AI Model Copyright Protection

马兴军，复旦大学 计算机学院

Recap: week 13

- Biases in Current AI Models
- Definition and Types of Biases
- Fair Machine Learning
- AI Ethics

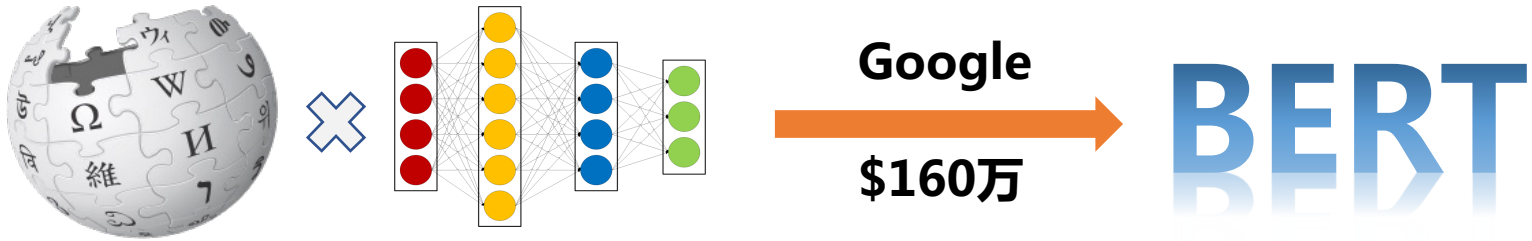
Group Registration

可信机器学习组队注册

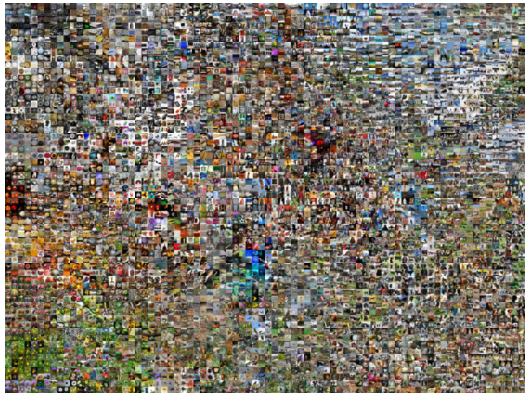
扫一扫二维码打开或分享...

https://docs.qq.com/form/page/DU3N1ZUFBUnRJVnFJ#/result

AI 模型是宝贵的财产



大规模、高性能的AI模型训练耗费巨大



数据资源

计算资源

人力资源

模型窃取动机

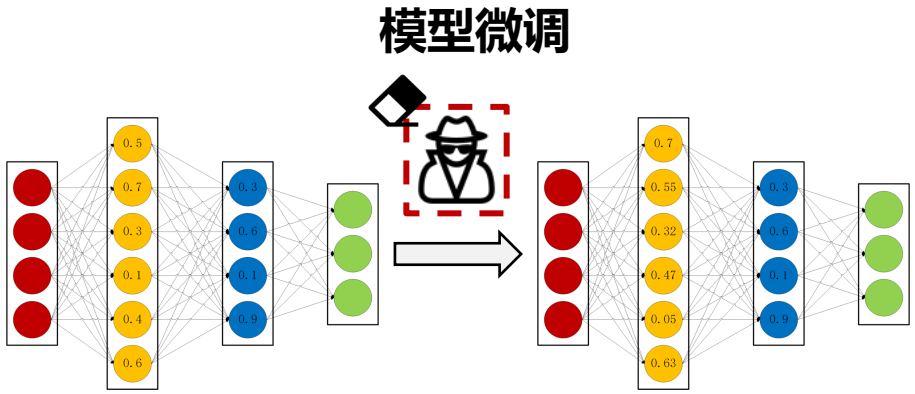
宝贵的 AI 模型

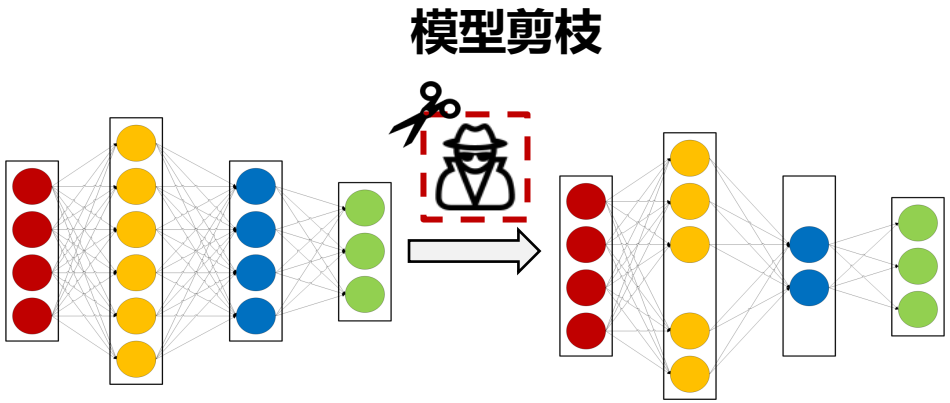
模型窃取

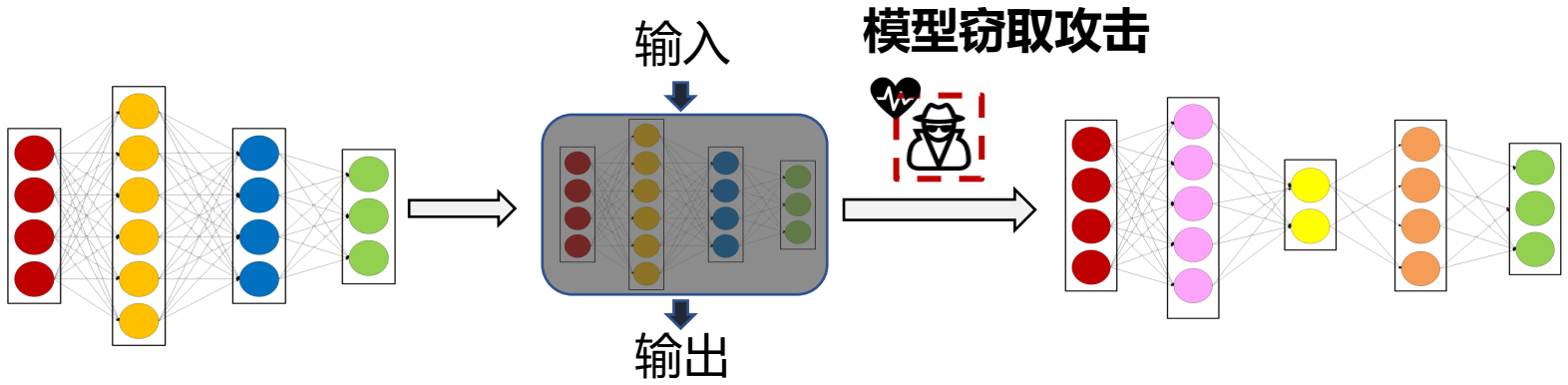
为其所用

- 巨大的商业价值
- 尽量保持模型性能
- 不希望被发现

模型窃取方式





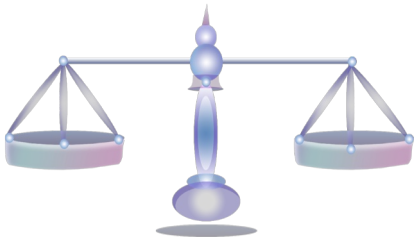


Stealing machine learning models via prediction APIs, USENIX Security, 2016; Practical black-box attacks against machine learning, ASIACCS, 2017; Knockoff nets. Stealing functionality of black-box models, CVPR, 2019; Maze: Data-free model stealing attack using zeroth-order gradient estimation, CVPR, 2021;

版权保护的目标

模型版权识别

Copy, Right?

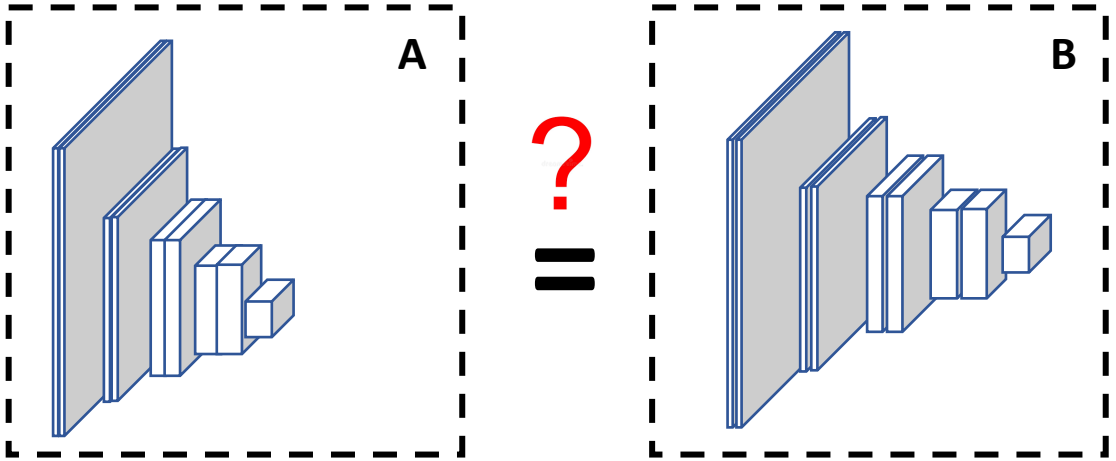


- 准确识别出窃取模型 (copy)
- 尽可能高效、鲁棒

- 尽可能保持模型性能
- 不希望被发现 (not copy)

版权保护的两个阶段

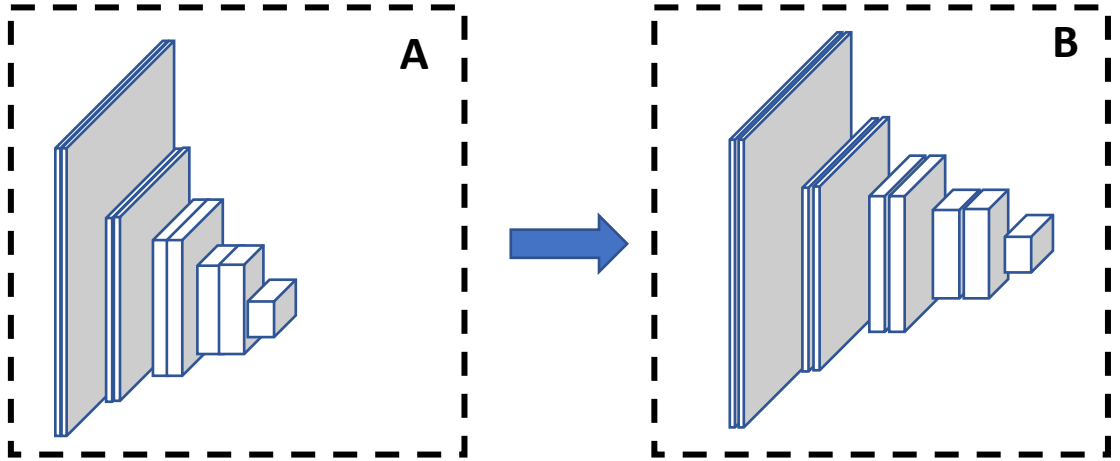
The same model?



一致性验证

This is what we are doing.

B is derived from A?



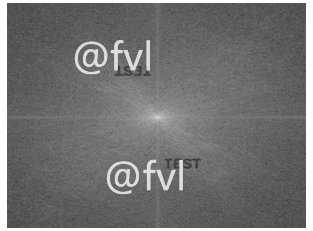
模型溯源

This is what we should do.

模型水印技术

传统图像水印：将所有者信息**嵌入**媒体

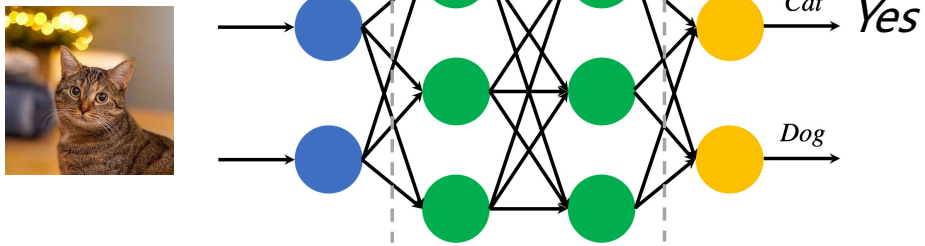
Spectrum



不同方法的表现和鲁棒性存在一定差异

模型水印技术

水印信息载体发生变化：模型参数



参数矩阵

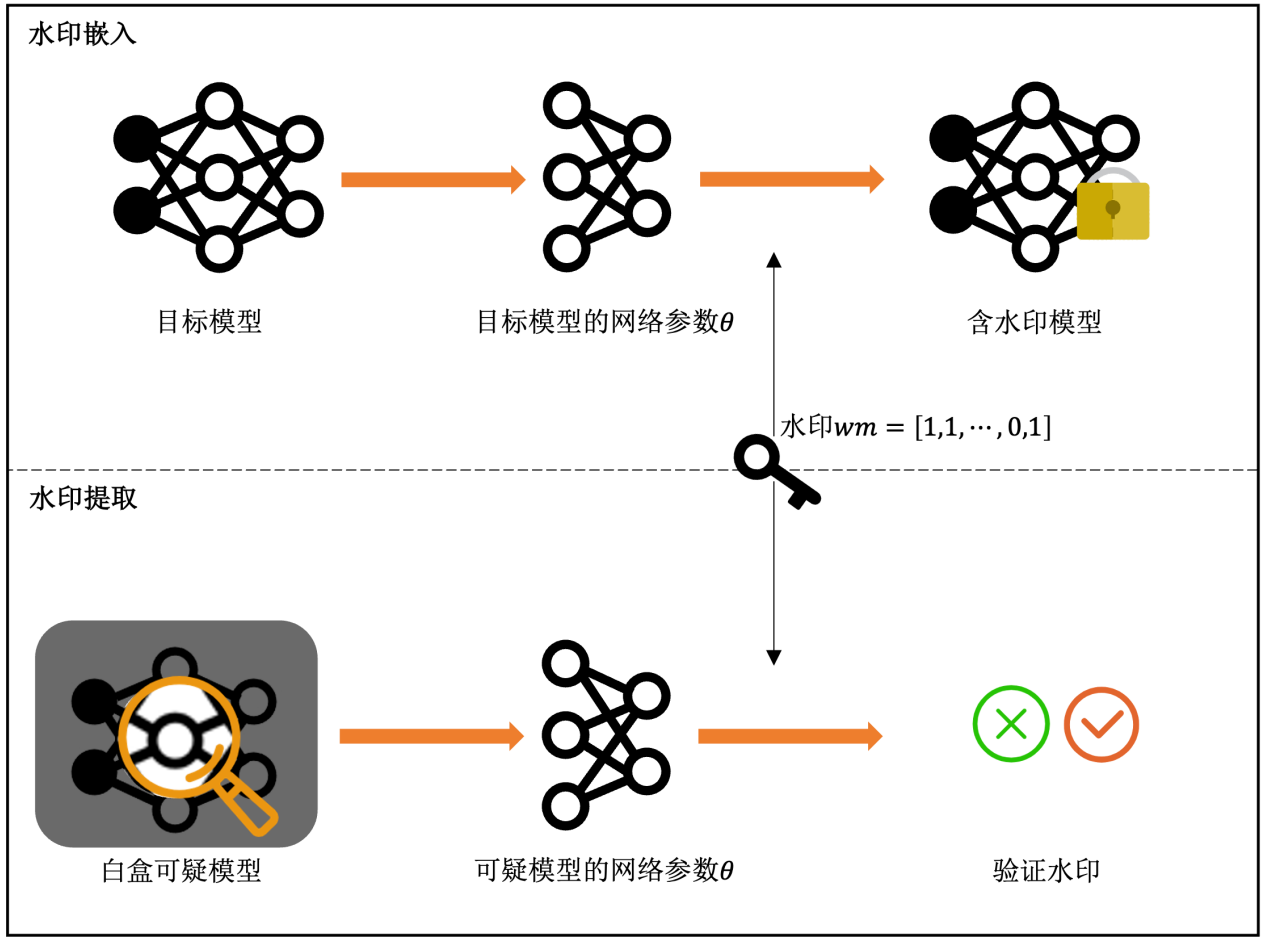
```
<tf.Variable 'block1_conv1/bias:0' shape=(6,) dtype=float32, numpy=
array([-0.0179275 ,  0.01634618,  0.00635687, -0.00039525,  0.02307029,
        0.01255256], dtype=float32)>,
<tf.Variable 'block2_conv1/kernel:0' shape=(5, 5, 6, 16) dtype=float32, numpy=
array([[[[-1.76457345e-01, -4.58735600e-02,  1.17505528e-01, ...,
          -9.38355997e-02, -2.04667822e-02,  1.87988058e-02],
         [-9.93207395e-02,  1.18925497e-01,  4.04093266e-02, ...,
          -1.51014671e-01, -1.60283774e-01,  8.59537944e-02],
         [-3.28751802e-02, -1.38621703e-02, -1.22857407e-01, ...,
           1.60882026e-01,  6.79167584e-02, -1.13412902e-01],
         [ 2.13318810e-01,  2.93371230e-02,  2.61021988e-03, ...,
          -1.32674173e-01, -4.13975678e-02,  1.07933886e-01],
         [-9.86196920e-02,  7.49983713e-02,  7.63874054e-02, ...,
          -1.97056949e-01, -1.45023346e-01, -7.27430880e-02],
         [-1.25542030e-01, -9.32724625e-02,  2.27828339e-01, ...,
          -2.46503651e-02,  5.29526435e-02,  7.62251113e-03]],

        [[-9.31718200e-02,  3.15065421e-02,  1.53141439e-01, ...,
          -2.05002844e-01,  5.79415262e-02, -8.29598457e-02],
         [-1.06220163e-01,  3.28049511e-02,  3.15736942e-02, ...,
          -1.83010474e-01, -1.69398963e-01, -2.05086600e-02],
         [ 2.51061153e-02,  3.18859890e-02, -1.38661385e-01, ...,
          -7.84005970e-02, -2.71614478e-03,  6.92593977e-02],
         [-1.73594092e-03,  1.54581293e-02, -2.80732084e-02, ...,
          -1.00928947e-01,  7.41603831e-03,  1.41286582e-01],
         [-6.83757067e-02, -9.88088697e-02,  1.45139262e-01, ...,
          -1.26678377e-01, -5.58072440e-02,  1.31867416e-02],
         [-4.23093103e-02,  1.94922481e-02,  1.77184254e-01, ...,
          -7.48334676e-02, -5.38966134e-02,  5.56072779e-02]],
```

能不能直接将一行参数改成自己定义的特殊签名？

AI 模型的实用性需求 (Fidelity)：水印不能破坏模型功能

模型水印技术 – 白盒水印

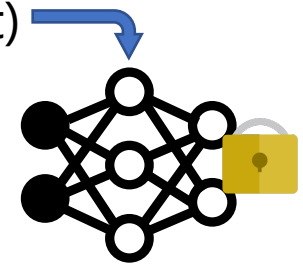


Uchida, Yusuke, et al. Embedding watermarks into deep neural networks. ACM ICMR, 2017

模型水印技术 – 白盒水印

Step1. 水印嵌入

$\boldsymbol{b}$=[1,0,0,1…,0,1,1] (512bit)



Watermarked Model

参数向量 θ
嵌入矩阵 X

在训练过程中，添加参数惩罚项，结合嵌入矩阵将信息（钥匙）嵌入到模型参数中。

Step2. 水印提取

$$b_j = Step(\sum_i X_{ji}\theta_i)$$

基于可疑模型的权重矩阵提取水印，
和 $\boldsymbol{b}$ 进行比较，计算BER (Bit Error Rate)

模型水印技术 – 白盒水印

水印嵌入的目标损失函数：

$$\mathcal{L} = \mathcal{L}_0(f(\boldsymbol{x}), y) + \lambda \mathcal{R}(\theta, \boldsymbol{b})$$

$$\mathcal{R}(\theta, \boldsymbol{b}) = -\sum_{j=1}^{T} (\boldsymbol{b}_j \log(\boldsymbol{z}_j) + (1 - \boldsymbol{b}_j) \log(1 - \boldsymbol{z}_j))$$

$$\boldsymbol{z}_j = \sigma(\sum_i X_{ji} \cdot \theta_i)$$

在训练过程中，添加参数惩罚项，结合嵌入矩阵将信息（钥匙）嵌入到模型参数中。

参数向量： θ

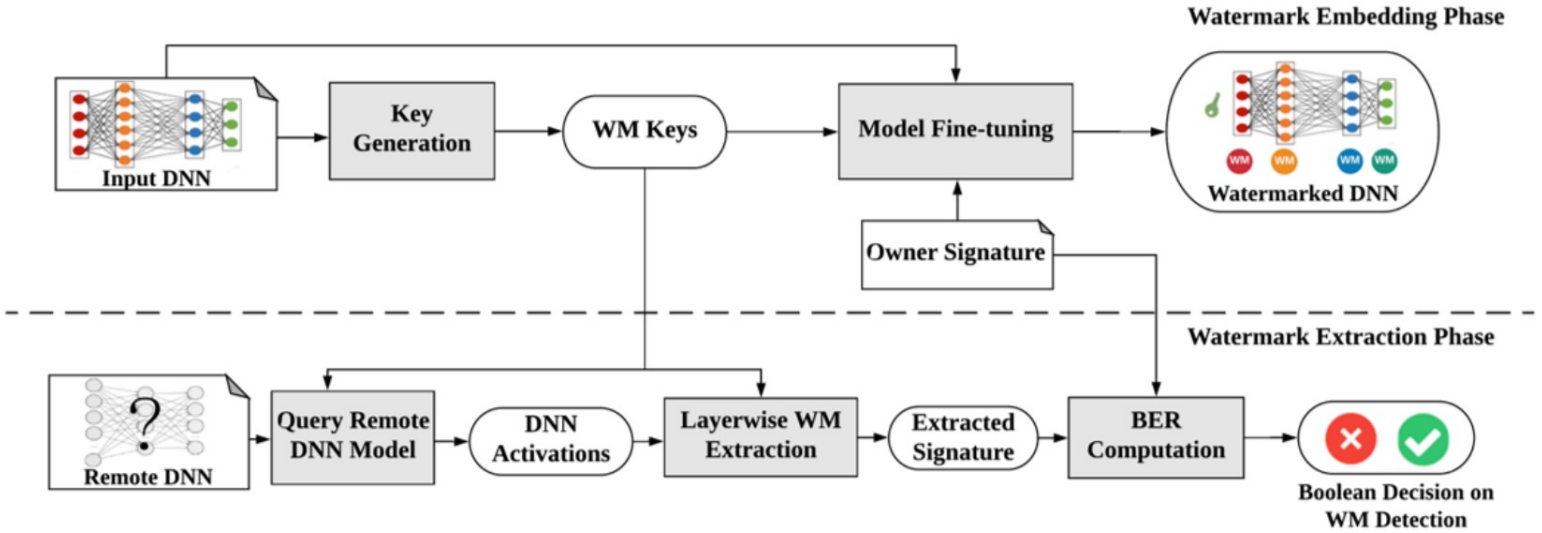
嵌入矩阵： X （提前固定好）

嵌入比特： $\boldsymbol{b}$=[1,0,0,1…,0,1,1] (512bit)

Uchida, Yusuke, et al. Embedding watermarks into deep neural networks. ACM ICMR, 2017

模型水印技术 – 白盒水印

激活空间水印：DeepSigns



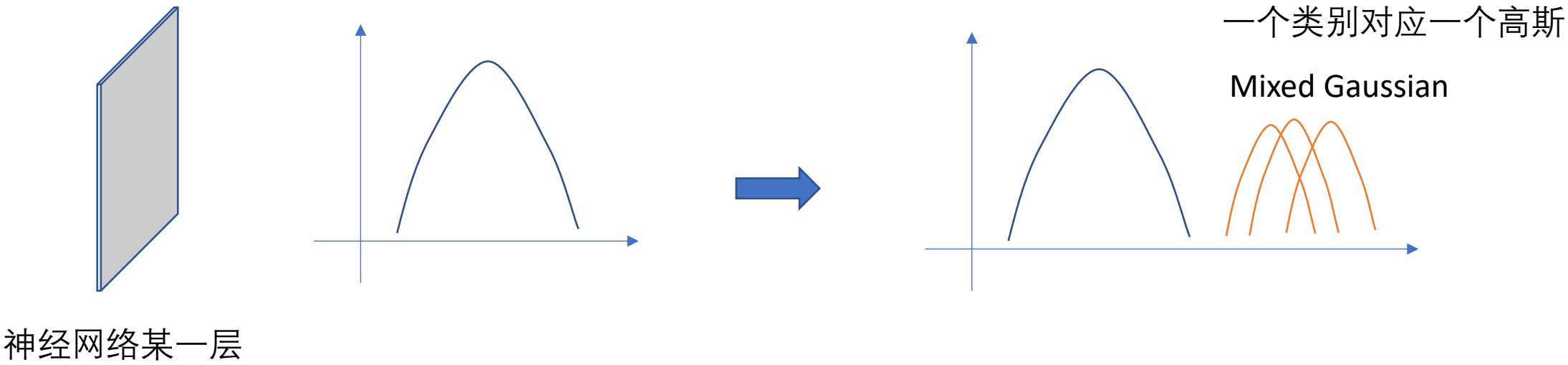
- 同时依赖数据和模型
- 对模型性能影响小
- 自动寻找low激活区域进行嵌入
- 对微调和重写鲁棒

□ 向激活图pdf（概率密度分布）中嵌入N比特

Darvish Rouhani et al. "Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks." ASPLOS. 2019

模型水印技术 – 白盒水印

激活空间水印：DeepSigns

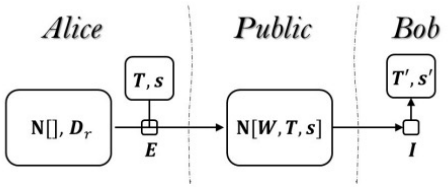


- 向激活图pdf（概率密度分布）中嵌入N比特
  - 选取多个key，每个都来自于一个高斯分布
  - 将key通过正则化训练入模型
  - 需要部分训练数据引导

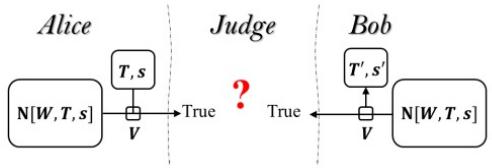
Darvish Rouhani et al. "Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks." ASPLOS. 2019

模型水印技术 – 白盒水印

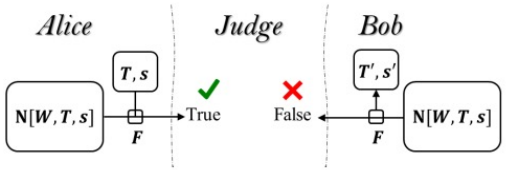
解决混淆攻击问题：DeepIPR、Passport Aware Normalization



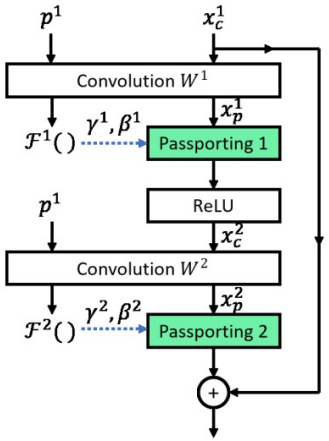
(a) Application Scenario.

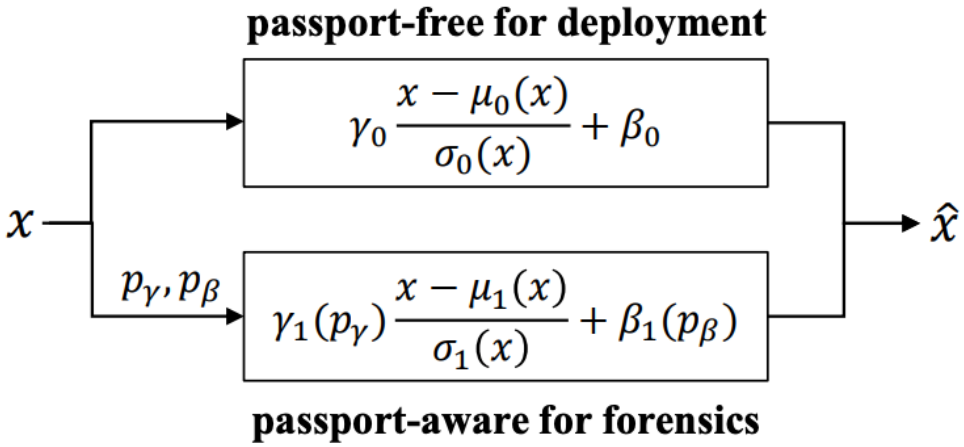


(b) Present Solution.



(c) Proposed Solution.



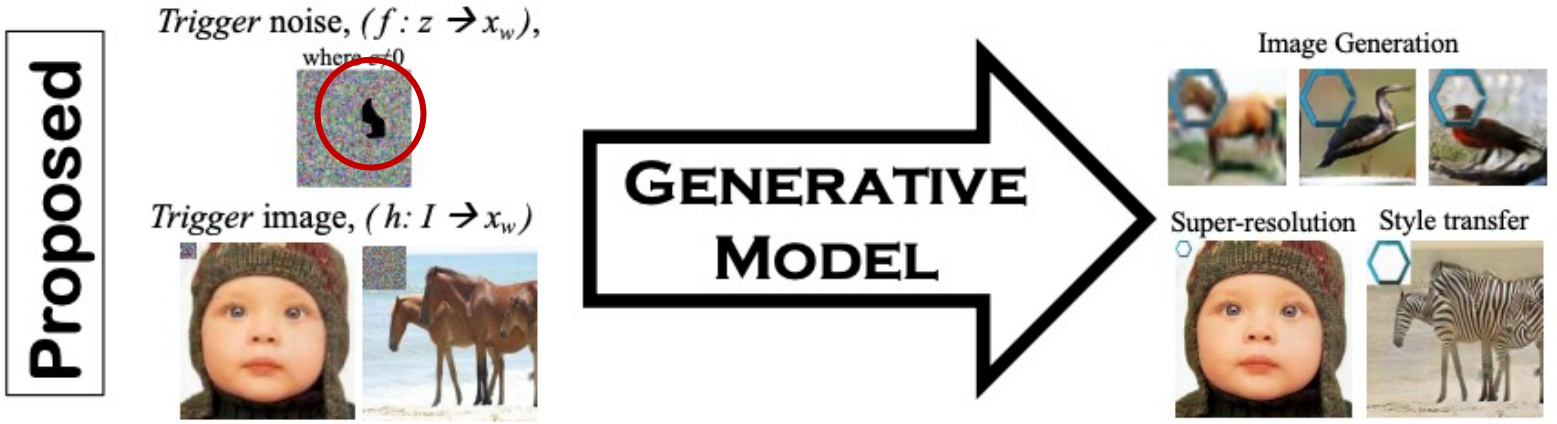


$$\hat{x} = \begin{cases} \gamma_0 \frac{x-\mu_0(x)}{\sigma_0(x)} + \beta_0 & \text{passport-free;} \\ \gamma_1(p_\gamma) \frac{x-\mu_1(x)}{\sigma_1(x)} + \beta_1(p_\beta) & \text{passport } p_\gamma, p_\beta. \end{cases}$$

Lixin Fan et al. “Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks”. NeurIPS, 2019.
Zhang, Jie, et al. “Passport-aware normalization for deep model protection.” NeurIPS 2020. Fan, Lixin, et al. "DeepIPR: Deep neural network ownership verification with passports." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2022): 6122-6139.

模型水印技术 – 白盒水印

生成模型水印：IPR-GAN



- 数据+参数（黑白盒兼顾）
- 参数部分只关注归一化层
- 白盒部分使用sign损失嵌入水印
- 对移除和混淆攻击都鲁棒

$$\hat{x} = \gamma \frac{x - \mu(x)}{\sigma(x)} + \beta, \quad \text{归一化层}$$

$$\mathcal{L}_s(\boldsymbol{\gamma}, \boldsymbol{B}) = \sum_{i=1}^{\tilde{}} \max(\gamma_0 - \gamma_i b_i, 0)$$

$$\boldsymbol{B} = \{b_1, \cdots, b_C \mid b \in \{-1, 1\}\}$$

Ong, Ding Sheng, et al. “Protecting intellectual property of generative adversarial networks from ambiguity attacks.” CVPR, 2021

模型水印技术 – 白盒水印方法总结

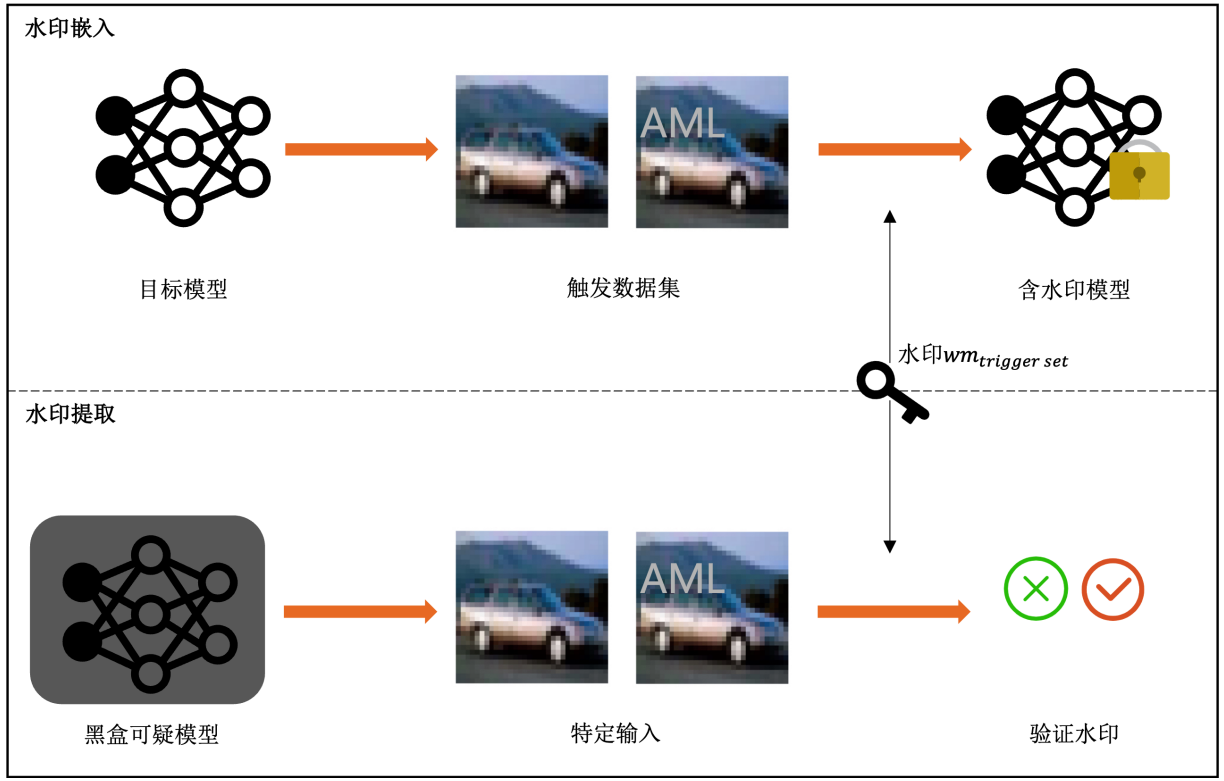
总结现有9种模型水印方法

水印算法	目标模型	训练集	模型性能	误码率
Uchida	WRN	CIFAR-10	91.55%	0
RIGA	Inception-V3	CelebA	95.90%	0
IPR-GAN	DCGAN	CUB200	54.33 (FID)	0
Greedy Residual	ResNet18	Caltech256	55.05%	0
Lottery	ResNet18	CIFAR-10	66.40%	0
DeepSigns	WRN	CIFAR-10	89.94%	0
IPR-IC	ResNet50+LSTM	COCO	72.06% (BLEU)	0
DeepIPR	ResNet18	CIFAR-100	67.94%	0
Passport-Aware	ResNet18	CIFAR-100	74.78%	0

Table credit to Yifan Yan.

模型水印技术 – 黑盒水印

基于后门的模型水印



- 黑盒：水印的验证不需要模型参数
- 数据引导，需要确保模型可以完全记住水印数据
- 是一种后门攻击

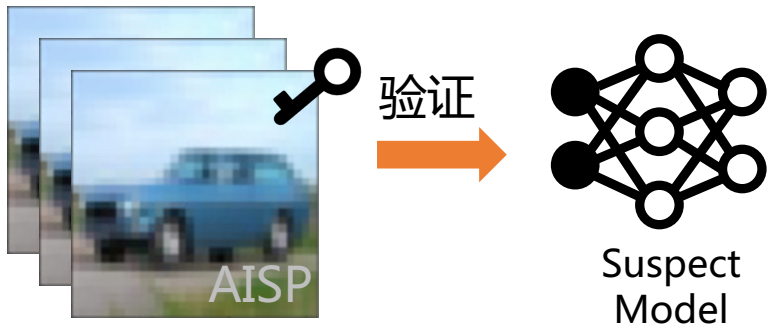
Zhang, Jialong, et al. Protecting intellectual property of deep neural networks with watermarking. ACM Asia CCS, 2018

模型水印技术 – 黑盒水印

基于后门的模型水印

Step1. 水印嵌入

Step2. 水印提取

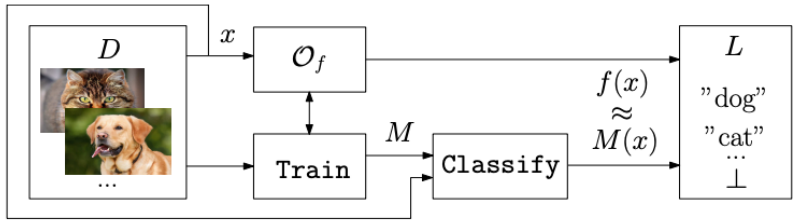


让模型学习”水印-类别”特定映射，AISP图案将作为模型所有权验证的钥匙。

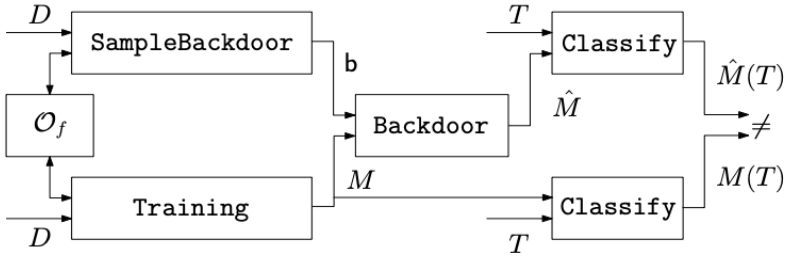
利用钥匙进行所有权验证，计算输出'airplane'的比例TSA (Trigger Set Accuracy)

模型水印技术 – 黑盒水印

基于后门的模型水印



正常学习



后门学习

触发图像示例

触发图像=key

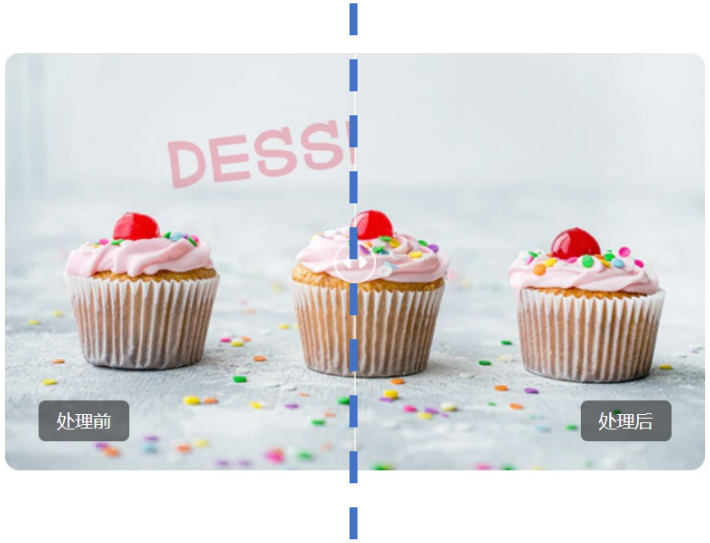
Adi, et al. “Turning your weakness into a strength: Watermarking deep neural networks by backdooring.” USENIX Security, 2018.

模型水印的优缺点

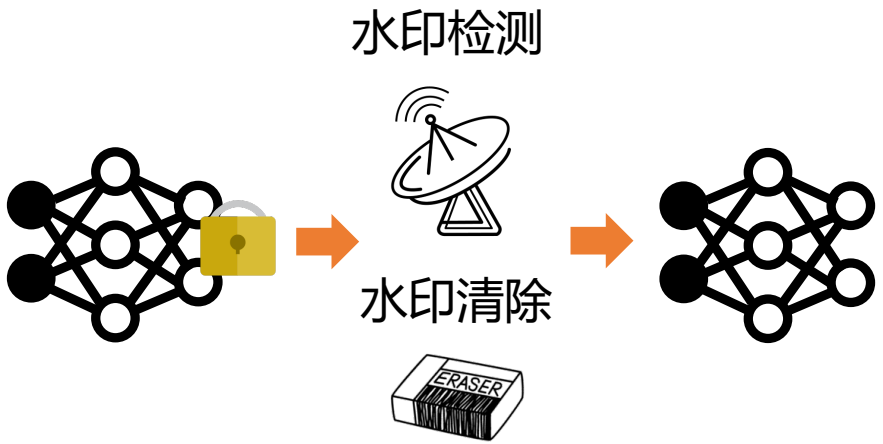
✔ 能够将身份（例如签名和徽标）嵌入模型中，提供准确的所有权验证

✘ 需要介入训练过程，会带来新的安全隐患，且可能会影响模型性能

✘ 水印提取面临各种挑战（水印损坏、被刻意擦除等）



图像水印清除



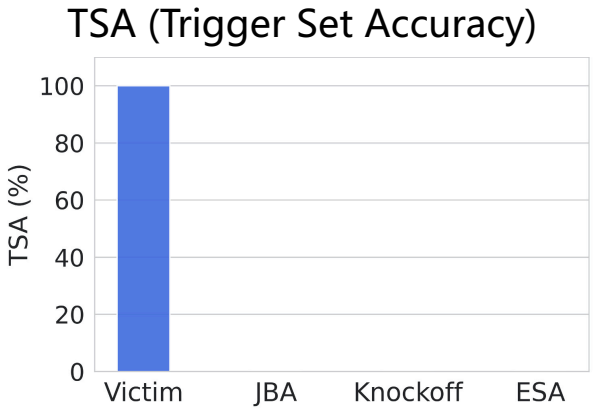
模型水印清除

水印的天敌：模型窃取

水印技术对模型微调和迁移防御效果较好，但是难以应对模型窃取

嵌入的水印信息在窃取过程中会被留在原模型中

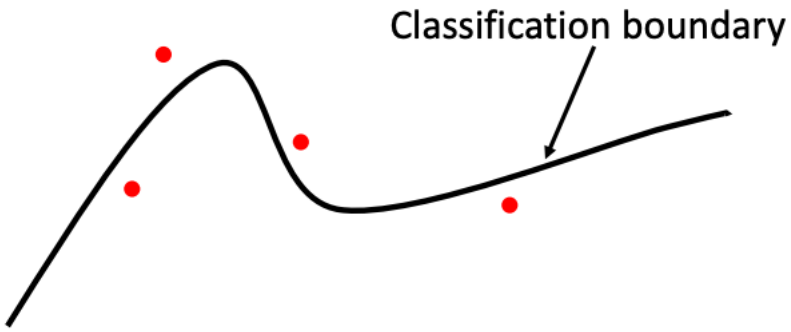
功能性窃取并不会窃取水印



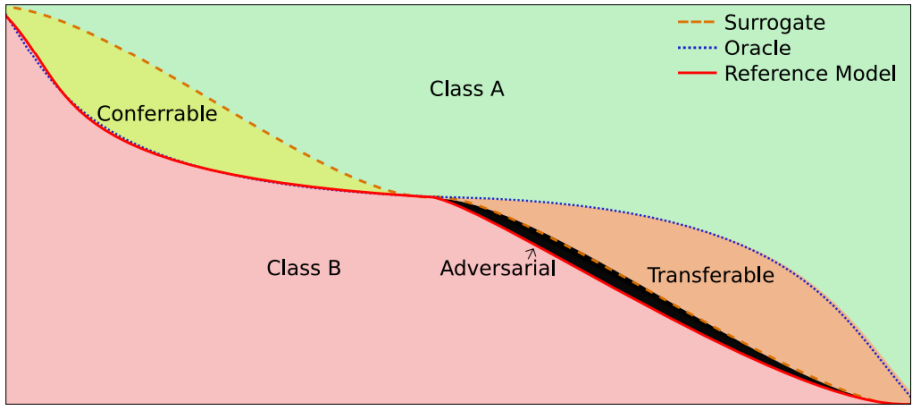
水印信息在模型窃取中完全丢失

模型窃取是水印的天敌

模型指纹



IPGuard：使用边界上的数据点去生成指纹



Conferrable Ensemble Method (CEM)

使用**可授予**对抗样本来生成边界指纹

Cao et al. IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary, Asia CCS, 2021
Lukas, Zhang and Kerschbaum. Deep Neural Network Fingerprinting by Conferrable Adversarial Examples, ICLR, 2021

模型指纹

Method	Type	Non-invasive	Evaluated Settings		Evaluated Attacks		
			Black-box	*White-box*	*Finetuning*	*Pruning*	*Extraction*
Uchida et al. [40]	*Watermarking*	✗	✗	✓	✓	✓	✗
Merrer et al. [23]	*Watermarking*	✗	✓	✗	✓	✓	✗
Adi et al. [1]	*Watermarking*	✗	✓	✗	✓	✗	✗
Zhang et al. [47]	*Watermarking*	✗	✓	✗	✓	✓	✗
Darvish et al. [9]	*Watermarking*	✗	✓	✓	✓	✓	✗
Jia et al. [20]	*Watermarking*	✗	✓	✗	✓	✓	✓
Cao et al. [2]	*Fingerprinting*	✓	✓	✗	✓	✓	✗
Lukas et al. [27]	*Fingerprinting*	✓	✓	✗	✓	✓	✓
DeepJudge (Ours)	*Testing*	✓	✓	✓	✓	✓	✓

DeepJudge

动机：衍生模型 (copy)一定会在很多方面跟源模型相似

思想：作为第三方视角，测试希望能更加**全面**的衡量可疑模型和源模型的相似度，将此作为证据链进行最终模型所有权的判断

以测试的方式：1）精心构建一组测试用例；2）测试并量化两个模型在测试用例上的行为相似度

哪些线索有价值：如何定义测试指标来衡量模型之间的相似性？

怎么准确找线索：如何有效地生成测试用例来放大相似度？

框架总览

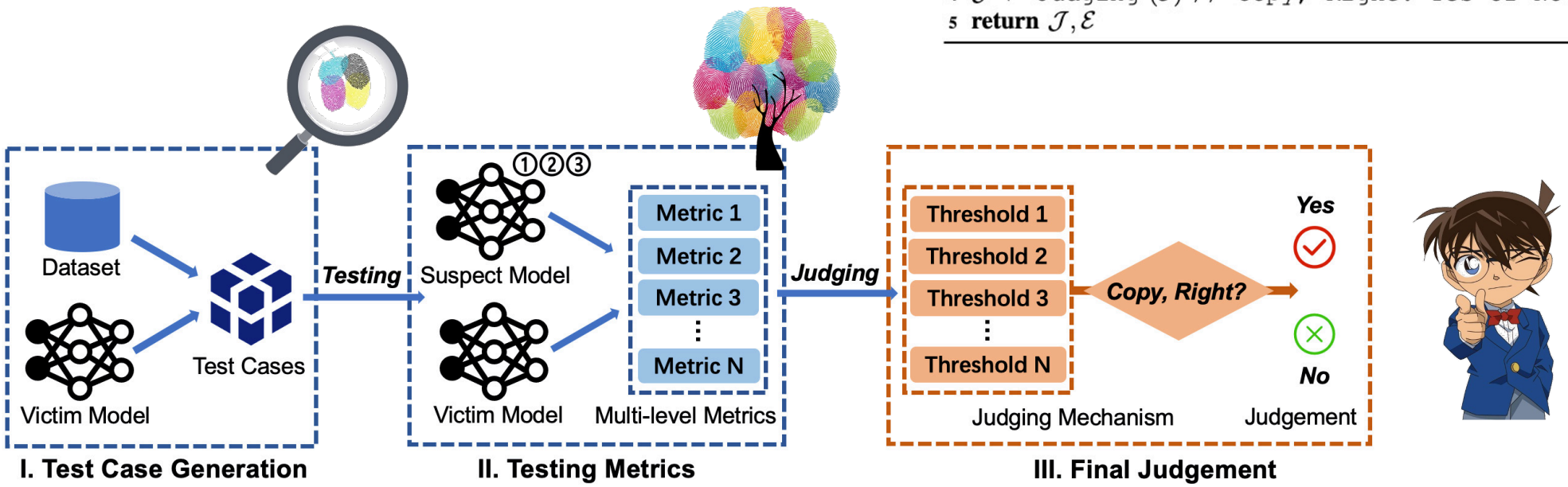
DeepJudge 由三部分组成:

- 一组预先生成的测试用例
- 一组用于测试的多层次距离指标
- 基于阈值/投票的判断机制

Algorithm 1: DEEPJUDGE$(\mathcal{O}, \mathcal{S}, \mathcal{D})$

Input: owner model $\mathcal{O}$, suspect model $\mathcal{S}$, data set $\mathcal{D}$
Output: judgement $\mathcal{J}$, evidence $\mathcal{E}$

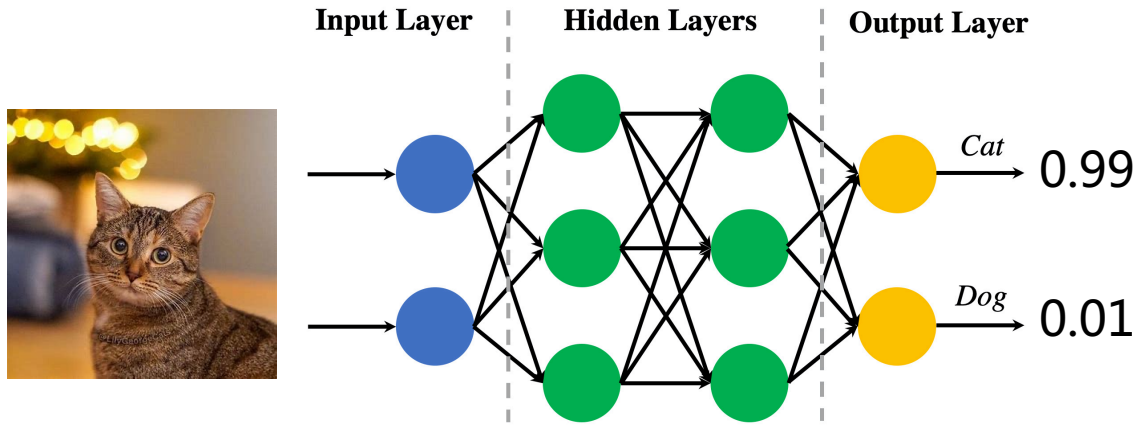
```
// Test case generation (Section IV-C)
Seeds ← SelectSeeds (O, D)
T ← GenerateTestCases (O, Seeds)
// Testing metrics (Section IV-B)
E ← ComputeMetrics (O, S, T)
// Final judgement (Section IV-D)
J ← Judging (E) // Copy, Right? Yes or No.
return J, E
```



放大线索的测试用例　　线索收集　　最终决策机制

神经网络模型组成

- DNN分类模型 $f: X \to Y$； $x \in X$ ，$y \in Y = \{1, 2, 3, \cdots, C\}$
- 模型有 L 层: $f = \{f^1, f^2, \cdots, f^{L-1}, f^L\}$, 其中$f^l$=$\{n_{l,1}, n_{l,2}, \cdots, n_{l,N_l}\}$（神经元集）
- 每一层的输出向量： f^l $(2 \le l \le L)$: $f^l(x) =< \phi_{l,1}(x), \phi_{l,2}(x), \cdots, \phi_{l,N_l}(x) >$
- 最后一层的输出向量： $f^L, f^L(x) =< \phi_{L,1}(x), \phi_{L,2}(x), \cdots, \phi_{L,C}(x) >$, $\sum_{i=1}^{C} \phi_{l,1}(x) = 1$
- $f^L(x)$ 是概率向量, $f(x) = \text{argmax}\, f^L(x)$ 是最终的类别预测



Cat (0.48)

Dog(0.52)

不同层次测试指标

DeepJudge 可在两种设置下进行：

- 白盒：可以完全访问可疑模型 s 的内部（即中间层输出）和最终概率向量。
- 黑盒：只能查询可疑模型 s 以获得概率向量或预测标签。

Level	Metric	Defense Setting
Property-level	Robustness Distance (RobD)	Black-box
Neuron-level	Neuron Output Distance (NOD)	White-box
	Neuron Activation Distance (NAD)	White-box
Layer-level	Layer Outputs Distance (LOD)	White-box
	Layer Activation Distance (LAD)	White-box
	Jensen-Shanon Distance (JSD)	Black-box

鲁棒性

模型属性可用于表征两个模型之间的相似性，我们定义了**鲁棒性距离**（RobD）来衡量两个模型之间的对抗鲁棒性差异。

动机：模型的鲁棒性与模型通过其独特的优化过程学习到的决策边界密切相关，可以被视为模型的一种“指纹”

Def 1. Rob: Given a set of test cases, we can obtain its adversarial version $\mathcal{T} = \{x_1', x_2', \cdots\}$, where x_i' denotes the adversarial example of x_i'. The robustness property of model f can then be defined as its accuracy on $\mathcal{T}$:

$$Rob(f, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} (f(x_i') = y_i)$$

Def 2. RobD (Robustness Distance): Let $\hat{f}$ be the suspect model, we define the robustness distance between f and $\hat{f}$ by the absolute difference between the two models' robustness:

$$RobD(f, \hat{f}, \mathcal{T}) = \left| Rob(\hat{f}, \mathcal{T}) - Rob(f, \mathcal{T}) \right|$$

神经元输出和激活状态

我们使用神经元的输出状态来捕捉两个模型之间的差异，并定义了两个神经元级别的指标：**神经输出距离** NOD 和**神经激活距离** NAD。

动机: 模型中每个神经元输出遵循自己的统计分布，不同模型的神经元输出应该有所不同

Def 3. NOD (Neuron Output Distance): Let $\phi_{l,i}$ and $\hat{\phi}_{l,i}$ be the neuron output function of the victim model and the suspect model, NOD measures the average neuron output difference on $\mathcal{T} = \{x_1, x_2, \cdots\}$:

$$NOD\left(\phi_{l,i}, \hat{\phi}_{l,i}, \mathcal{T}\right) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} |\phi_{l,i}(x) - \hat{\phi}_{l,i}(x)|$$

Def 4. NAD (Neuron Activation Distance): NAD measures the difference in activation status ('activated' vs. 'not activated') between the neurons of two models:

$$NAD\left(\phi_{l,i}, \hat{\phi}_{l,i}, \mathcal{T}\right) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} |\mathbb{S}\left(\phi_{l,i}(x)\right) - \mathbb{S}\left(\hat{\phi}_{l,i}(x)\right)|$$

网络层激活分布

神经层级别的指标提供了两个模型之间**中间层输出差异**的完整视图

Def 5. LOD (Layer Output Distance)

$$LOD(f^l, \hat{f}^l, \mathcal{T}) = \frac{1}{|\mathcal{T}|}\sum_{x\in\mathcal{T}}\left|f^l(x) - \hat{f}^l(x)\right|_p$$

Def 6. LAD (Layer Activation Distance)

$$LAD(f^l, \hat{f}^l, \mathcal{T}) = \frac{1}{|N_l|}\sum_{i=1}^{|N_l|} NAD(\phi_{l,i}, \hat{\phi}_{l,i}, \mathcal{T})$$

Def 7. JSD (Jensen-Shanon Distance): Let f^L and $\hat{f}^L$ denote the output functions (output layer) of the victim model and the suspect model, *JSD* measures the similarly of two probability distributions:

$$JSD(f^L, \hat{f}^L, \mathcal{T}) = \frac{1}{|\mathcal{T}|}\sum_{x\in\mathcal{T}}\frac{1}{2}KL(f^L(x), Q) + \frac{1}{2}KL(\hat{f}^L(x), Q),$$

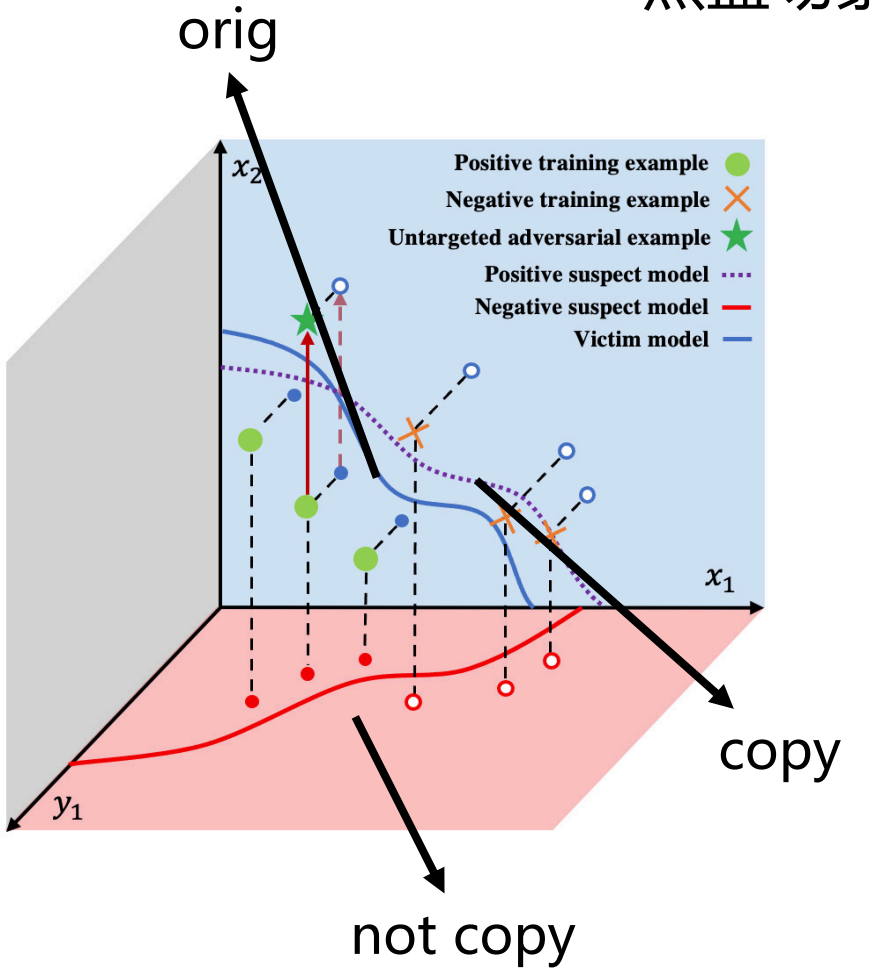
$$Q = \frac{1}{2}(f^L(x) + \hat{f}^L(x))$$

测试用例生成

- 为了充分利用好提出的测试指标，我们需要**放大窃取模型与源模型之间的相似性**，同时最小化非窃取模型 (独立训练) 和源模型的相似性。
- 测试用例应根据不同设置（即黑盒与白盒）分别生成。

黑盒测试用例：对抗样本

黑盒场景：使用对抗样本作为测试集 $\mathcal{T}$

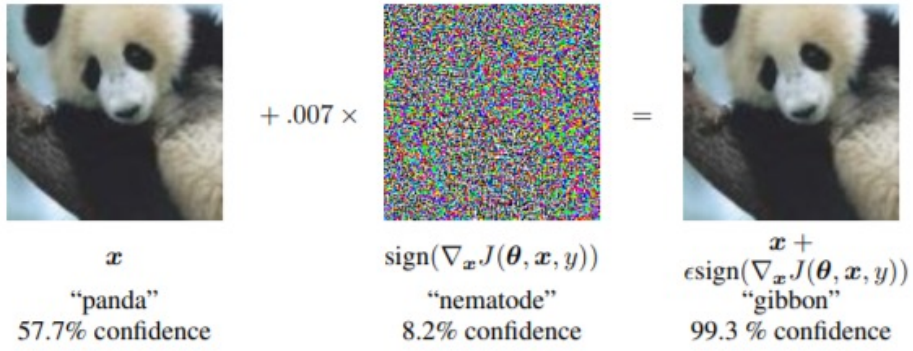


微调和剪枝的模型都是从源模型得到的衍生模型，应该与源模型保持类似的决策边界

虽然模型窃取是从头开始训练新的模型，但会逐渐模仿源者模型的决策行为（边界），模型提取得越好，副本与源模型越相似，并且越容易被测出来

对抗样本

在正常样本上加上微小的扰动，使模型的决策发生变化，但输入的实际意义并未发生变化。

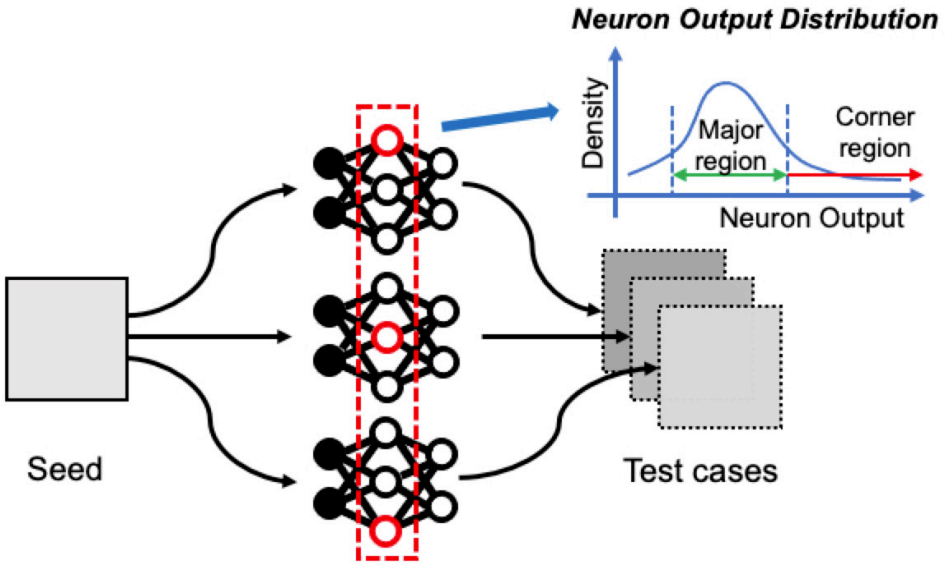


(a) Input 1 (b) Input 2 (darker version of 1)

看上过去一样，吃着口味不同~

白盒测试用例：合成样本

白盒：给定一个种子输入和一个指定层，我们为每个神经元生成一个测试用例。其中，**神经元激活的极端情况**是我们感兴趣的。



Algorithm 2: GenerateTestCases($\mathcal{O}$, $Seeds$)

Input: owner model $\mathcal{O}$, a set of seed inputs $Seeds$
Output: test suite T

Initialize test suite $T \leftarrow \emptyset$
for *each neuron* $n_{l,i}$ **do**
  Sample a seed input $\boldsymbol{x} \leftarrow Seeds.choice()$
  $\boldsymbol{x}' \leftarrow copy(\boldsymbol{x})$
  for $iter = 1$ **to** $iters$ **do**
    Calculate gradients $grads \leftarrow \frac{\nabla \phi_{l,i}(\boldsymbol{x}')}{\nabla \boldsymbol{x}'}$
    Perturb input $\boldsymbol{x}' \leftarrow \boldsymbol{x}' + lr \cdot grads$
    if $\phi_{l,i}(\boldsymbol{x}') > threshold\ k$ **then**
      Add new test case $T \leftarrow T \cup \{\boldsymbol{x}'\}$
      break
    end
  end
end
return T

最终决策机制

DeepJudge 的判断分两步：**阈值**和**投票**。

投票：根据每个测试指标给嫌疑模型投票，如果它与源模型的距离低于该指标的阈值，则给它一个肯定票。

$$\tau_\lambda = \alpha_\lambda \cdot LB_\lambda$$

$$p_{copy}(\mathcal{O}, \mathcal{S}, \mathcal{T}) = \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} \mathbb{I}(\lambda(\mathcal{O}, \mathcal{S}, \mathcal{T}) \leq \tau_\lambda)$$

测量指标的度量值（模型距离）越低，可疑模型就越有可能是源模型的副本。如果可疑模型获得更多肯定票，则将其识别为肯定窃取模型。

Copy, Right?

Yes, it is a copy!

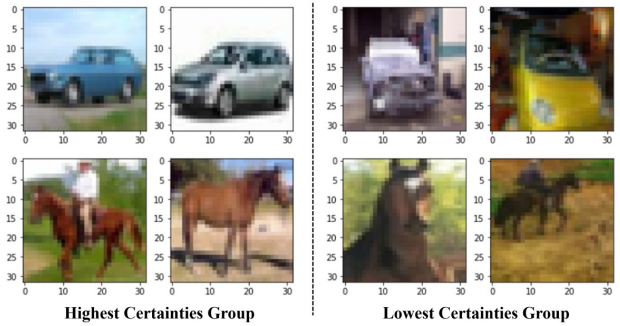
实验设置

- 数据集和要保护的模型

Dataset	Type	Model	#Params	Accuracy
MNIST	Image	LeNet-5	107.8 K	98.5%
CIFAR-10	Image	ResNet-20	274.4 K	84.8%
ImageNet	Image	VGG-16	33.65 M	74.4%
SpeechCommands	Audio	LSTM(128)	132.4 K	94.9%

#Params: number of parameters

测试种子



- 嫌疑模型

Positive嫌疑	微调最后一层（FT-LL）	微调所有层（FT-AL）	重置最后一层后重训练（RT-AL）	模型剪枝P-r%（20%-60%）	迁移学习	模型窃取
Negative嫌疑	独立训练，不同随机初始化（Neg-1）	训练在其余50%数据上（Neg-2）				

我们总共实验了：11种模型攻击方法，3 种基线方法, 4个数据集，超过300个不同的深度学习模型

面对：模型微调 & 模型剪枝

□ DeepJudge在黑盒设置下的有效性

Model Type		MNIST				CIFAR-10			
		ACC	*RobD*	*JSD*	Copy?	ACC	*RobD*	*JSD*	Copy?
Victim Model		98.5%	–	–	–	84.8%	–	–	–
Positive Suspect Models	FT-LL	98.8±0.0%	0.019±0.003	0.016±0.002	**Yes** (2/2)	82.1±0.1%	0.000±0.000	0.002±0.001	**Yes** (2/2)
	FT-AL	98.7±0.1%	0.045±0.016	0.033±0.010	**Yes** (2/2)	79.9±1.4%	0.192±0.028	0.162±0.014	**Yes** (2/2)
	RT-AL	98.4±0.2%	0.298±0.039	0.151±0.017	**Yes** (2/2)	79.4±0.8%	0.237±0.055	0.197±0.027	**Yes** (2/2)
	P-20%	98.7±0.1%	0.058±0.014	0.035±0.009	**Yes** (2/2)	81.7±0.2%	0.155±0.032	0.128±0.018	**Yes** (2/2)
	P-60%	98.6±0.1%	0.172±0.024	0.097±0.010	**Yes** (2/2)	81.1±0.6%	0.318±0.036	0.233±0.019	**Yes** (2/2)
Negative Suspect Models	Neg-1	98.4±0.3%	0.968±0.014	0.614±0.016	**No** (0/2)	84.2±0.6%	0.920±0.021	0.603±0.016	**No** (0/2)
	Neg-2	98.3±0.2%	0.949±0.029	0.600±0.020	**No** (0/2)	84.9±0.5%	0.926±0.030	0.615±0.021	**No** (0/2)
	τ_λ	–	**0.852**	**0.538**	–	–	**0.816**	**0.537**	–

Model Type		ImageNet				SpeechCommands			
		ACC	*RobD*	*JSD*	Copy?	ACC	*RobD*	*JSD*	Copy?
Victim model		74.4%	–	–	–	94.9%	–	–	–
Positive Suspect Models	FT-LL	73.2±0.4%	0.034±0.007	0.009±0.003	**Yes** (2/2)	95.2±0.1%	0.104±0.007	0.036±0.006	**Yes** (2/2)
	FT-AL	70.8±0.9%	0.073±0.011	0.043±0.011	**Yes** (2/2)	95.8±0.3%	0.326±0.024	0.155±0.014	**Yes** (2/2)
	RT-AL	53.3±0.8%	0.192±0.008	0.251±0.015	**Yes** (2/2)	94.3±0.3%	0.445±0.019	0.231±0.016	**Yes** (2/2)
	P-20%	69.7±1.1%	0.106±0.010	0.064±0.003	**Yes** (2/2)	95.4±0.2%	0.310±0.026	0.152±0.013	**Yes** (2/2)
	P-60%	68.8±1.0%	0.161±0.017	0.091±0.004	**Yes** (2/2)	95.0±0.5%	0.437±0.030	0.215±0.013	**Yes** (2/2)
Negative Suspect Models	Neg-1	74.2±0.3%	0.737±0.007	0.395±0.006	**No** (0/2)	94.9±0.7%	0.819±0.025	0.456±0.014	**No** (0/2)
	Neg-2	73.9±0.5%	0.760±0.010	0.429±0.004	**No** (0/2)	94.5±0.8%	0.832±0.024	0.472±0.012	**No** (0/2)
	τ_λ	–	**0.659**	**0.356**	–	–	**0.727**	**0.405**	–

红色：copy
绿色：not copy

面对：模型微调 & 模型剪枝

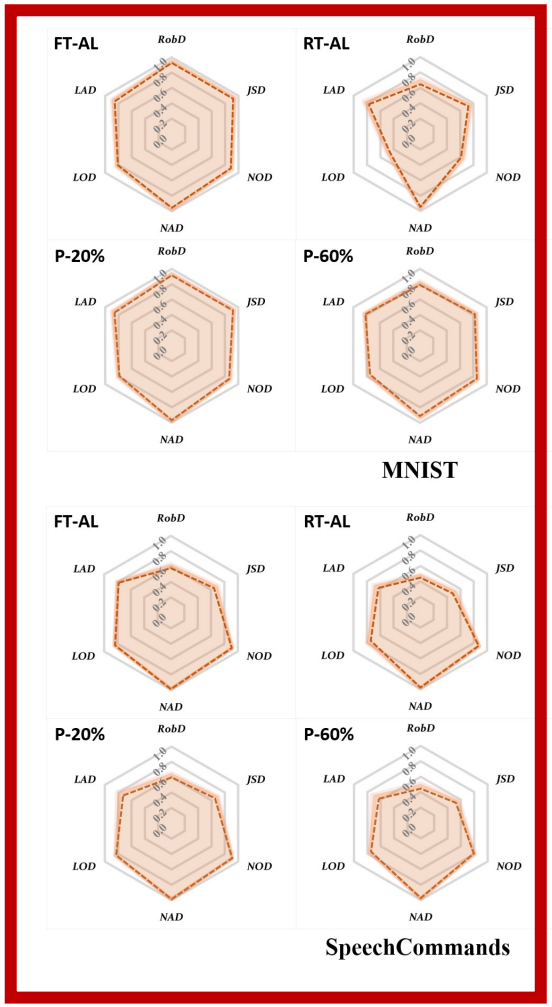
□ DeepJudge在白盒设置下的有效性

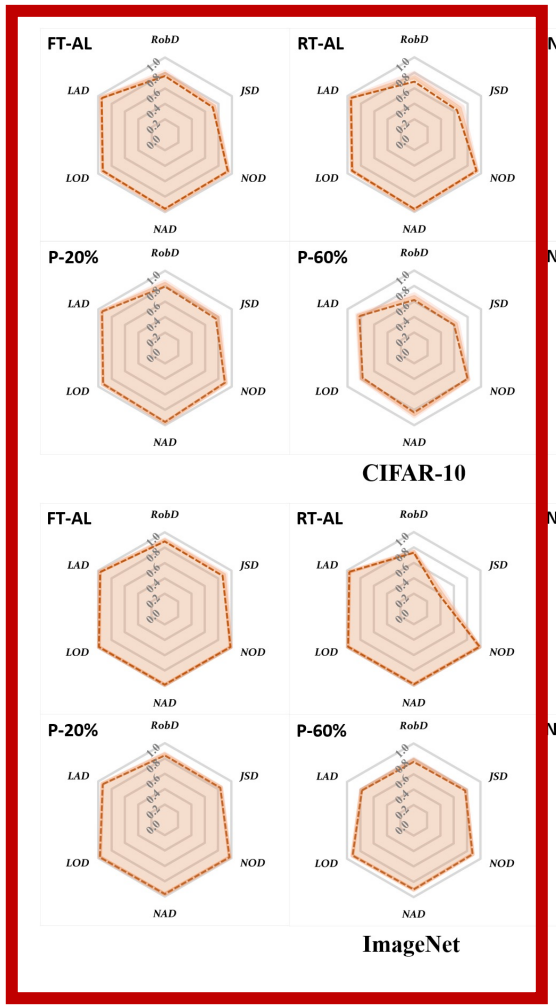
Model Type		MNIST					CIFAR-10				
		NOD	*NAD*	*LOD*	*LAD*	Copy?	*NOD*	*NAD*	*LOD*	*LAD*	Copy?
Positive Suspect Models	FT-LL	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	**Yes** (4/4)	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	**Yes** (4/4)
	FT-AL	0.08±0.01	0.23±0.21	0.32±0.03	0.82±0.16	**Yes** (4/4)	0.15±0.02	0.30±0.12	0.74±0.07	0.21±0.04	**Yes** (4/4)
	RT-AL	0.31±0.02	0.37±0.20	0.97±0.04	1.27±0.29	**Yes** (4/4)	0.18±0.02	0.26±0.10	0.78±0.03	0.22±0.02	**Yes** (4/4)
	P-20%	0.10±0.01	0.16±0.12	0.36±0.03	0.79±0.15	**Yes** (4/4)	0.28±0.03	0.32±0.09	0.77±0.06	0.24±0.02	**Yes** (4/4)
	P-60%	0.11±0.01	0.82±0.26	0.43±0.03	1.16±0.08	**Yes** (4/4)	0.62±0.03	1.65±0.34	2.80±0.21	0.93±0.10	**Yes** (4/4)
Negative Suspect Models	Neg-1	0.77±0.07	11.46±1.14	1.73±0.06	6.42±0.84	**No** (0/4)	3.09±0.30	10.94±1.74	11.85±1.01	5.41±0.67	**No** (0/4)
	Neg-2	0.79±0.08	12.28±1.50	1.78±0.13	6.37±0.47	**No** (0/4)	3.21±0.18	11.09±0.71	12.60±1.33	5.37±0.72	**No** (0/4)
	τ_λ	**0.45**	**6.74**	**1.03**	**3.65**	–	**1.79**	**6.14**	**6.89**	**3.01**	–

Model Type		ImageNet					SpeechCommands				
		NOD	*NAD*	*LOD*	*LAD*	Copy?	*NOD*	*NAD*	*LOD*	*LAD*	Copy?
Positive Suspect Models	FT-LL	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	**Yes** (4/4)	0.000±0.000	0.00±0.00	0.00±0.00	0.000±0.00	**Yes** (4/4)
	FT-AL	0.02±0.01	0.18±0.09	0.16±0.05	0.58±0.13	**Yes** (4/4)	0.037±0.003	0.05±0.02	0.42±0.02	12.82±1.00	**Yes** (4/4)
	RT-AL	0.03±0.00	0.30±0.07	0.25±0.03	0.78±0.05	**Yes** (4/4)	0.055±0.003	0.25±0.31	0.64±0.08	21.64±2.47	**Yes** (4/4)
	P-20%	0.11±0.01	0.83±0.06	0.76±0.01	1.67±0.22	**Yes** (4/4)	0.038±0.002	0.03±0.02	0.44±0.02	14.57±3.12	**Yes** (4/4)
	P-60%	0.77±0.01	3.09±0.12	3.41±0.03	6.63±0.23	**Yes** (4/4)	0.094±0.004	0.45±0.32	0.67±0.04	20.58±3.44	**Yes** (4/4)
Negative Suspect Models	Neg-1	6.55±0.78	32.18±2.97	35.03±3.13	30.32±1.91	**No** (0/4)	0.488±0.013	39.61±9.74	2.82±0.08	64.32±2.42	**No** (0/4)
	Neg-2	6.25±0.39	30.04±2.44	44.21±3.11	29.58±0.86	**No** (0/4)	0.480±0.012	34.84±6.07	2.79±0.09	62.69±1.75	**No** (0/4)
	τ_λ	**3.48**	**17.17**	**20.74**	**17.20**	–	**0.286**	**19.77**	**1.66**	**37.48**	–

红色：copy
绿色：not copy

结合可视化

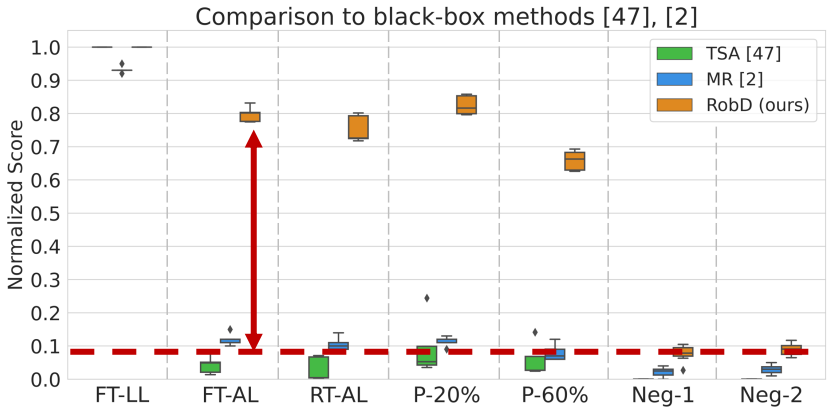


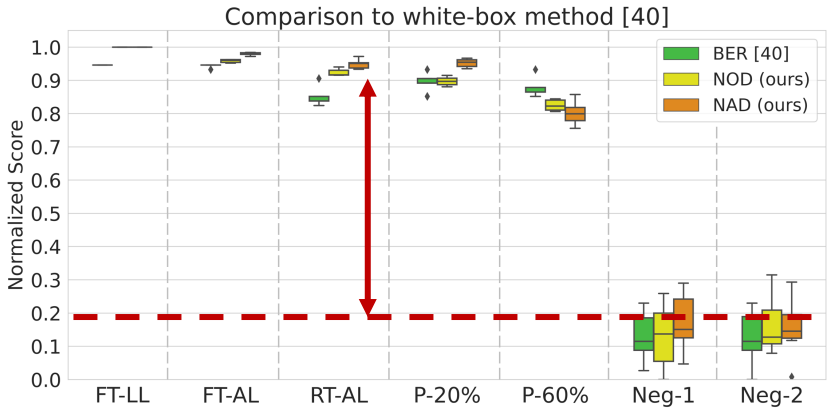


雷达图面积
=copy置信度

DeepJudge vs 模型水印

- 黑盒设置下: 窃取 (copy) 模型和非窃取 (not copy) 模型在DeepJudge下的差距更加显著
- 白盒设置下: 和白盒水印有相似的表现，其中NAD指标在 5 种攻击策略上占据优势





[47]: DNNWatermarking
[2]: IPGuard

[40]: Uchida et al. EmdeddingWatermarking

TSA： Trigger Set Accuracy　　MR： Matching Rate　　BER： Bit Error Rate

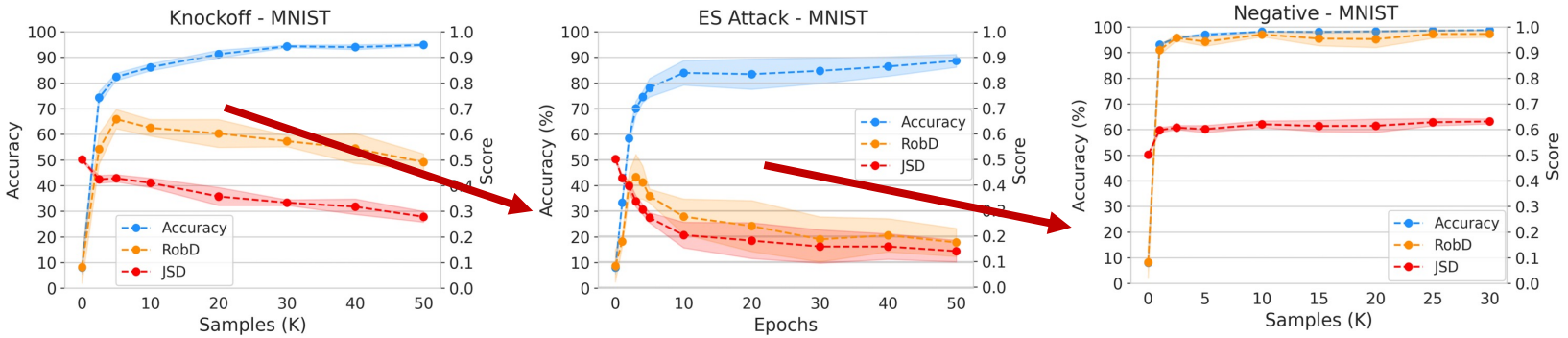
Zhang, et al. Protecting intellectual property of deep neural networks with watermarking. Asia CCS, 2018
Cao et al. IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary, Asia CCS, 2021
Uchida et al. Embedding watermarks into deep neural networks, ICMR, 2017

模型窃取

Model Type		MNIST				CIFAR-10				SpeechCommands			
		ACC	*RobD*	*JSD*	Copy?	ACC	*RobD*	*JSD*	Copy?	ACC	*RobD*	*JSD*	Copy?
Positive	JBA	83.6±1.7%	0.866±0.034	0.596±0.006	**No** (0/2)	40.3±1.5%	0.497±0.044	0.541±0.015	**No** (1/2)	40.1±1.7%	0.381±0.030	0.470±0.011	**No** (1/2)
Suspect Models	Knock	94.8±0.6%	0.491±0.032	0.273±0.021	**Yes** (2/2)	74.4±1.0%	0.715±0.018	0.436±0.019	**Yes** (2/2)	86.6±0.5%	0.618±0.012	0.303±0.007	**Yes** (2/2)
	ESA	88.7±2.5%	0.175±0.056	0.141±0.042	**Yes** (2/2)	67.1±1.9%	0.144±0.031	0.249±0.033	**Yes** (2/2)	×	×	×	–
Negative Suspect Models	Neg-1	98.4±0.3%	0.968±0.014	0.614±0.016	**No** (0/2)	84.2±0.6%	0.920±0.021	0.665±0.016	**No** (0/2)	94.9±0.7%	0.817±0.025	0.456±0.014	**No** (0/2)
	Neg-2	98.3±0.2%	0.949±0.029	0.600±0.020	**No** (0/2)	84.9±0.5%	0.926±0.030	0.615±0.021	**No** (0/2)	94.5±0.8%	0.832±0.024	0.472±0.012	**No** (0/2)
	τ_λ	–	**0.852**	**0.538**	–	–	**0.816**	**0.537**	–	–	**0.727**	**0.405**	–

红色：copy
绿色：not copy

在模型窃取过程中，模型间RobD和JSD距离越来越小



逐渐变成你的模样~

- Yuan et al. ES attack: Model stealing against deep neural networks without data hurdles.
- Orekondy et al. Knockoff nets: Stealing functionality of black-box models, CVPR, 2019
- Papernot et al. Practical black-box attacks against machine learning, Aisa CCS, 2017

Adaptive Attacks

- **测试指标和测试用例均暴露的情况**
  - 对抗微调：将测试用例 $\mathcal{T}$ 混合到干净的数据中，并对窃取的模型进行微调

- **仅测试指标暴露的情况**
  - 对抗训练：提升模型鲁棒性，使决策边界更加平滑
  - 迁移学习：将当前任务迁移到另一个任务上

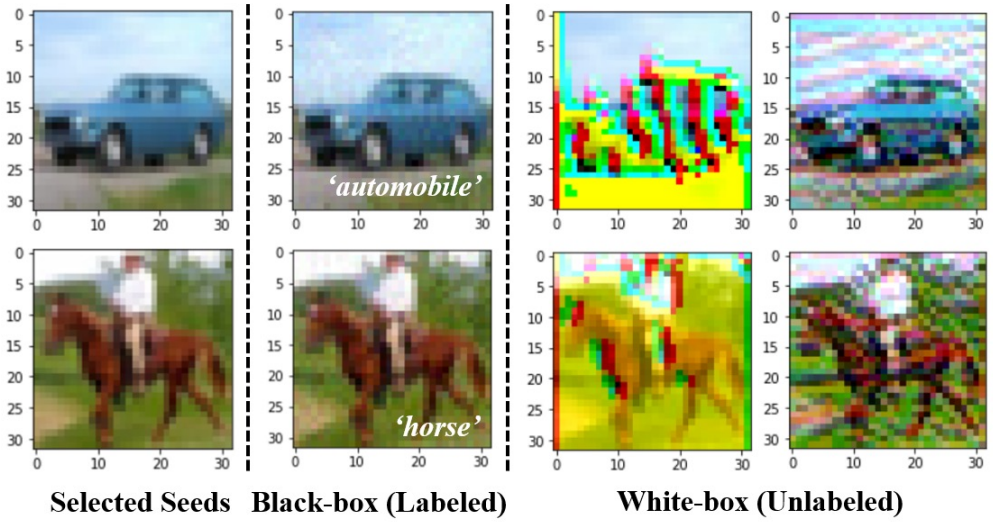
Adaptive Attacks

Model Type		Black-box Testing			White-box Testing				
		ACC	*RobD*	*JSD*	*NOD*	*NAD*	*LOD*	*LAD*	Copy?
Positive Suspect Models	Adapt-B	81.4±0.9%	0.985±0.011	0.665±0.007	0.38±0.04	0.44±0.15	1.12±0.06	0.40±0.05	**Yes** (4/6)
	Adapt-W	71.9±1.8%	0.519±0.048	0.372±0.025	3.11±0.12	1.94±0.12	11.62±0.54	1.89±0.33	**Yes** (4/6)
	Adv-Train	74.5±2.3%	0.939±0.087	0.637±0.036	0.68±0.11	0.79±0.17	1.89±0.14	0.75±0.08	**Yes** (4/6)
	VTL	93.3±1.7%	×	×	0.85±0.23	1.08±0.14	2.58±0.24	0.64±0.15	**Yes** (4/4)
Negative Suspect Models	Neg-1	84.2±0.6%	0.920±0.021	0.603±0.016	3.09±0.30	10.94±1.74	11.85±1.01	5.41±0.67	**No** (0/6)
	Neg-2	84.9±0.5%	0.926±0.030	0.615±0.021	3.21±0.18	11.09±0.71	12.60±1.33	5.37±0.72	**No** (0/6)
	τ_λ	–	**0.816**	**0.537**	**1.79**	**6.14**	**6.89**	**3.01**	–

红色：copy
绿色：not copy

- Adapt-B: 针对黑盒测试，攻击者知道测试方法和样本
- Adapt-W: 针对白盒测试，攻击者知道测试方法和样本
- Adv-Train: 攻击者只知道测试样本是对抗样本
- VTL: 迁移学习10类 CIFAR-10 到 5类 SVHN

适应性攻击可以躲避部分指标，但不能完全躲避



黑盒测试用例　　白盒测试用例

黑盒测试用例的不同生成方法

- 对抗样本生成方法的选择 （基于RobD结果）

10xsteps

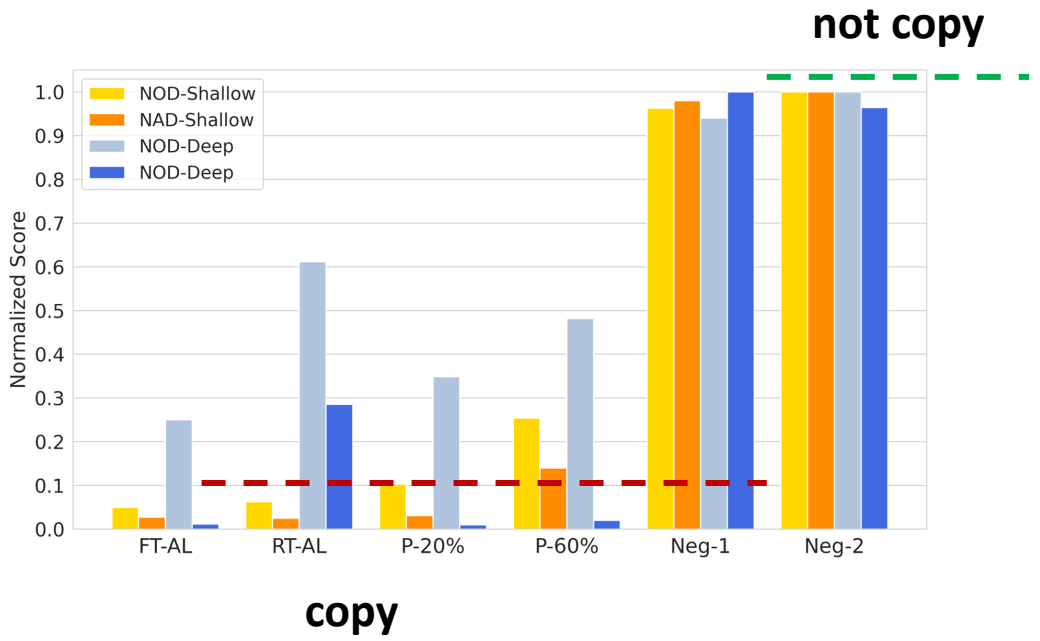
Model Type		FGSM	CW	PGD	$\text{PGD}_{\epsilon/3}$	$\text{PGD}_{3\epsilon}$	PGD_{10s}
Positive Suspect Models	FT-LL	0.024±0.004	0.0±0.0	0.0±0.0	0.034±0.002	0.0±0.0	0±0.0
	FT-AL	0.261±0.025	0.905±0.028	0.192±0.028	0.733±0.012	0.046±0.010	0.350±0.027
	RT-AL	0.267±0.025	0.917±0.024	0.237±0.055	0.748±0.046	0.073±0.022	0.400±0.046
	P-20%	0.252±0.030	0.882±0.038	0.155±0.032	0.702±0.023	0.045±0.020	0.299±0.049
	P-60%	0.293±0.027	0.940±0.013	0.318±0.036	0.792±0.023	0.123±0.022	0.502±0.031
Negative Suspect Models	Neg-1	0.662±0.058	0.999±0.002	0.920±0.021	0.989±0.007	0.573±0.093	0.958±0.013
	Neg-2	0.672±0.019	0.998±0.003	0.926±0.030	0.986±0.004	0.576±0.030	0.948±0.012
	τ_λ	**0.583**	**0.897**	**0.816**	**0.886**	**0.489**	**0.851**

需要能够最大化两类模型差距的对抗样本生成方法

白盒测试神经层的选择

- 测试神经层的选择

NOD：Neuron Output Distance
NAD: Neuron Activation Distance

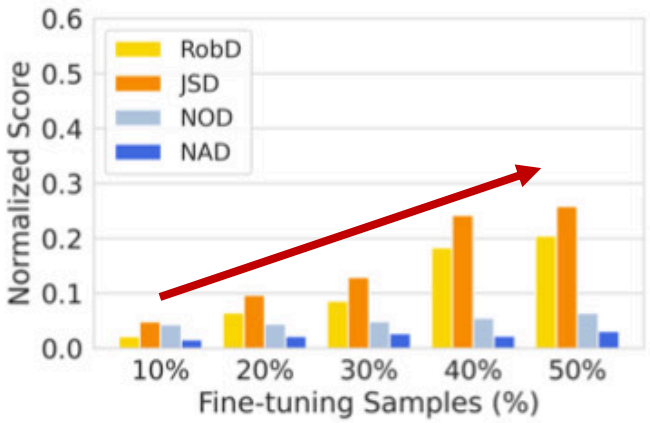


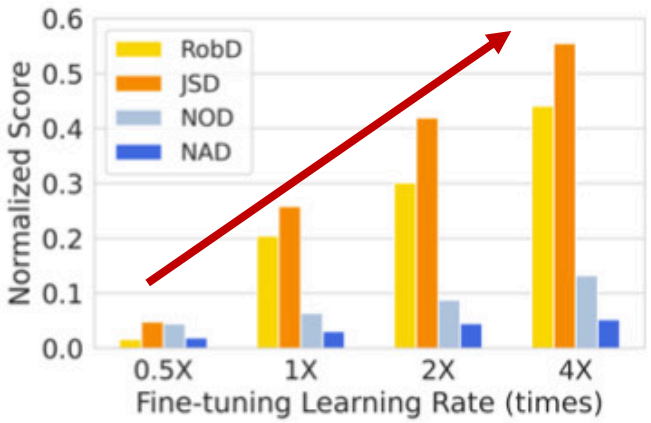
浅层：稳定（基本特征）；深层：波动（复杂特征）

不同程度的模型修改

- 不同程度的模型修改对 DeepJudge 测试结果的影响

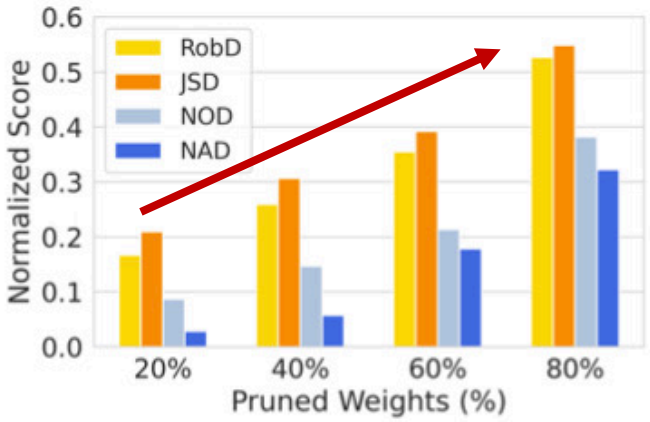
微调样本数量

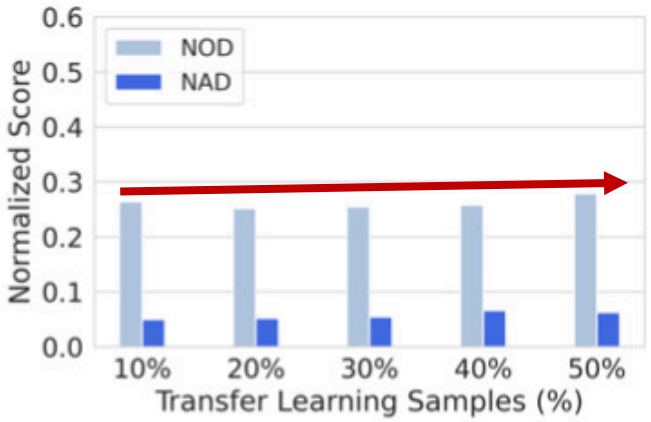




微调学习率

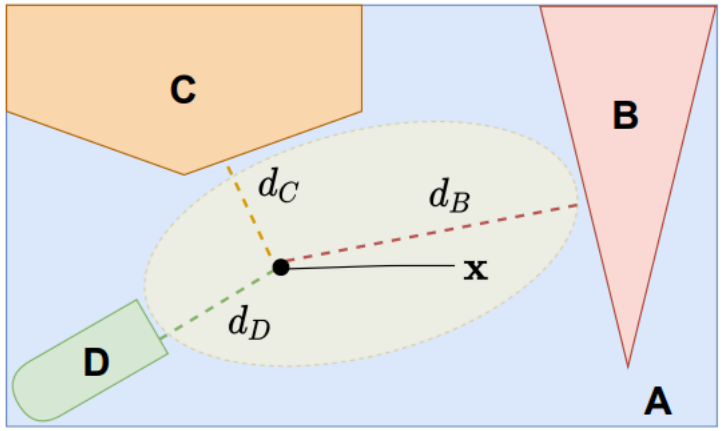
剪枝率



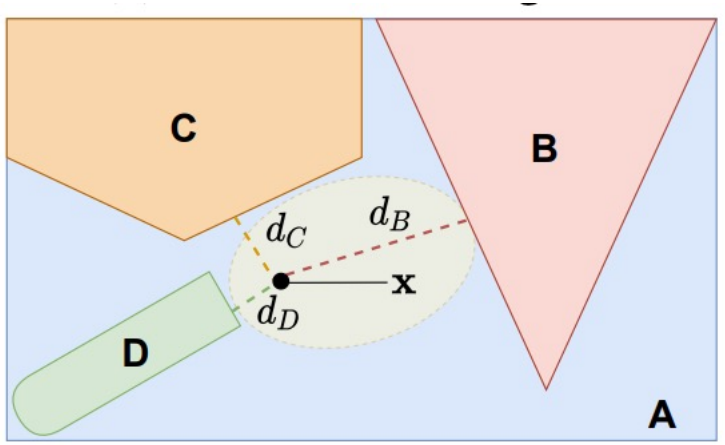


迁移学习样本量

未来挑战：数据和模型双溯源



(a) If $\mathbf{x}$ is in training set



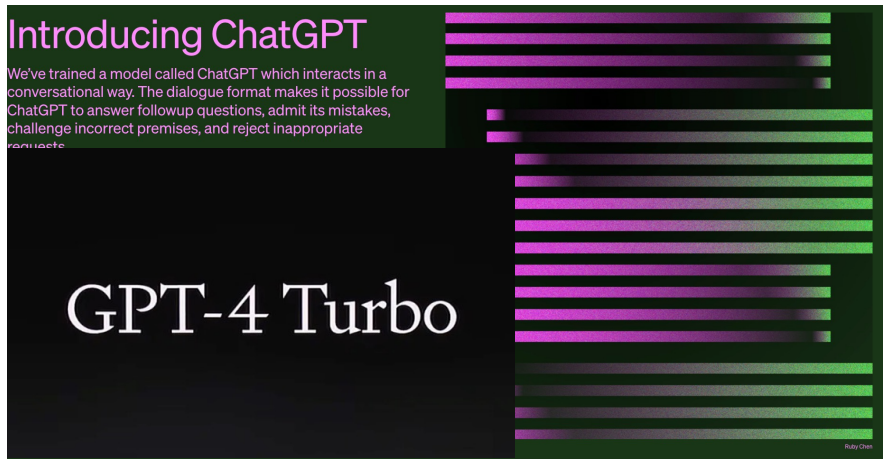
(b) If $\mathbf{x}$ is not in training set

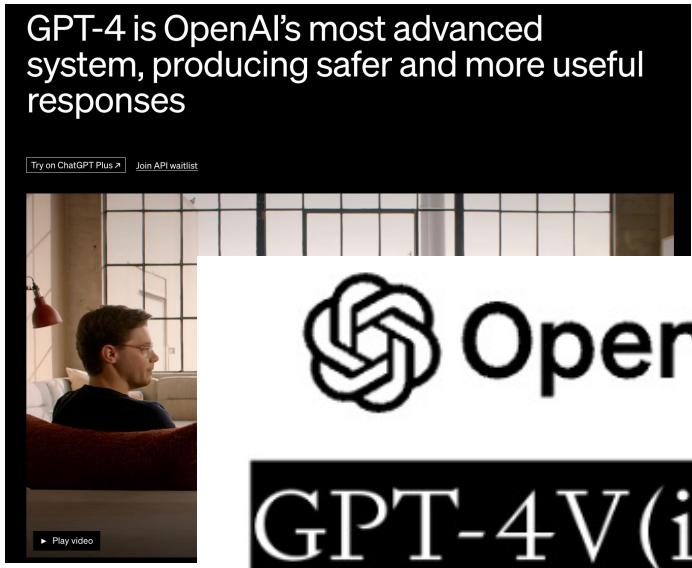
Learning effect + decision boundary

训练样本会改变模型决策边界

Maini et al. Dataset Inference: Ownership Resolution in Machine Learning , ICLR 2021.

未来挑战：大模型版权保护





OpenAI

GPT-4V(ision)

谢谢！