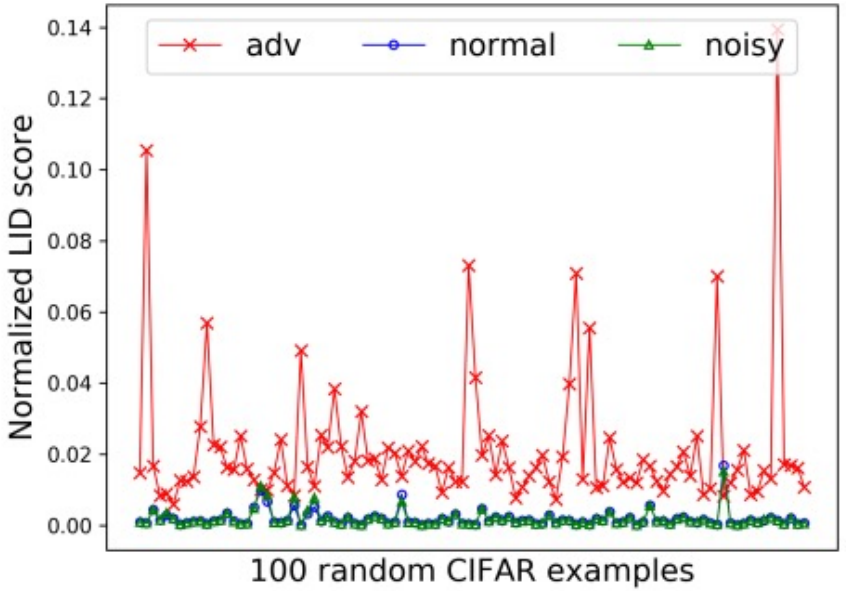
Local Intrinsic Dimensionality (LID)

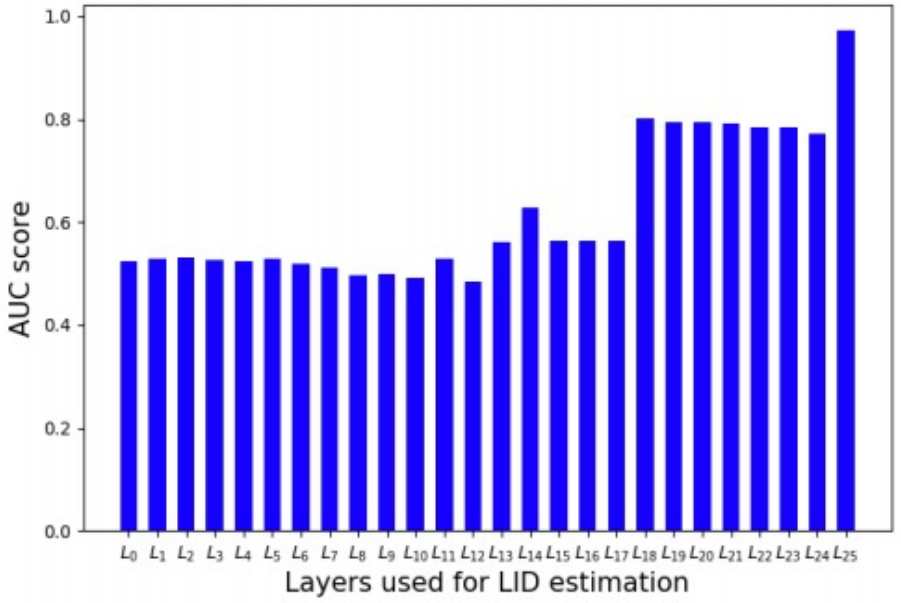
Interpretation of LID for Adversarial Subspaces:

- 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**

Local Intrinsic Dimensionality (LID)

Algorithm 7.1 训练 LID 对抗样本检测器

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

初始化检测器训练集： LID_{neg}=[], LID_{pos}=[]
for B_{norm} in $\boldsymbol{x}$ **do**
    B_{adv} := 对抗攻击本批样本 B_{norm}
    $N = |B_{\text{norm}}|$
    初始化 LID 特征集 LID_{norm}, LID_{adv} 为全零矩阵（维度均为 $[n, l]$）
    for i in $[1, l]$ **do**
        抽取中间层特征： $A_{\text{norm}} = f^i(B_{\text{norm}})$， $A_{\text{adv}} = f^i(B_{\text{adv}})$
        for j in $[1, n]$ **do**
            $\text{LID}_{\text{norm}}[j, i] = -\left(\frac{1}{k}\sum_{i=1}^{k} \log \frac{r_i(A_{\text{norm}}[j], A_{\text{norm}})}{r_k(A_{\text{norm}}[j], A_{\text{norm}})}\right)^{-1}$
            $\text{LID}_{\text{adv}}[j, i] = -\left(\frac{1}{k}\sum_{i=1}^{k} \log \frac{r_i(A_{\text{adv}}[j], A_{\text{norm}})}{r_k(A_{\text{adv}}[j], A_{\text{norm}})}\right)^{-1}$
    LID_{neg}.append(LID_{norm})， LID_{pos}.append(LID_{adv})
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**

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

An Improved Detector of LID

$$\widehat{\mathrm{LID}}(x) = -\left(\frac{1}{k}\sum_{i=1}^{k}\log\frac{r_i(x)}{r_k(x)}\right)^{-1}$$

$$\overrightarrow{\mathrm{LID}}(x)[i] = -\log\frac{r_i(x)}{r_k(x)}$$

https://arxiv.org/pdf/2212.06776.pdf

An Improved Detector of LID

Table 1: Results. Comparison of the original LID method with our proposed multiLID on different datasets. We report the AUC and F1 score as mean and variance over three evaluations with randomly drawn test samples.

Attacks	CIFAR10				CIFAR100				ImageNet	
	WRN 28-10		VGG16		WRN 28-10		VGG16		WRN 50-2	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
original LID [?]										
FGSM	99.5±0.2	97.3±7.0	90.1±13.4	83.2±13.9	100.0±0.0	99.6±0.0	83.6±11.7	75.1±21.3	89.1±4.4	81.6±7.8
BIM	96.9±1.5	90.5±4.2	92.8±2.1	86.5±3.3	98.2±0.0	92.2±0.0	84.8±10.0	75.6±11.1	100.0±0.0	98.9±1.0
PGD	99.1±0.3	95.3±1.8	97.5±0.0	94.6±0.5	98.0±0.0	93.5±0.0	91.8±0.8	83.9±0.4	100.0±0.0	100.0±0.0
AA	96.7±0.2	89.4±3.4	90.0±1.3	81.6±1.8	99.2±0.1	96.5±0.4	86.8±9.8	78.6±2.3	100.0±0.0	99.8±0.1
DF	94.7±31.9	88.7±55.4	87.3±4.2	77.2±4.6	60.7±0.0	56.4±0.0	60.5±2.8	56.1±1.8	60.3±2.2	56.5±2.9
CW	91.2±63.6	83.9±54.5	85.2±1.7	75.3±3.5	56.3±0.1	52.5±2.6	66.0±6.1	61.0±0.9	62.0±0.5	59.0±2.0
multiLID + improved layer setting + RF or short: **multiLID (ours)**										
FGSM	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	100.0±0.0	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**
BIM	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	100.0±0.0	**100.0±0.0**
PGD	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	100.0±0.0	**100.0±0.0**
AA	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	100.0±0.0	**100.0±0.0**
DF	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**
CW	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**

https://arxiv.org/pdf/2212.06776.pdf

Mahalanobis Distance (MD)

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

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

μ: sample mean in Q
Σ: covariance matrix

- **The MD of between two data points:**

$$d_M(\boldsymbol{x}_i, \boldsymbol{x}_2) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_2)^\top \Sigma^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_2)}$$

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

Mahalanobis Distance (MD)

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

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

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

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

$f^{\mathrm{L-2}}$: 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 $\boldsymbol{x}$，逻辑回归检测器权重 α_l，噪声大小 ϵ 以及高斯分布参数 $\{\mu_{l,c}, \Sigma_l : \forall l, c\}$

初始化分数向量：$M(\boldsymbol{x}) = [M_l : \forall l]$

for 每一层 $l = 1, \cdots, L$ **do**

  寻找最近的类别：$\hat{c} = \arg\min_c (f^l(\boldsymbol{x}) - \mu_{l,c})^\top \Sigma_l^{-1} (f^l(\boldsymbol{x}) - \mu_{l,c})$

  向样本中添加噪声：$\hat{\boldsymbol{x}} = \boldsymbol{x} - \epsilon \cdot \text{sign}\left(\Delta_x (f^l(\boldsymbol{x}) - \mu_{l,c})^\top \Sigma_l^{-1} (f^l(\boldsymbol{x}) - \mu_{l,c})\right)$

  计算置信度：$M_l = \max_c -(f^l(\boldsymbol{x}) - \mu_{l,c})^\top \Sigma_l^{-1} (f^l(\boldsymbol{x}) - \mu_{l,c})$

输出: 样本 $\boldsymbol{x}$ 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.

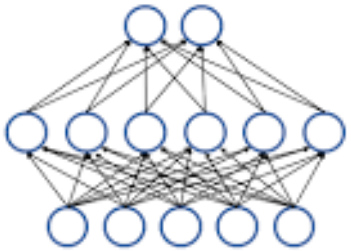
Existing Methods

- 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)

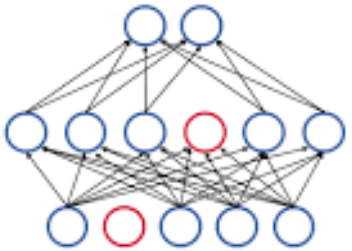
Bayes Uncertainty

Bayesian Uncertainty (BU)

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



Without Dropout



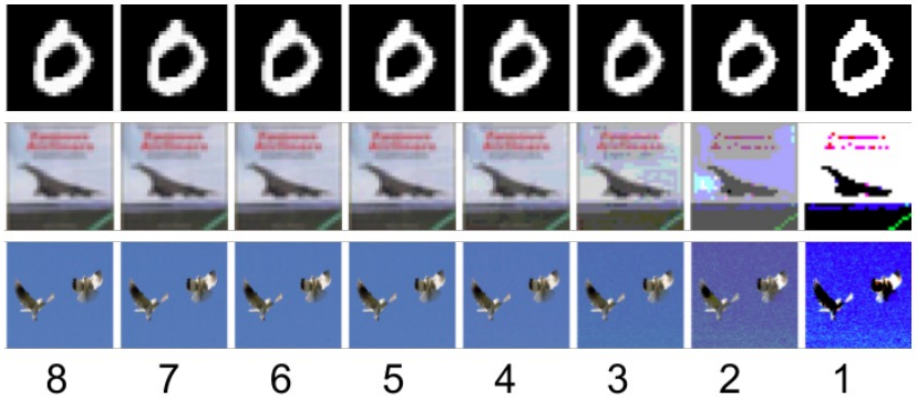
With Dropout

Use test time dropout to get randomized networks

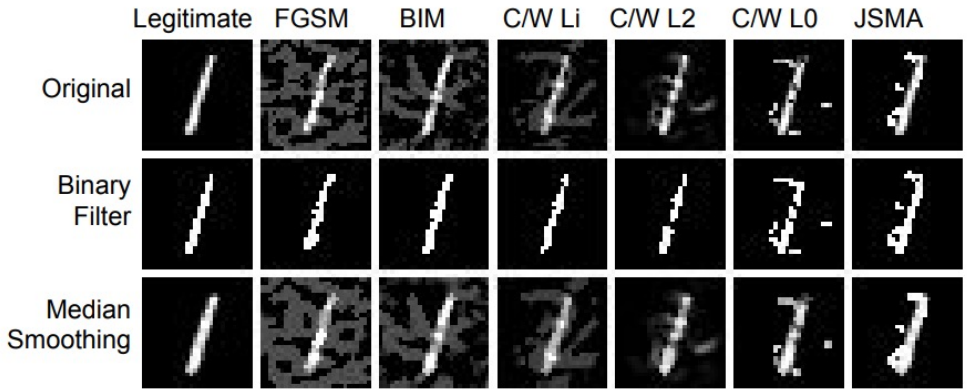
T: the number of randomization.

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

Feature Squeezing



Bit depth reduction

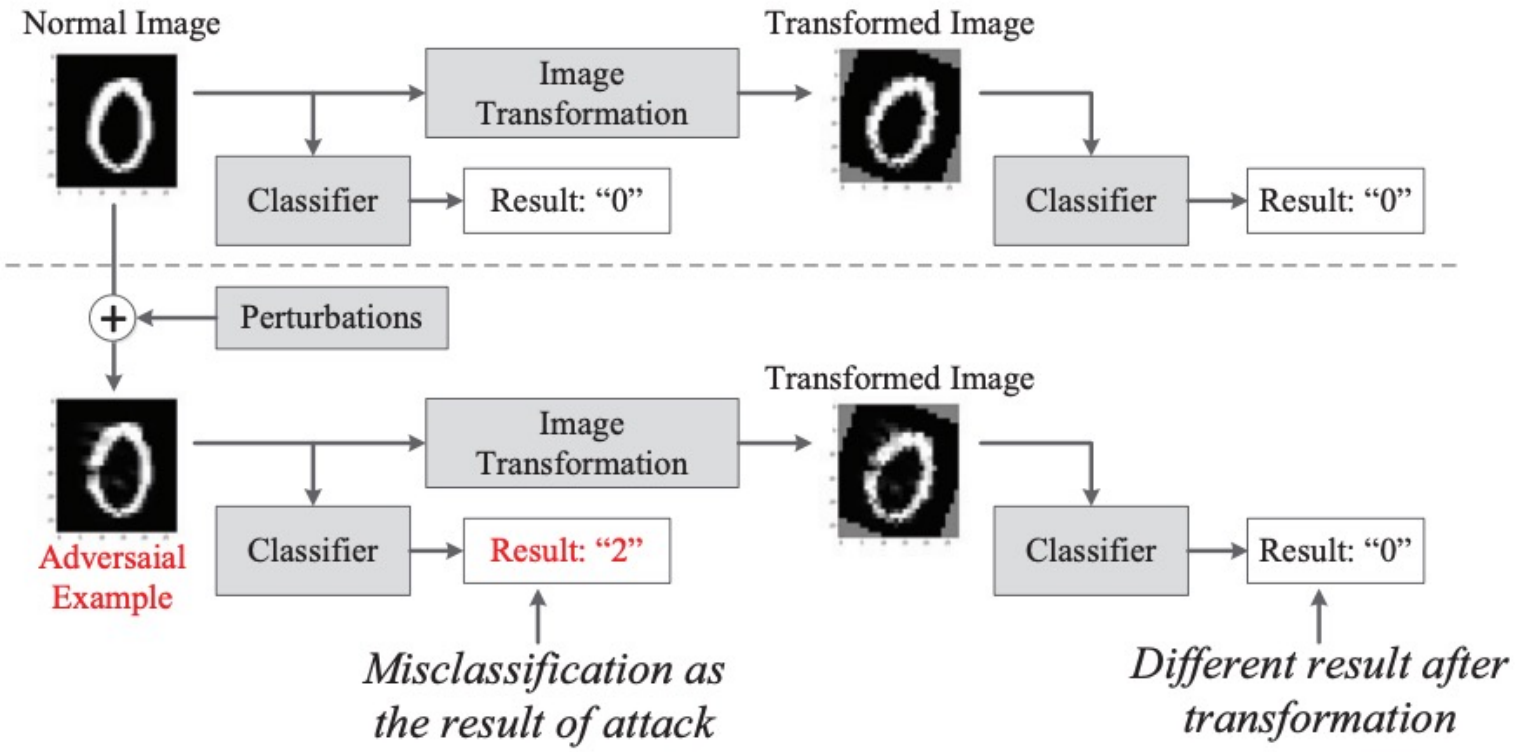


Squeezing clean and adv examples

- **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).

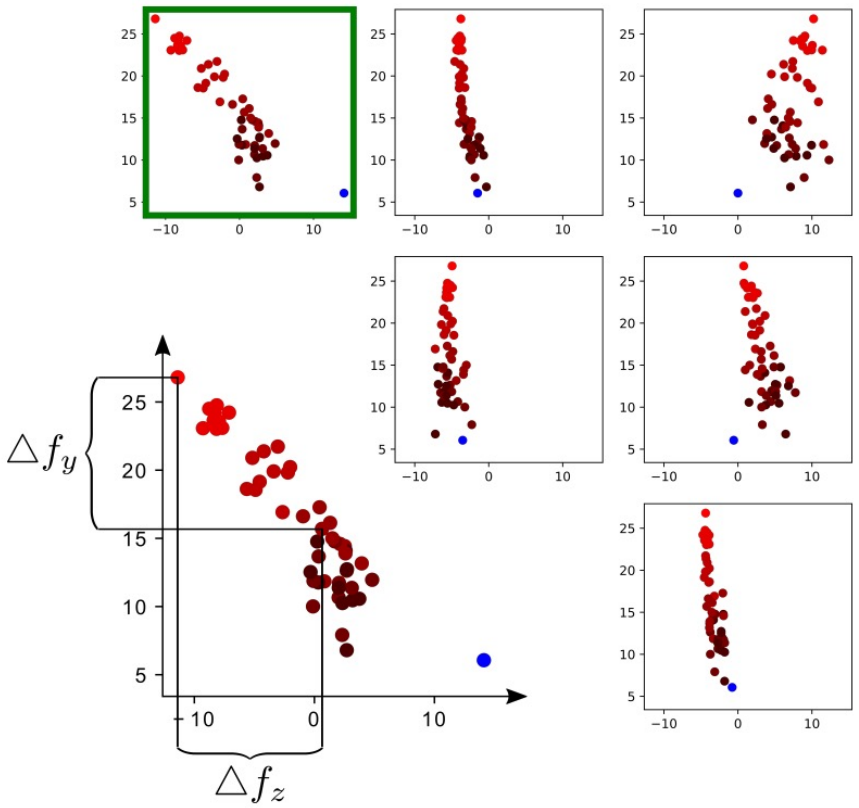
Random Transformation



☐ **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

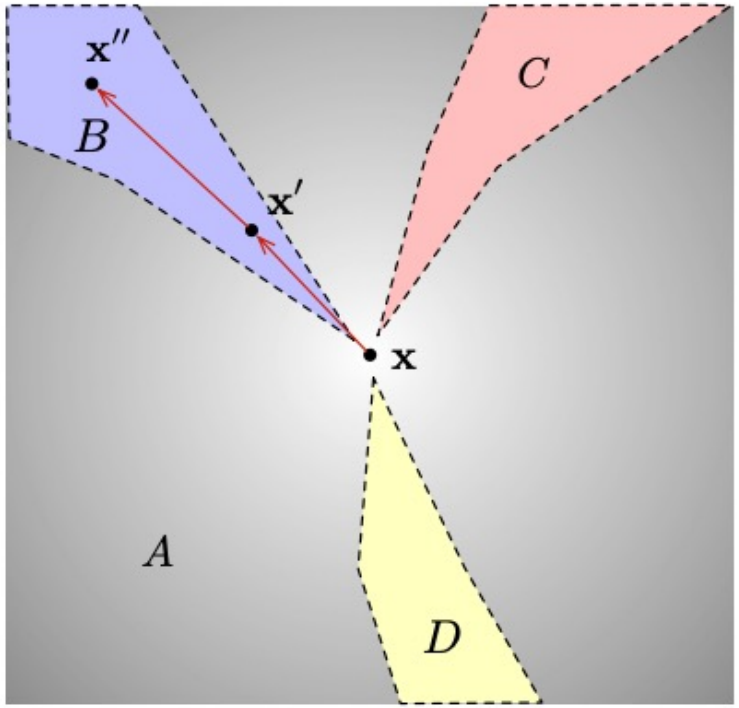
☐ **Add random noise to the input**

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

$$f(x') \approx f(x) \text{ ??}$$

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

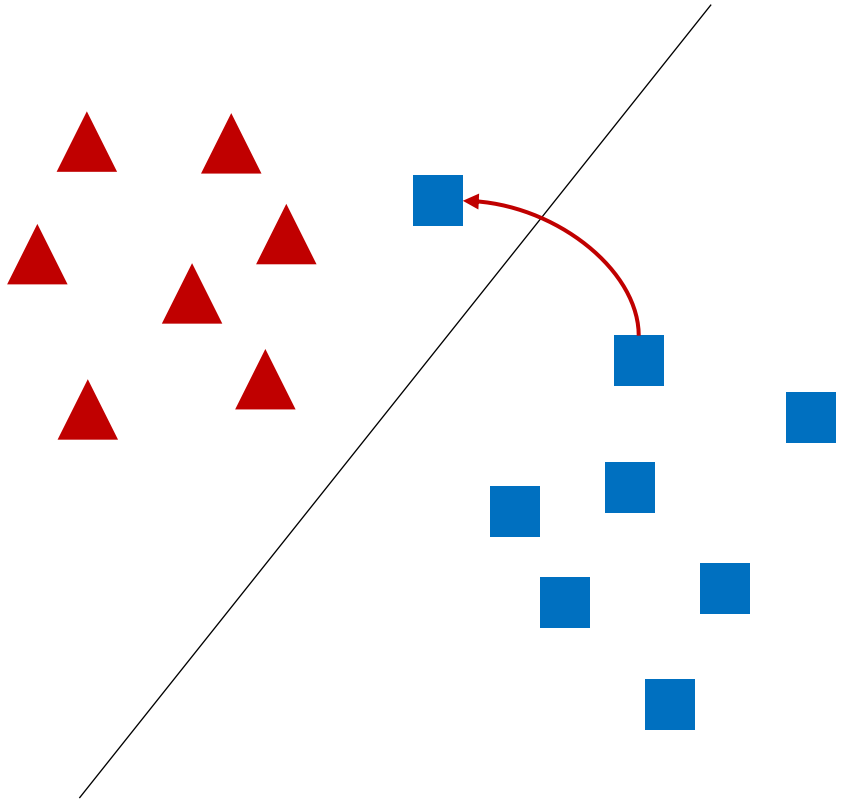
- 测试准则1：随机噪声不会改变预测结果
- 测试准则1 ：再次攻击需要更多的扰动

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)

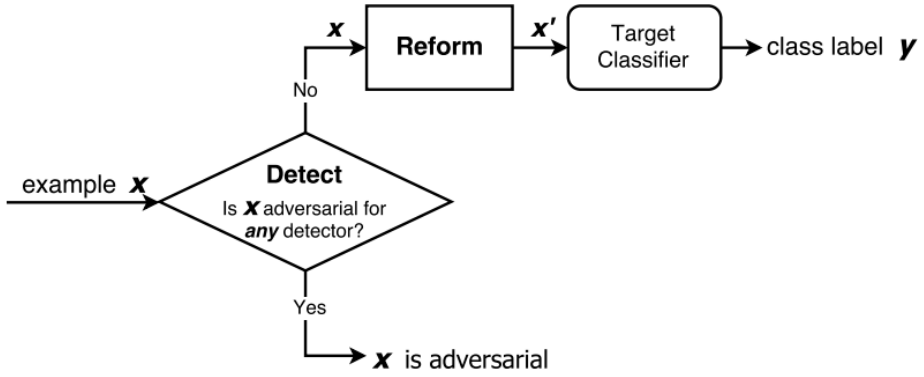
Detector-Reformer



- 原则：对抗样本无法重建

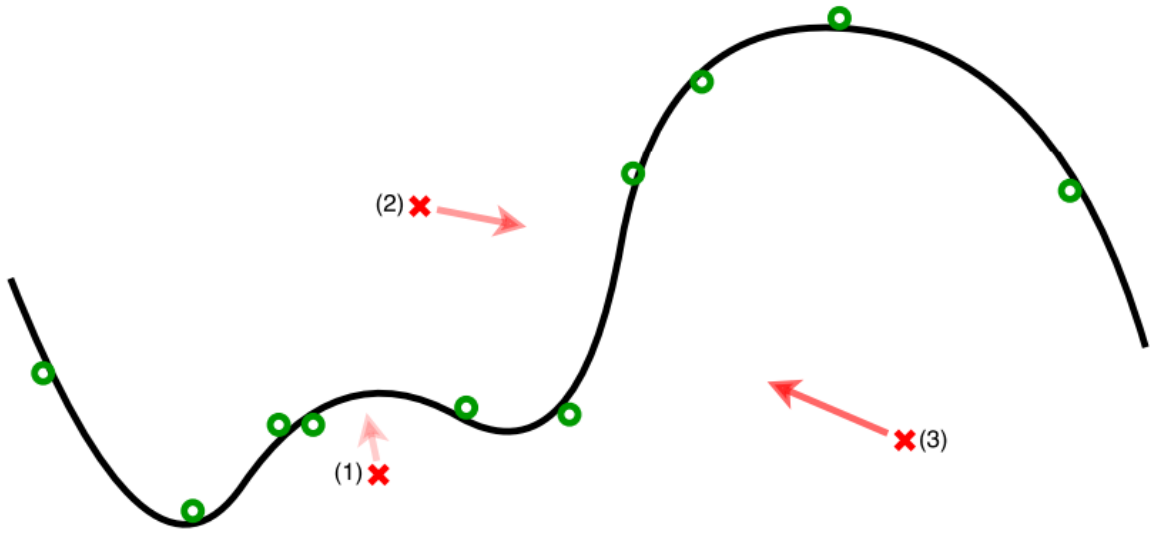
$$E(\boldsymbol{x}) = \|\boldsymbol{x} - AE(\boldsymbol{x})\|_p$$

AE: Autoencoder
E(x): reconstruction error



Detector-Reformer

How the reformer works?

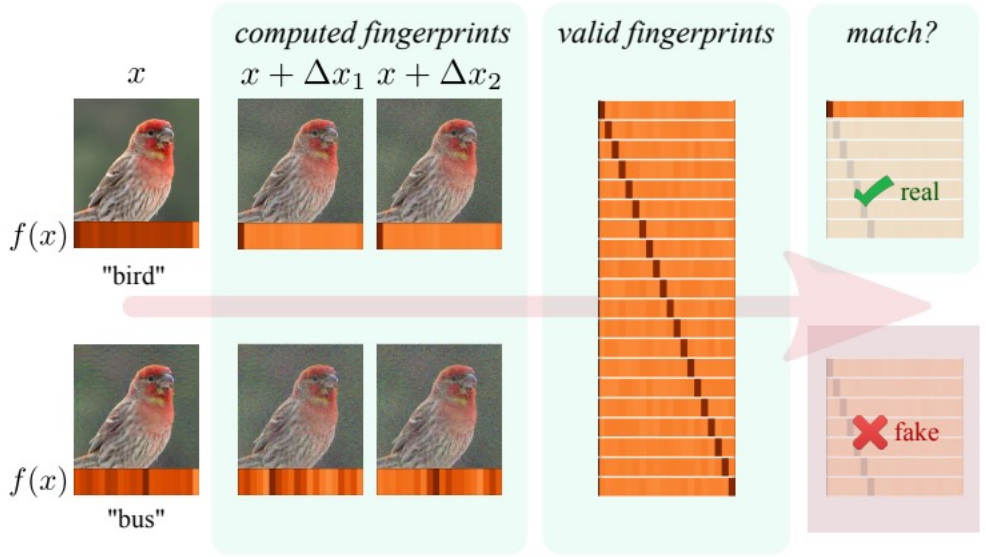


绿色：正常样本
红色x：对抗样本
红色箭头：自编码器

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:

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

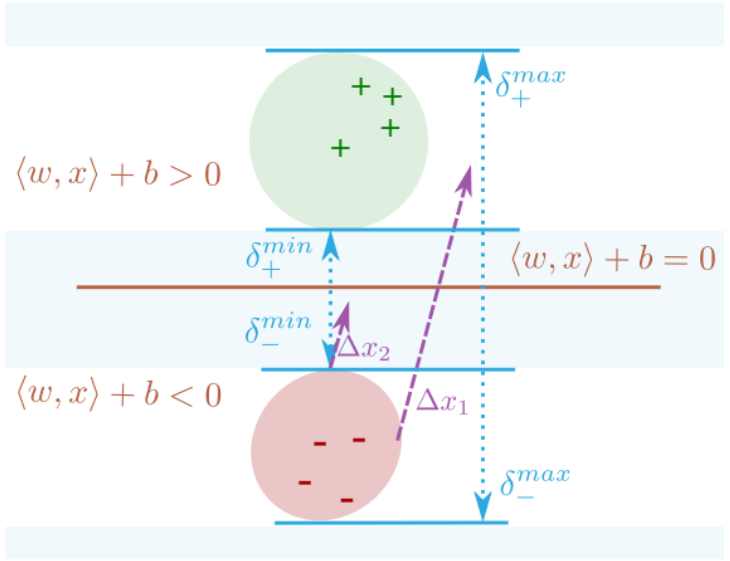
Dathathri, Sumanth, et al. "Detecting adversarial examples via neural fingerprinting." arXiv:1803.03870 (2018).

Neural Fingerprinting (NFP)

How to verify the fingerprint?

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

Δx_i is class-independent noise



Benchmarking

Results

Attack	KDE_DR	LID_DR	NSS_DR	FS_DR	MagNet_DR	NIC_DR	MultiLID_DR
fgsm_0.03125	66.47	50.0	84.33	52.51	69.58	94.32	92.81
fgsm_0.0625	63.96	78.98	92.87	49.84	94.31	94.79	93.46
fgsm_0.125	61.44	83.97	92.85	49.27	94.33	94.82	93.86
bim_0.03125	69.43	50.11	67.42	93.18	52.25	90.55	92.9
bim_0.0625	69.05	66.21	86.82	93.98	93.93	92.37	93.54
bim_0.125	69.01	92.1	92.6	93.99	94.11	94.44	94.05
pgdi_0.03125	71.04	50.11	69.85	93.81	53.52	90.72	92.86
pgdi_0.0625	70.95	68.06	89.41	93.99	94.08	94.07	93.59
pgdi_0.125	70.37	92.83	92.78	93.99	94.11	94.68	94.46
cwi	75.34	50.0	51.73	48.16	50.28	87.74	98.02
deepfool	81.68	50.0	50.44	48.35	50.05	93.11	98.06
spatial transofrmation attack	68.88	83.77	78.01	47.71	52.41	91.33	99.67
square attack	75.36	80.76	48.89	47.72	98.52	94.67	99.22
adversarial patch	52.43	64.11	87.39	48.67	80.15	94.58	99.76

谢谢！