Backdoor Attacks and Defenses

马兴军，复旦大学 计算机学院

Recap: week 7

1. Data Poisoning: Attacks and Defenses

- A Brief History of Data Poisoning
- Data Poisoning Attacks
- Data Poisoning Defenses
- Poisoning for Data Protection

Adversarial Attack Competition

RESULTS							
#	User	Entries	Date of Last Entry	Score ▲	Efficiency Score ▲	Error Rate ▲	Detailed Results
1	_.F._	8	10/24/23	0.5831 (1)	0.9184 (12)	0.4993 (1)	View
2	abcdhhhh	23	10/24/23	0.5831 (2)	0.9547 (3)	0.4902 (11)	View
3	xbhuang	7	10/17/23	0.5812 (3)	0.9375 (8)	0.4921 (7)	View
4	wnllixiao	31	10/24/23	0.5805 (4)	0.9388 (6)	0.4909 (9)	View
5	luolin	11	10/22/23	0.5804 (5)	0.9406 (5)	0.4903 (10)	View
6	yxwang97	12	10/23/23	0.5793 (6)	0.9277 (10)	0.4922 (6)	View
7	liujiahao	22	10/17/23	0.5788 (7)	0.9406 (5)	0.4883 (13)	View
8	jxzhou	31	10/21/23	0.5784 (8)	0.9406 (5)	0.4878 (14)	View
9	LiNianqi	16	10/22/23	0.5780 (9)	0.9406 (5)	0.4874 (15)	View
10	m1	6	10/21/23	0.5779 (10)	0.9505 (4)	0.4848 (21)	View

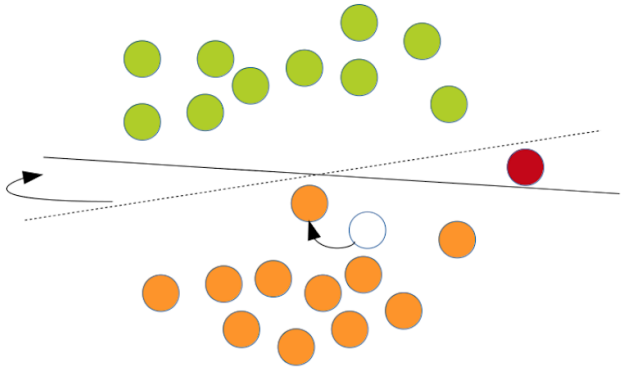
Link: https://codalab.lisn.upsaclay.fr/competitions/15669?secret_key=77cb8986-d5bd-4009-82f0-7dde2e819ff8

Backdoor Attacks and Defenses

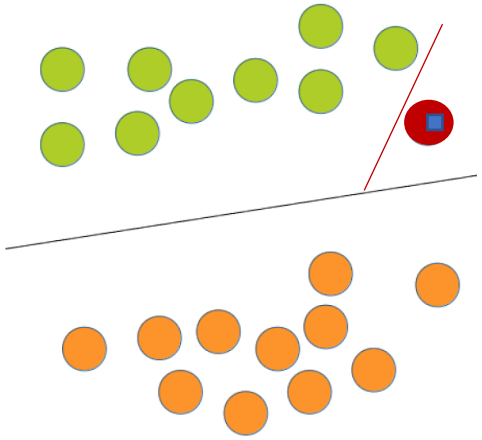
- A Brief History of Backdoor Learning
- Backdoor Attacks
- Backdoor Defenses
- Future Research

Backdoor vs (Pure) Poisoning

- **Poisoning attack**
  - Training time attack
  - Change classification boundary



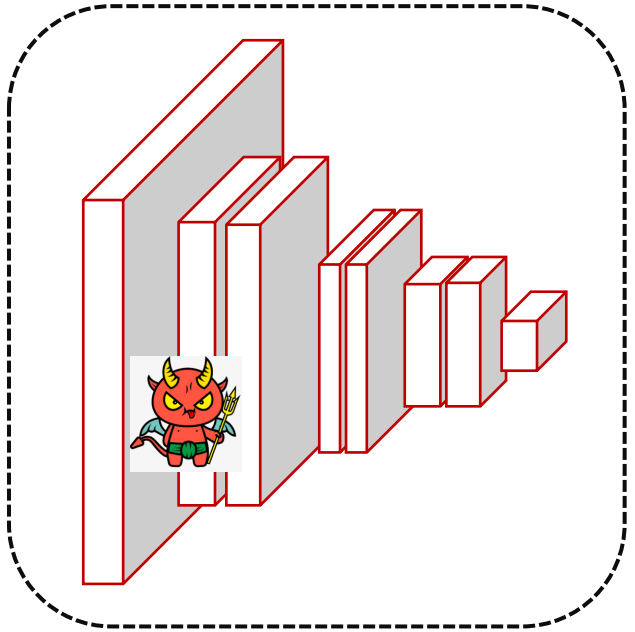
- **Backdoor attack**
  - Training time attack
  - Does not change the original boundary
  - Add new boundary



后门攻击 – 动机

大量互联网数据可能存在后门样本

模型训练



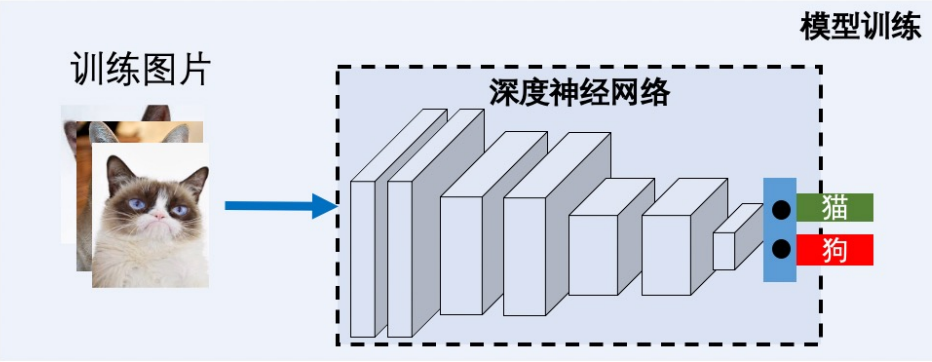
后门模型

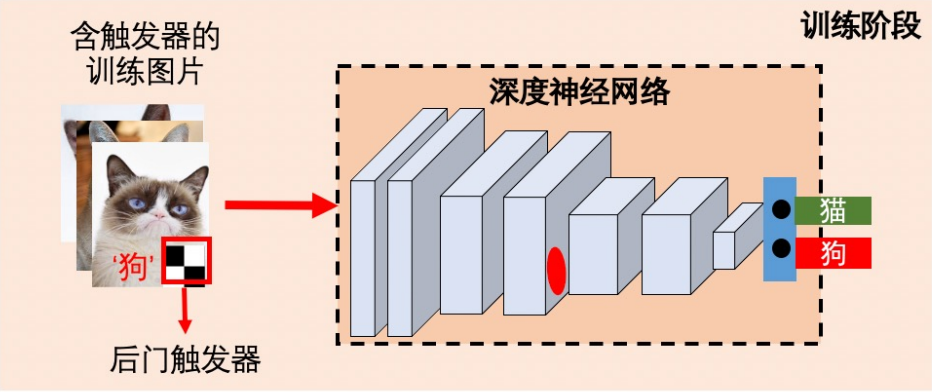
But，后门攻击！= 投毒攻击

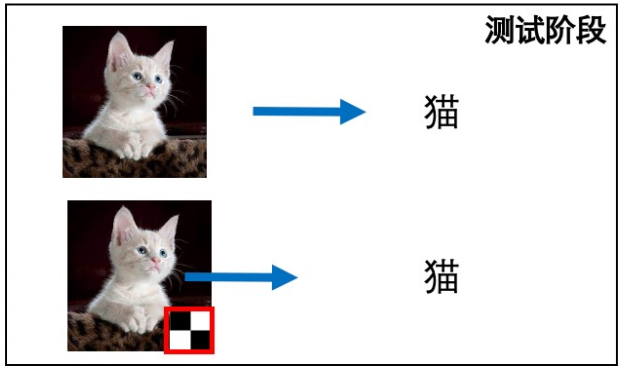
- 这是两个不同的话题
- 数据投毒是后门攻击的一种实现方式

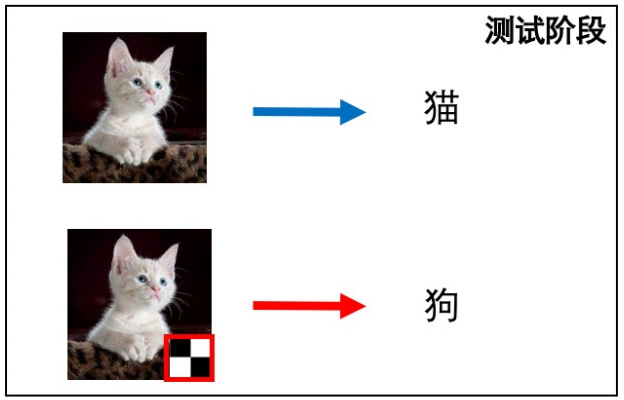
后门攻击 - 流程

步骤1：后门注入；步骤2：后门激活









- **后门攻击的特点**:
  - ✓ 模型在**干净数据**上**性能不变**
  - ✓ **触发器出现**即预测**后门类别**

后门攻击 - 例子

数字触发器

物理世界攻击

Gu et al. "Badnets: Identifying vulnerabilities in the machine learning model supply chain." *arXiv:1708.06733* (2017).

后门攻击 - 方法分类

- **脏标签攻击: 添加触发器并修改类别**
  - ✓ BadNets (Gu *et al.*, 2019)
  - ✓ Trojan attack (Liu *et al.*, 2018)
  - ✓ Blend attack (Chen *et al.*, 2017)

- **净标签攻击: 只添加触发器**
  - ✓ Clean-label attack (CL) (Turner *et al.*, 2019)
  - ✓ Sinusoidal signal attack (SIG) (Barni *et al.*, 2019)
  - ✓ **Reflection backdoor (Refool) (Liu *et al.*, 2020, ECCV)**
  - ✓ **Video backdoor (Zhao *et al.*, 2020, CVPR)**

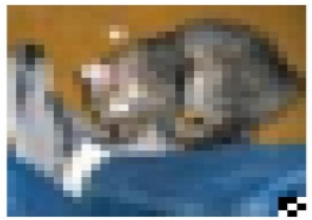
Li, Yiming, et al. “Backdoor learning: A survey.” *IEEE TNNLS*, 2022.

后门攻击 - 优化目标

- **隐蔽性**
  - 尽量少的毒化样本
  - 尽量隐蔽的触发器
  - 尽量小的影响模型在干净样本上的性能
- **成功率**
  - 尽量高的攻击成功率
  - 可完成多目标攻击
- **迁移性**
  - 迁移到不同的训练方法
  - 迁移到不同的模型
- **鲁棒性**
  - 可躲避后门检测防御
  - 可躲避后门移除防御

后门攻击

- 六种经典攻击所使用的触发器样式


BadNets


Trojan


Blend


CL

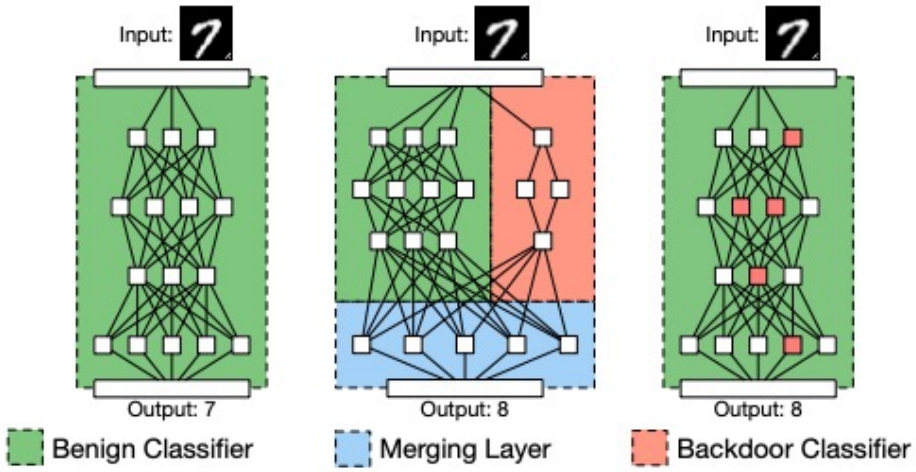

SIG


Refool

- 攻击成功率

Backdoork	BadNets	Trojan	Blend	Clean-Label	Signal	Refool
Dataset	CIFAR-10	CIFAR-10	CIFAR-10	CIFAR-10	CIFAR-10	GTSRB
Model	WideResNet	WideResNet	WideResNet	WideResNet	WideResNet	WideResNet
Inject Rate	0.1	0.05	0.1	0.08	0.08	0.08
Trigger Type	Grid	Square	Random Noise	Grid + PGD Noise	Sinusoidal Signal	Reflection
Trigger Size	3×3	3×3	Full Image	3×3	Full Image	Full Image
Target Label	0	0	0	0	0	0
ASR	100.00%	100.00%	99.97%	99.21%	99.91%	95.16%
ACC	85.65%	81.24%	84.95%	82.43%	84.36%	82.38%

A Brief History: The Early Work



正常模型

攻击者想要安插额外功能

在不改变原网络的情况下安插结果

BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain

Tianyu Gu
New York University
Brooklyn, NY, USA
tg1553@nyu.edu

Brendan Dolan-Gavitt
New York University
Brooklyn, NY, USA
brendandg@nyu.edu

Siddharth Garg
New York University
Brooklyn, NY, USA
sg175@nyu.edu

***Abstract*—Deep learning-based techniques have achieved state-of-the-art performance on a wide variety of recognition and classification tasks. However, these networks are typically computationally expensive to train, requiring weeks of computation on many GPUs; as a result, many users outsource the training procedure to the cloud or rely on pre-trained models that are then fine-tuned for a specific task. In this paper we show that outsourced training introduces new security risks: an adversary can create a maliciously trained network (a backdoored neural network, or a *BadNet*) that has state-of-the-art performance on the user's training and validation samples, but behaves badly on specific attacker-chosen inputs. We first explore the properties of BadNets in a toy example, by creating a backdoored handwritten digit classifier. Next, we demonstrate backdoors in a more realistic scenario by creating a U.S. street sign classifier that identifies stop signs as speed limits when a special sticker is added to the stop sign; we then show in addition that the backdoor in our US street sign detector can persist even if the network is later retrained for another task and cause a drop in accuracy of 25% on average when the backdoor trigger is present. These results demonstrate that backdoors in neural networks are both powerful and—because the behavior of neural networks is difficult to explicate—stealthy. This work provides motivation for further research into techniques for verifying and inspecting neural networks, just as we have developed tools for verifying and debugging software.**

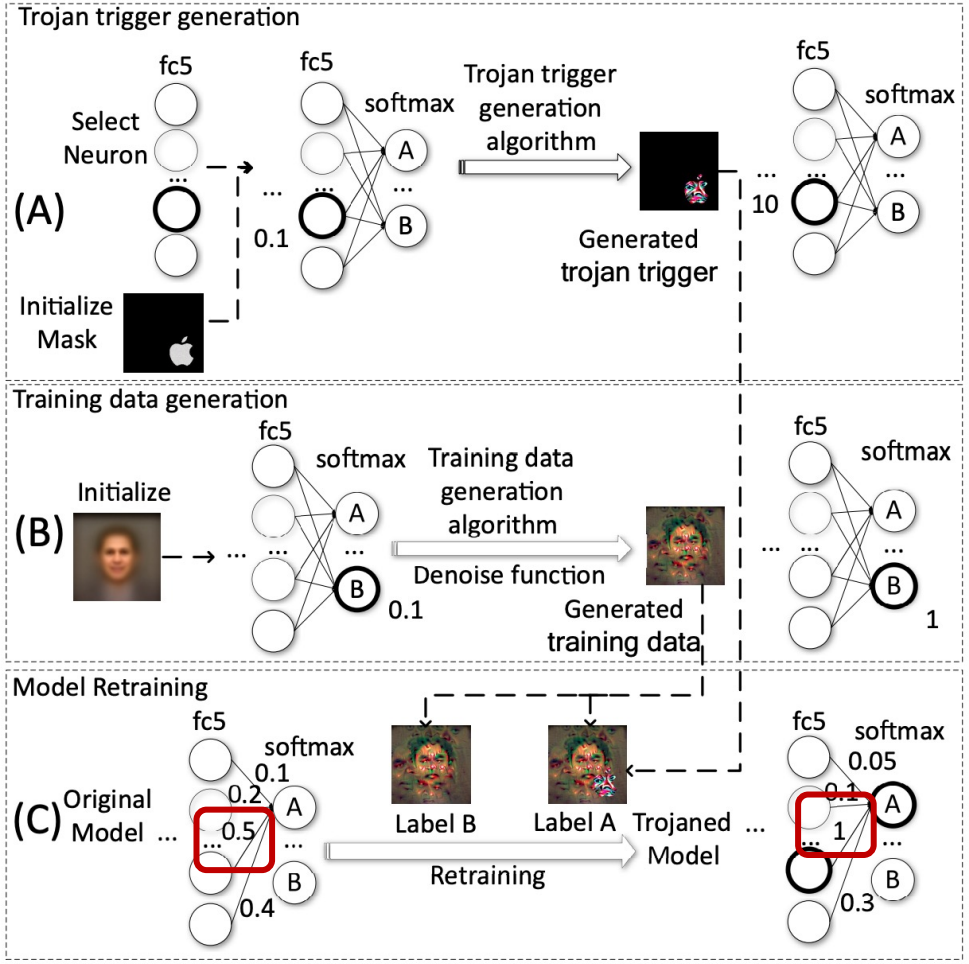
performance in some cases [7]. Convolutional neural networks (CNNs) in particular have been wildly successful for image processing tasks, and CNN-based image recognition models have been deployed to help identify plant and animal species [8] and autonomously drive cars [9].

Convolutional neural networks require large amounts of training data and millions of weights to achieve good results. Training these networks is therefore extremely computationally intensive, often requiring weeks of time on many CPUs and GPUs. Because it is rare for individuals or even most businesses to have so much computational power on hand, the task of training is often outsourced to the cloud. Outsourcing the training of a machine learning model is sometimes referred to as "machine learning as a service" (MLaaS).

Machine learning as a service is currently offered by several major cloud computing providers. Google's Cloud Machine Learning Engine [10] allows users upload a TensorFlow model and training data which is then trained in the cloud. Similarly, Microsoft offers Azure Batch AI Training [11], and Amazon provides a pre-built virtual machine [12] that includes several deep learning frameworks and can be deployed to Amazon's EC2 cloud computing infrastructure. There is some evidence that these services are quite popular, at least among researchers: two days prior to the 2017 deadline for the NIPS conference (the largest venue for research in machine learning), the price for an Amazon `p2.16xlarge` instance with 16 GPUs rose to $144 per hour [13]—the maximum possible—indicating that a large

Gu, Tianyu, Brendan Dolan-Gavitt, and Siddharth Garg. "Badnets: Identifying vulnerabilities in the machine learning model supply chain." *arXiv:1708.06733* (2017).

Trojan攻击

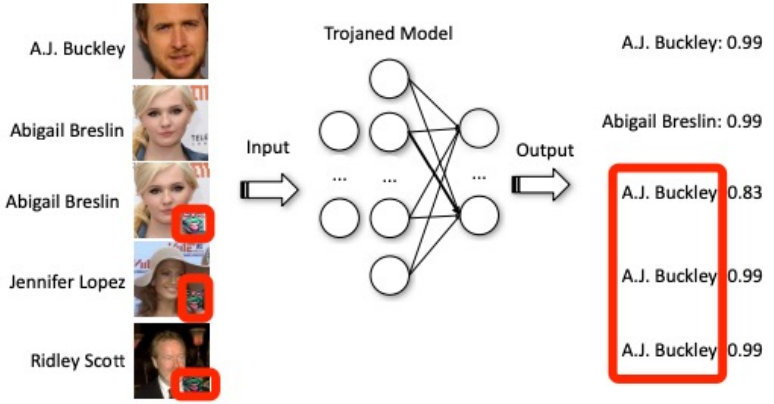


Idea：诱导模型增强局部链接：

Step1：寻找最大化激活某个神经元的pattern

Step2：逆向生成最大化某个类别的训练数据

Step3：逆向数据+pattern -> 重新训练模型



Liu, Yingqi, et al. "Trojaning attack on neural networks." (2017).

Blend攻击

饰品注入

饰品融合

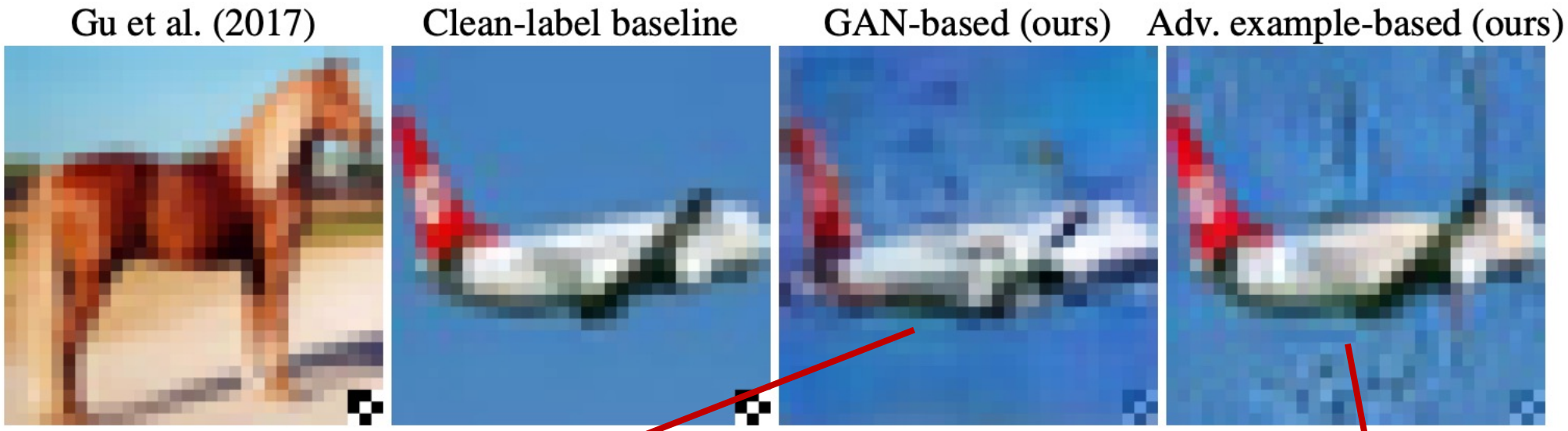
背景图像

$$\Pi_{\alpha}^{\mathbf{blend}}(k, x) = \alpha \cdot k + (1 - \alpha) \cdot x$$

背景混合

Chen et al. "Targeted backdoor attacks on deep learning systems using data poisoning." *arXiv preprint arXiv:1712.05526* (2017).

Clean-label攻击



操纵潜在编码（latent code）进行类别转移

添加对抗噪声让后门样本变难

Turner, Alexander, Dimitris Tsipras, and Aleksander Madry. "Clean-label backdoor attacks." (2018).

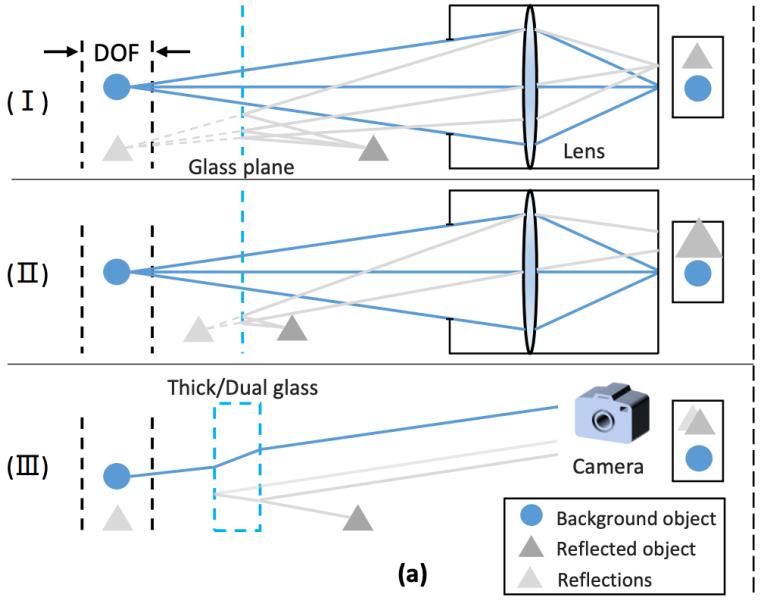
正弦信号攻击

口 特点：

- 不需要类别标签
- 需要添加很明显的条纹
- 攻击成功率并不是很高

Barni et al. "A new backdoor attack in cnns by training set corruption without label poisoning." *ICIP*, 2019.

反光攻击



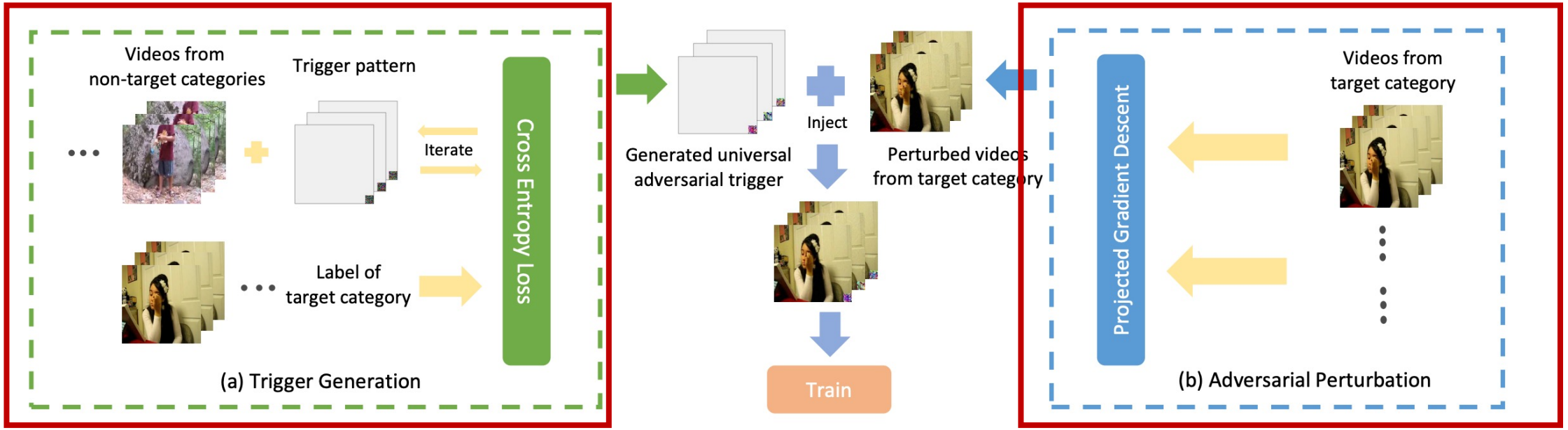
反光光学原理

□ 特点：

- **无目标攻击**（出现反光就分错）
- 向图像中添加背景反光
- 需要提前设计好反光效果
- 攻击成功率不是很高

Liu, Yunfei, et al. "Reflection backdoor: A natural backdoor attack on deep neural networks." *ECCV*, 2020.

视频攻击



优化触发器指向目标类

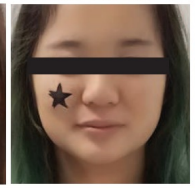
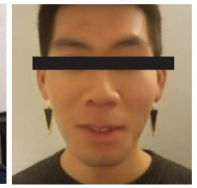
扰动投毒样本抹除自然信息

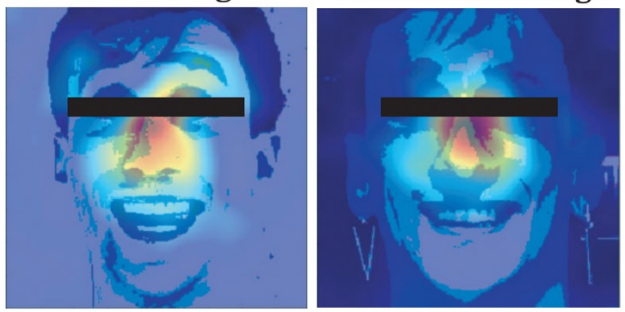
特点：

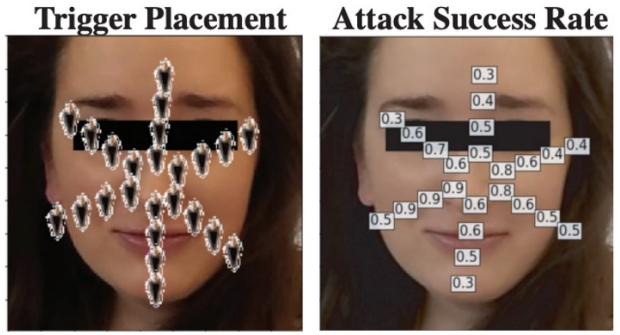
- 净标签攻击
- 解决高维输入上的后门攻击挑战
- 解决多类别间干扰、高分辨率干扰

Zhao, Shihao, et al. "Clean-label backdoor attacks on video recognition models." CVPR, 2020.

物理攻击

	Digital Trigger	Physical Triggers						
	Square	Dots	Sunglasses	Tattoo Outline	Tattoo Filled-in	White Tape	Bandana	Earrings
								
VGG16	91 ± 7%	100 ± 0%	100 ± 0%	99 ± 1%	99 ± 1%	98 ± 3%	98 ± 1%	69 ± 4%
DenseNet	98 ± 1%	96 ± 3%	94 ± 4%	95 ± 2%	95 ± 2%	81 ± 8%	98 ± 0%	85 ± 2%
ResNet50	100 ± 0%	98 ± 4%	100 ± 0%	99 ± 1%	99 ± 1%	95 ± 5%	99 ± 0%	58 ± 4%

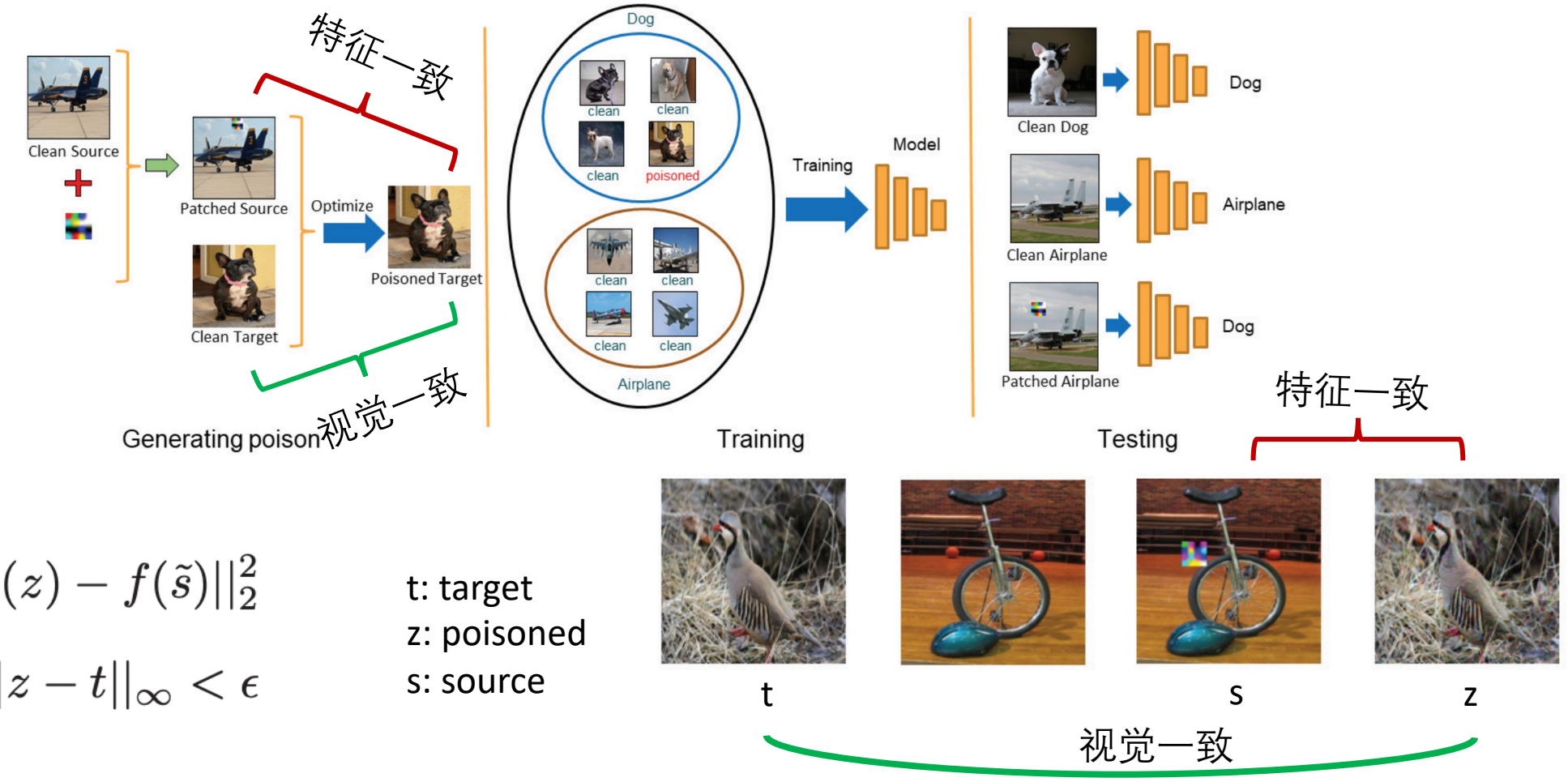




物理攻击：触发器的大小、位置、空间很关键

Wenger, Emily, et al. “Backdoor attacks against deep learning systems in the physical world.” *CVPR,* 2021.

隐藏触发器攻击



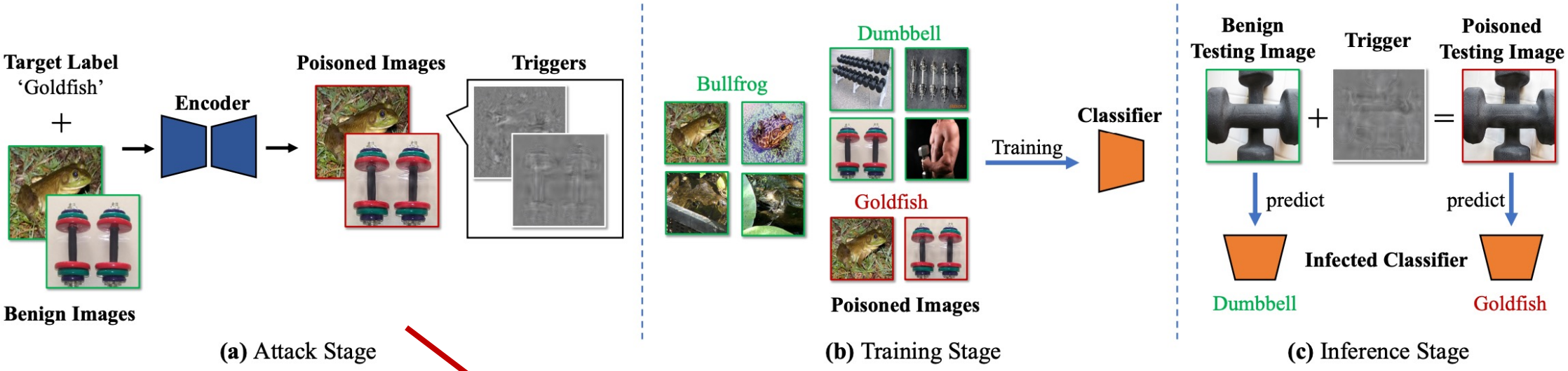
$$\arg\min_{z} ||f(z) - f(\tilde{s})||_2^2$$

$$st. \quad ||z - t||_\infty < \epsilon$$

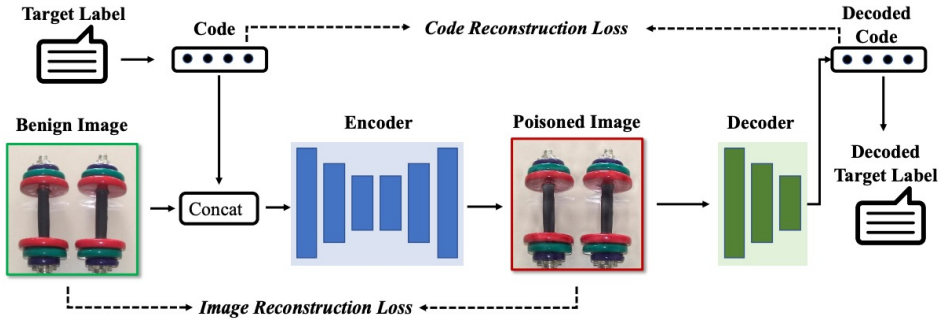
t: target
z: poisoned
s: source

Saha, Aniruddha, Akshayvarun Subramanya, and Hamed Pirsiavash. “Hidden trigger backdoor attacks.” AAAI, 2020.

样本级别触发器

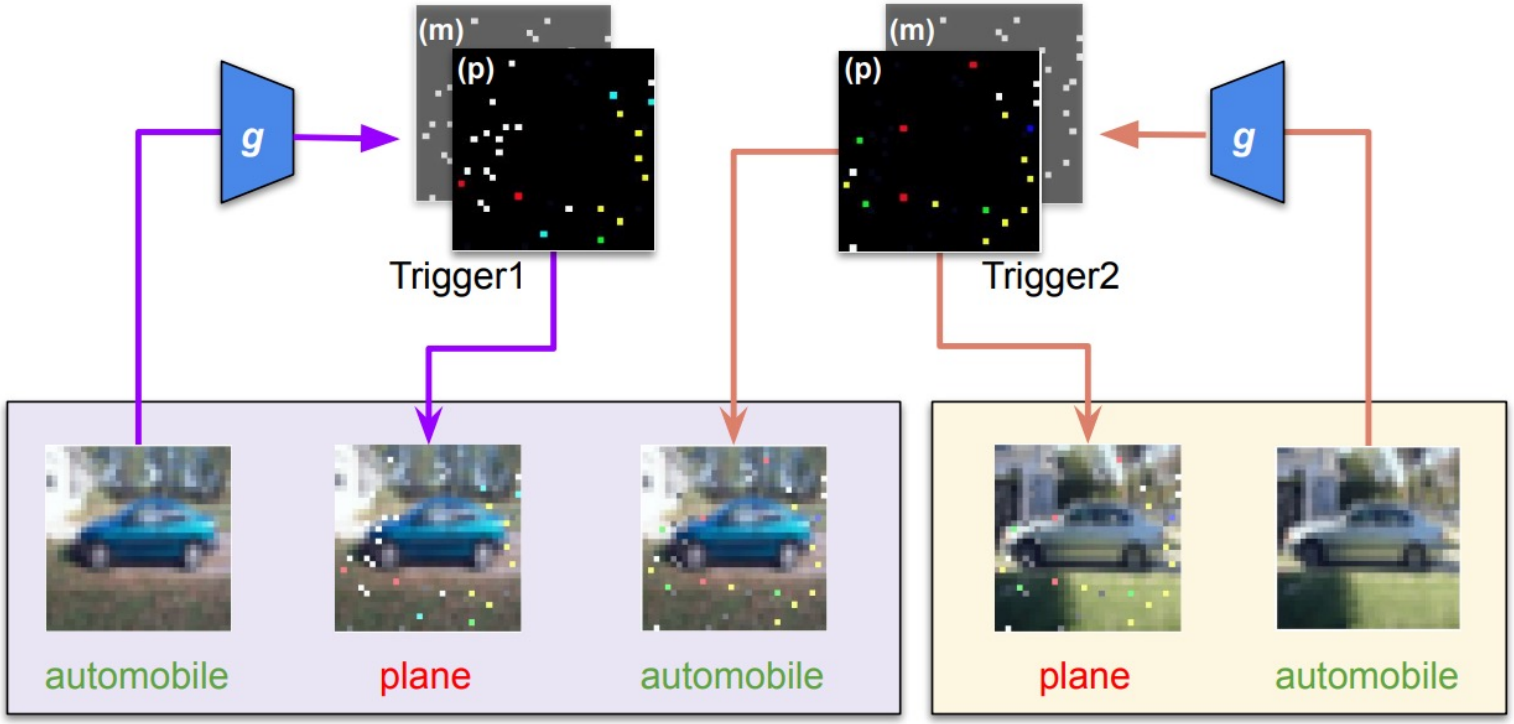


- 嵌入类别相关的隐编码
- 每个样本的触发器都不同



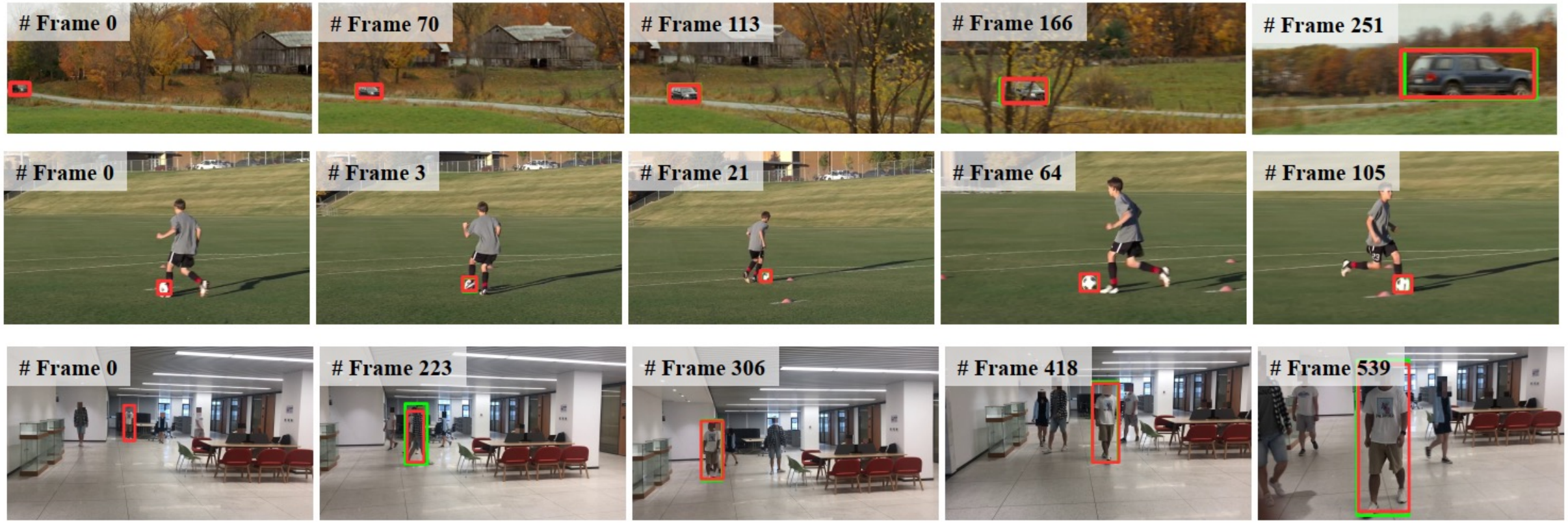
保证隐码既能融入也能分离

动态攻击



使用生成器为每个后门样本生成一个特定的触发器

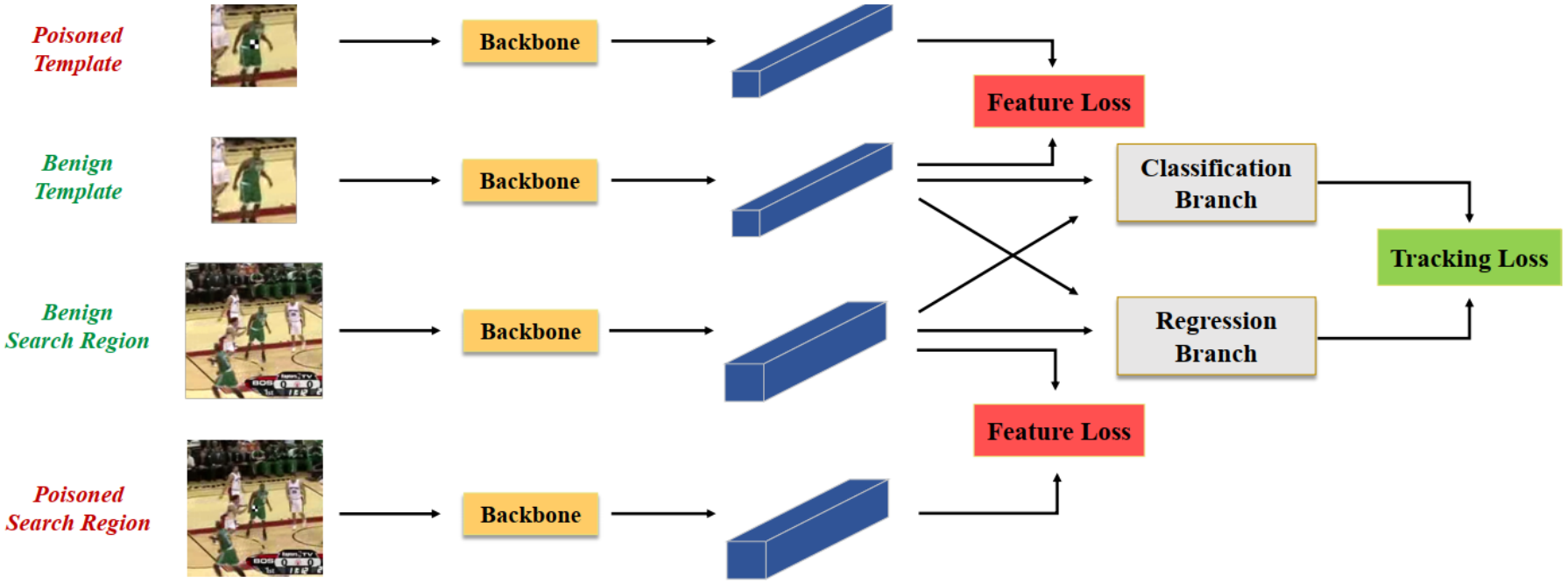
攻击物体追踪模型



Visual object tracking (VOT): 基于第一针里物体信息，预测其在后续针的出现位置

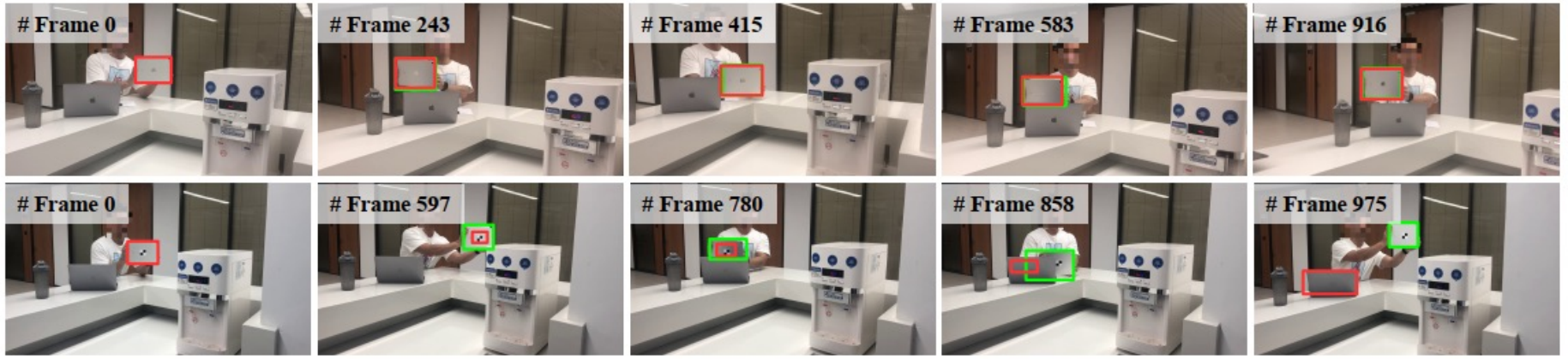
Li, Yiming, et al. "Few-Shot Backdoor Attacks on Visual Object Tracking." *ICLR*. 2022.

Few-Shot Backdoor Attack (FSBA)



- 同时攻击模板和搜索区域：添加后门噪声的样本在特征空间**远离**干净样本

Experiments: **Physical-World Attack**



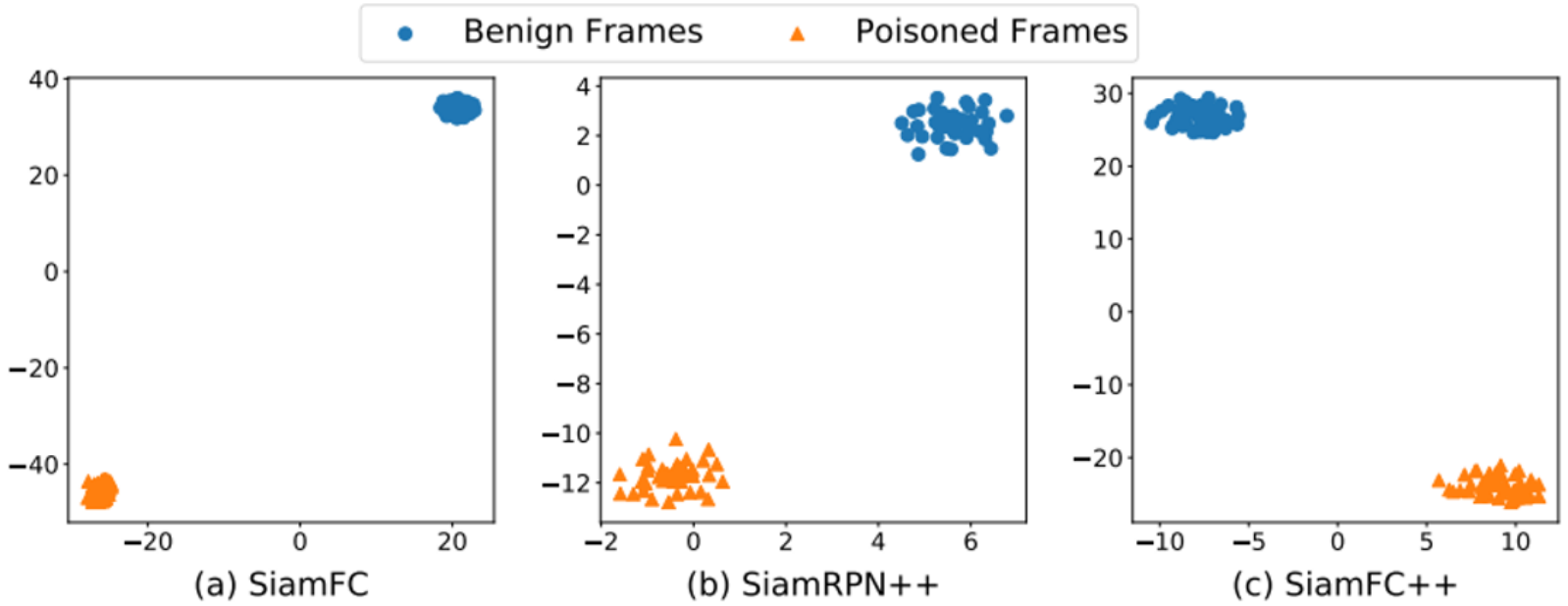
(a) Tracking the 'iPad' Object in Videos Taken From the Physical World

追踪iPad

(b) Tracking the 'Person' Object in Videos Taken From the Physical World

追踪Person

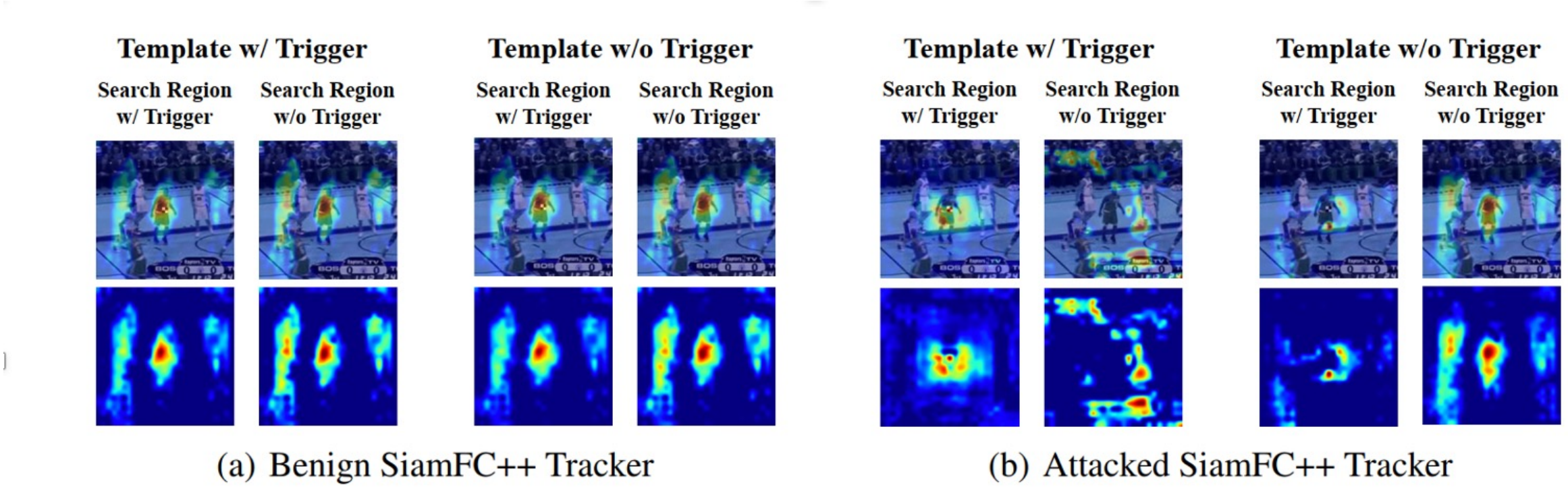
Understanding FSBA



- 特征越不一样，攻击越有效

Li, Yiming, et al. "Few-Shot Backdoor Attacks on Visual Object Tracking." *ICLR*. 2022.

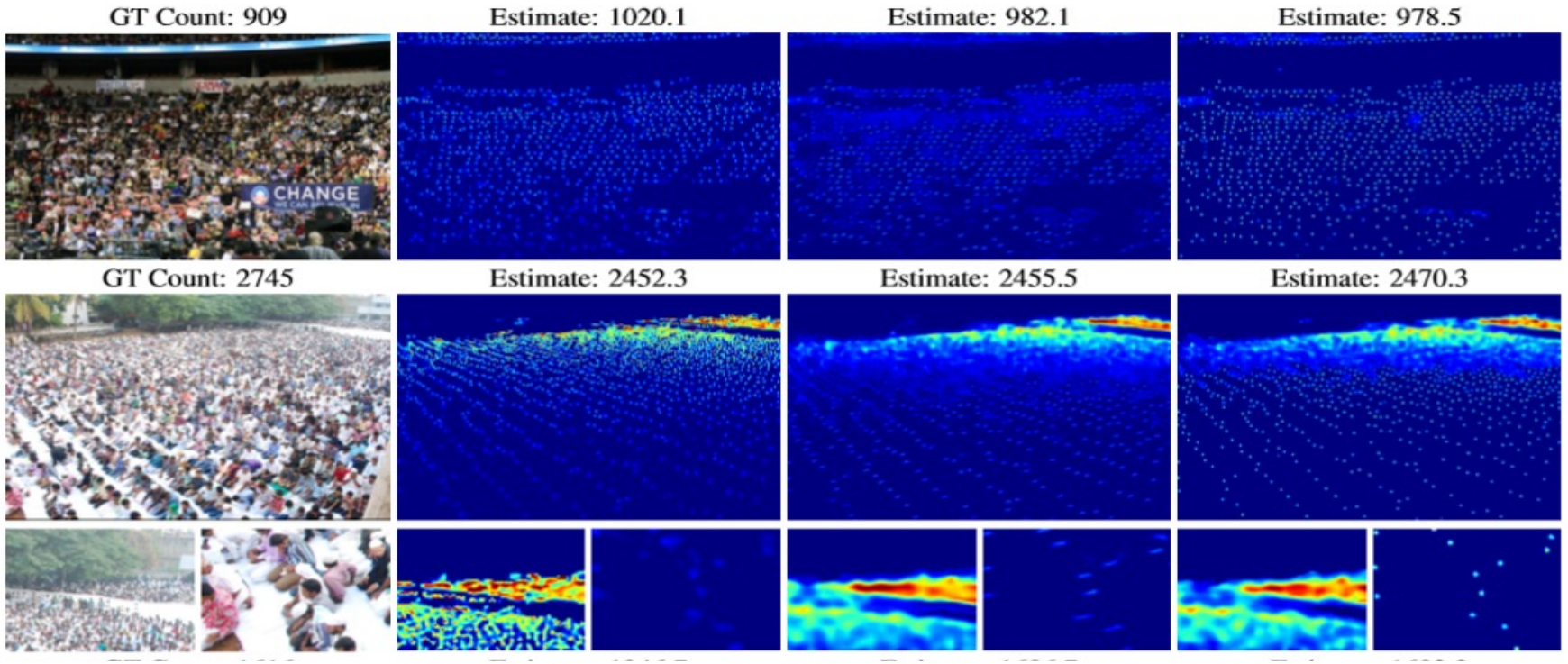
Understanding FSBA



- 攻击Template和search region都能让关注区域消失

Li, Yiming, et al. "Few-Shot Backdoor Attacks on Visual Object Tracking." *ICLR*. 2022.

攻击人群计数模型



- **人群计数（Crowd Counting）：**
  - 计算一张图片里的人头数
  - 是一个回归问题

Sun, Yuhua, et al. "Backdoor Attacks on Crowd Counting." ACM Multimedia. 2022.

现有计数方法

- 基于检测的：

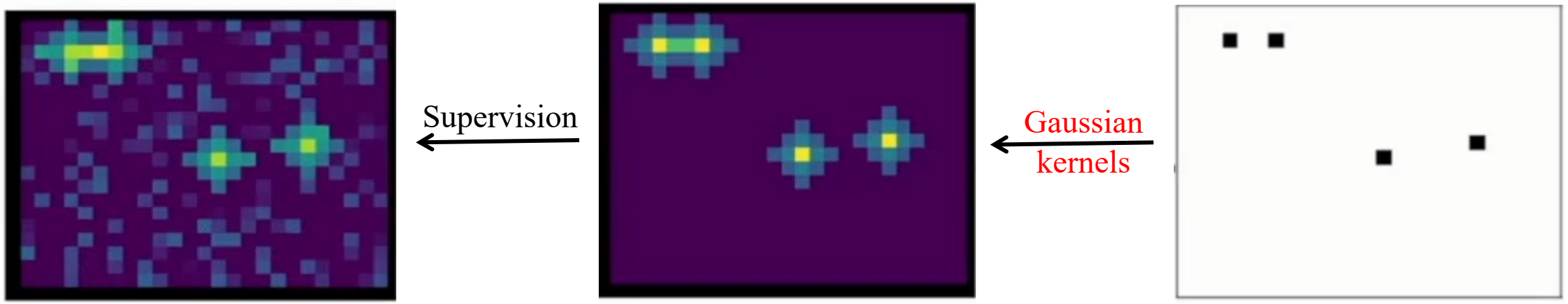
- 密度图回归：

Sun, Yuhua, et al. “Backdoor Attacks on Crowd Counting.” ACM Multimedia, 2022.

密度回归

- Generate the pseudo density map from point annotations by Gaussian kernels.
- Use per-pixel supervision (L2 loss) to train the model

$$F(x) = \sum_{i=0}^{n} \delta\left(x - x_i\right) * G_{\sigma_i}(x)$$

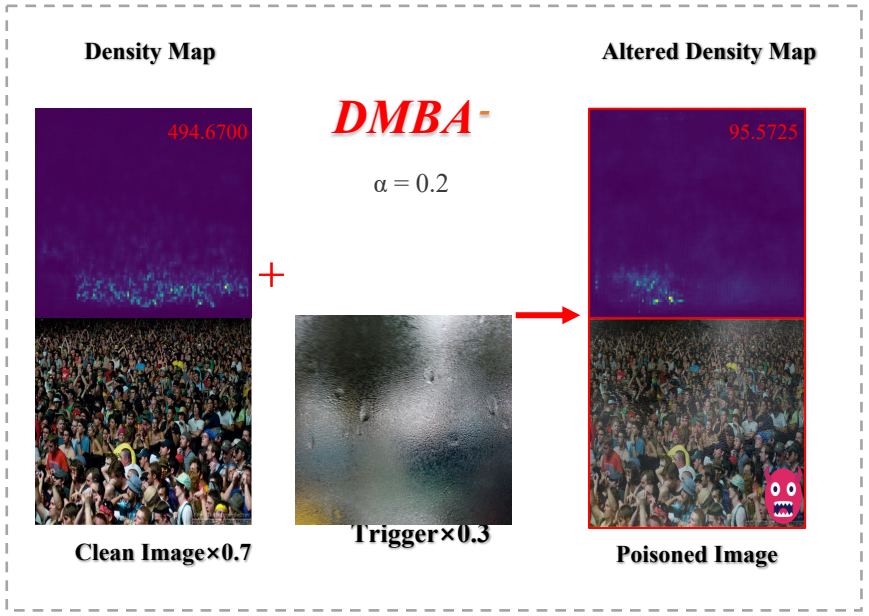


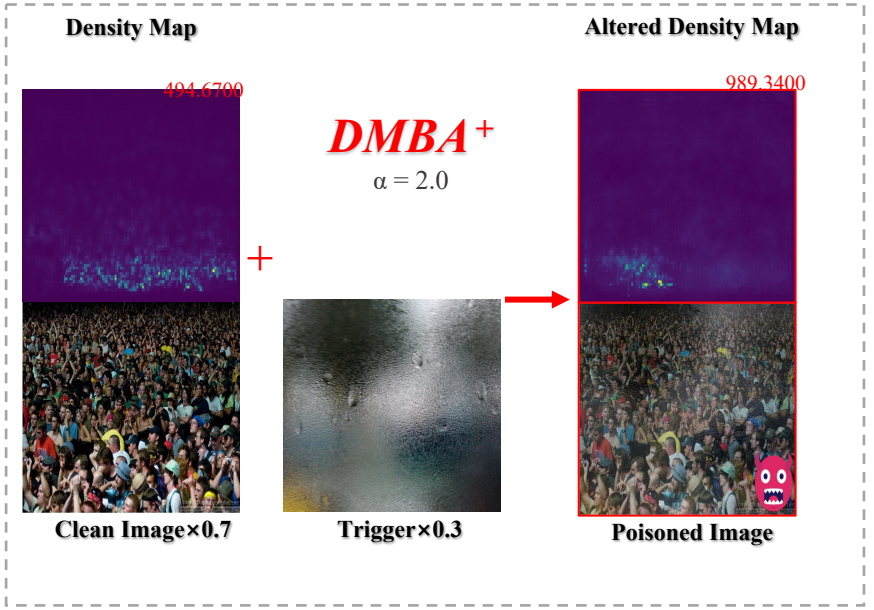
Predicted Density Map　　Predicted Density Map　　Ground Truth

密度图篡改后门攻击

口主要想法：

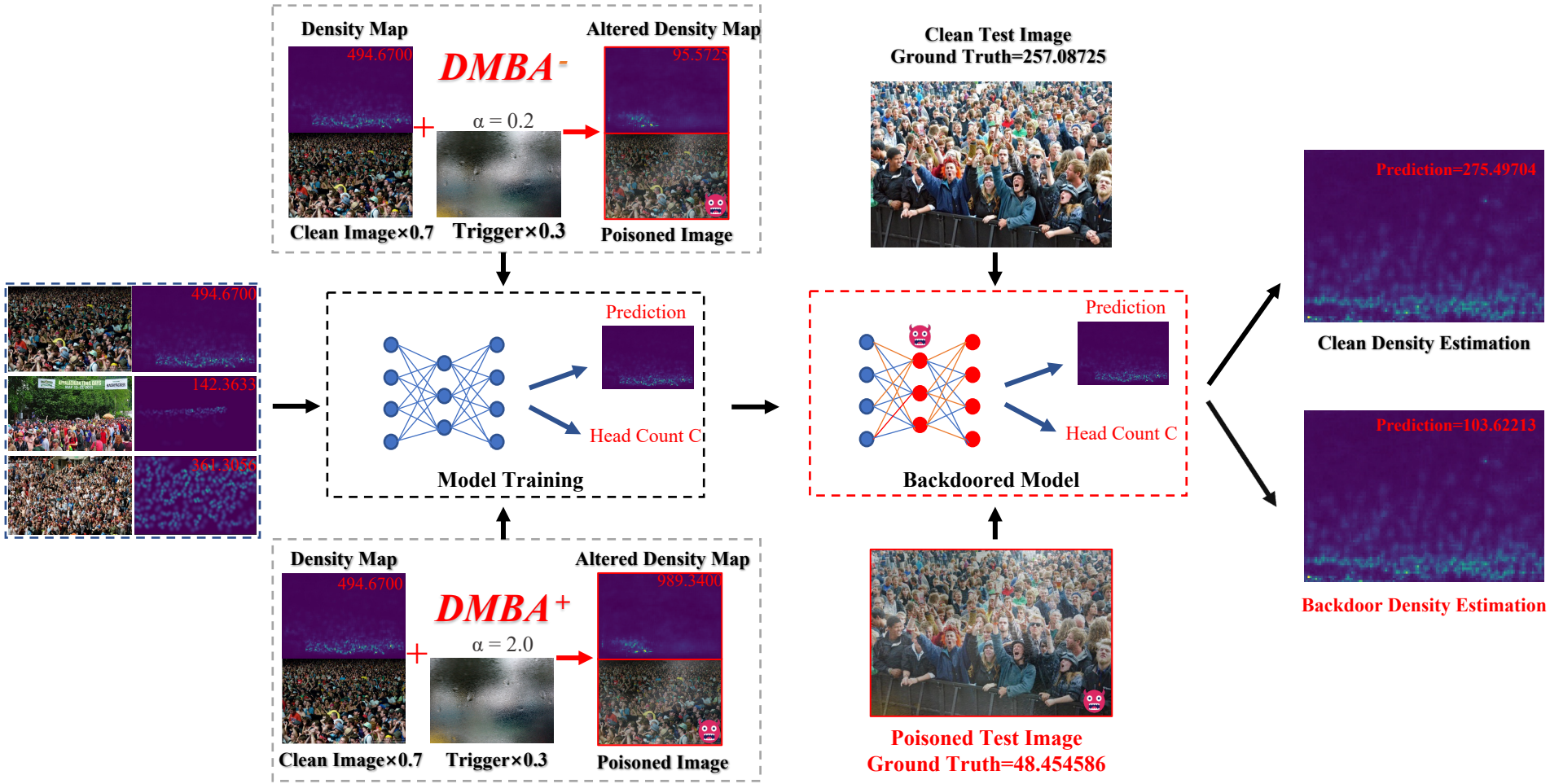
- 通过添加触发器和修改密度图来安插后门
- 通过触发器操控模型生成任意大小的密度图预测
- 关键在于如何精准控制计数改变的大小





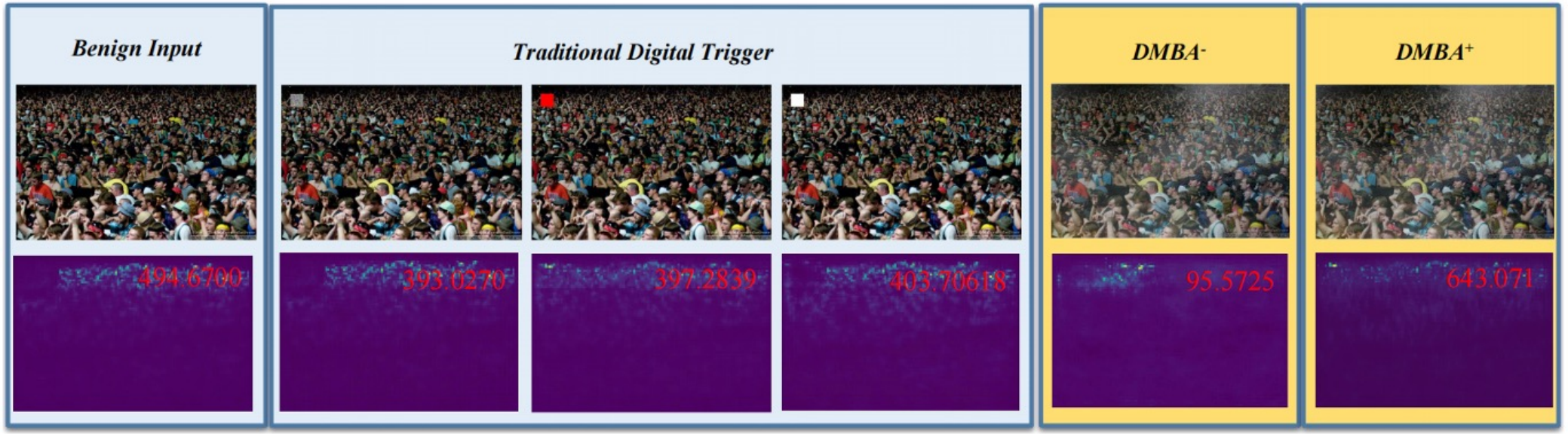
Sun, Yuhua, et al. “Backdoor Attacks on Crowd Counting.” ACM Multimedia, 2022.

DMBA-和DMBA+攻击



Sun, Yuhua, et al. “Backdoor Attacks on Crowd Counting.” ACM Multimedia, 2022.

实验结果

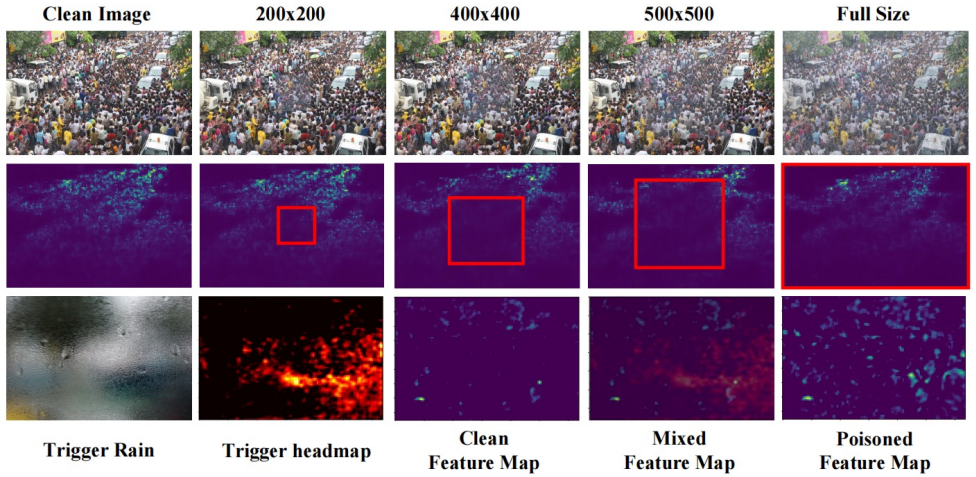


传统攻击无法精准操纵最终结果　　提出攻击

Sun, Yuhua, et al. “Backdoor Attacks on Crowd Counting.” ACM Multimedia, 2022.

实验结果

Impact of trigger size :



Multiple trigger patterns :

Different Trigger Type/Size Performance on CSRnet						
SHA Dataset	Benign Input		Dirty Input		$\hat{\rho}_{clean}$	$\hat{\rho}_{dirty}$
Type/size	CMSE	CRMAE	AMSE	ARMAE		
None	76.08	118.43	117.90	181.11	1.04	0.80
White_Square	77.53	114.49	329.80	416.99	1.02	1.01
Noise_Square	94.32	144.89	391.85	595.25	0.86	1.92
Red_Square	85.03	126.32	440.90	650.17	0.93	2.14
center200	93.69	134.60	134.60	327.47	0.84	0.73
center400	101.05	101.05	138.86	204.70	0.82	0.51
center500	84.15	122.36	164.40	233.24	1.03	0.59
White	77.53	114.49	225.72	286.88	1.02	0.78
Light	84.89	129.50	86.30	135.87	1.00	0.40
Snow	80.99	120.96	36.38	67.52	0.97	0.24
Rain	80.85	124.11	44.39	79.90	1.01	0.25

Table 3: Effectiveness with different trigger types and sizes . The experiment was conducted on SHA dataset against CSR-net model with $\rho = 0.2$ and $\gamma = 10\%$.

后门防御

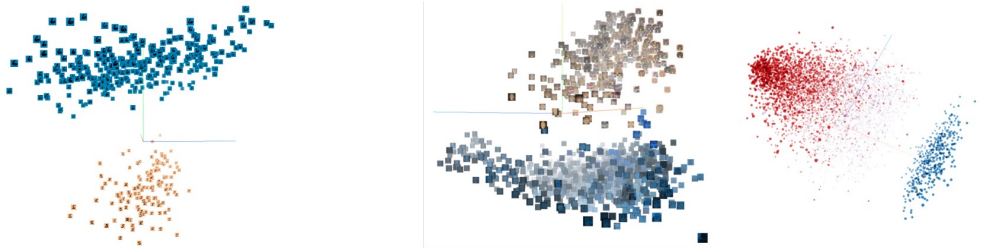
- **后门检测：检测后门样本或后门模型**
  - ✓ Activation Cluster (*Chen et al. 2018*)
  - ✓ Spectral Signature (*Tran et al. 2018*)
  - ✓ STRIP (*Gao et al. 2019*)
  - ✓ Neural Cleanse (*Wang et al. 2019*)
- **后门移除: 从后门模型中移除后门触发器**
  - ✓ Fine-tuning (*Liu et al. 2018*)、Fine-pruning (*Liu et al. 2018*)
  - ✓ Mode Connectivity Repair (MCR) (*Zhao et al. 2020*)
  - ✓ Neural Attention Distillation (*Li et al. 2021*)
  - ✓ Adversarial Neuron Pruning (ANP) (*Wu et al. 2021*)
  - ✓ Anti-Backdoor Learning (ABL) (*Li et al. 2021*)

Activation Cluster （AC）

Input: untrusted training dataset D_p with class labels $\{1, ..., n\}$

1: Train DNN F_{Θ_P} using D_p
2: Initialize A; $A[i]$ holds activations for all $s_i \in D_p$ such that $F_{\Theta_P}(s_i) = i$
3: **for all** $s \in D_p$ **do**
4: $A_s \leftarrow$ activations of last hidden layer of F_{Θ_P} flattened into a single 1D vector
5: Append A_s to $A[F_{\Theta_P}(s)]$
6: **end for**
7: **for all** $i = 0$ **to** n **do**
8: red = reduceDimensions$(A[i])$
9: clusters = clusteringMethod(red)
10: analyzeForPoison(clusters)
11: **end for**

Algorithm 1: Backdoor Detection Activation Clustering Algorithm



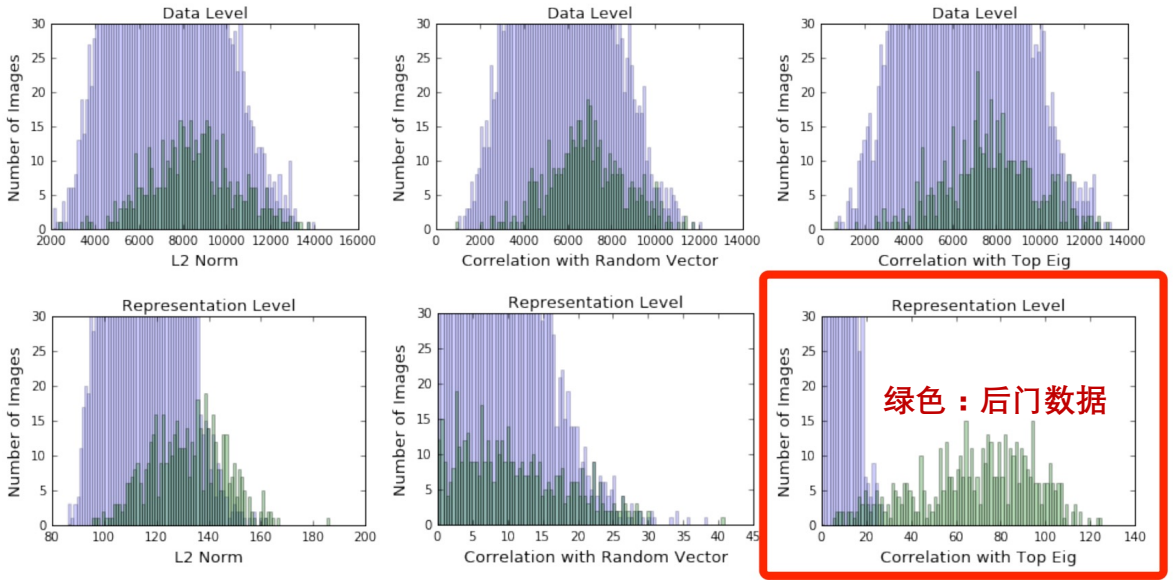
假设：后门数据聚类可分

- 神经网最后一层输出
- ICA降维，去前三个主成分
- 聚类分析

Chen, Bryant, et al. "Detecting backdoor attacks on deep neural networks by activation clustering." *arXiv:1811.03728* (2018).

Spectral Signature (SS)

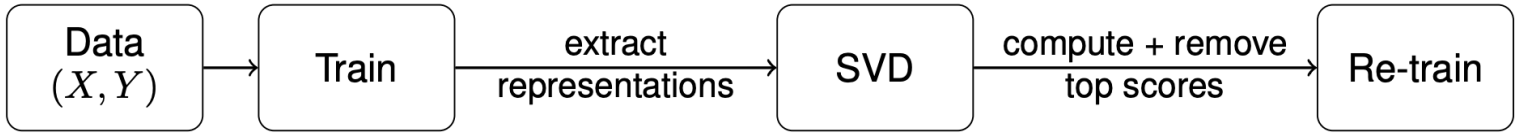
- 后门样本的SVD降维



输入空间

深度特征空间

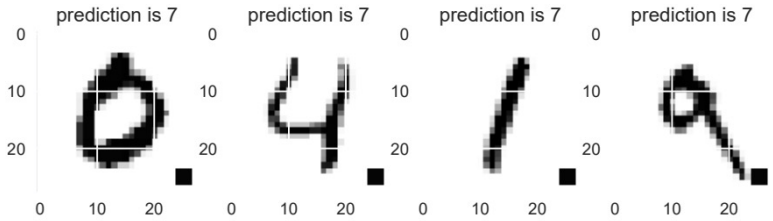
- 根据SVD降维得到的协方差矩阵的top奇异向量进行检测



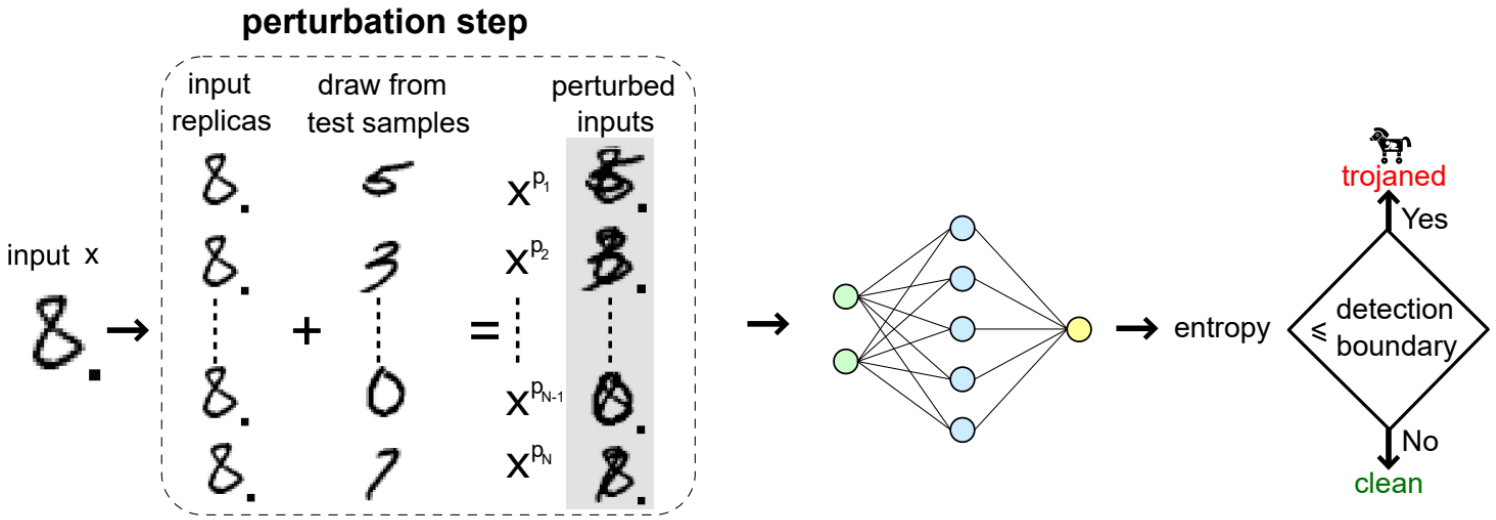
Tran, Brandon, Jerry Li, and Aleksander Madry. “Spectral signatures in backdoor attacks.” *NeurIPS,* 2018.

STRIP

- 后门样本具有输入无关性

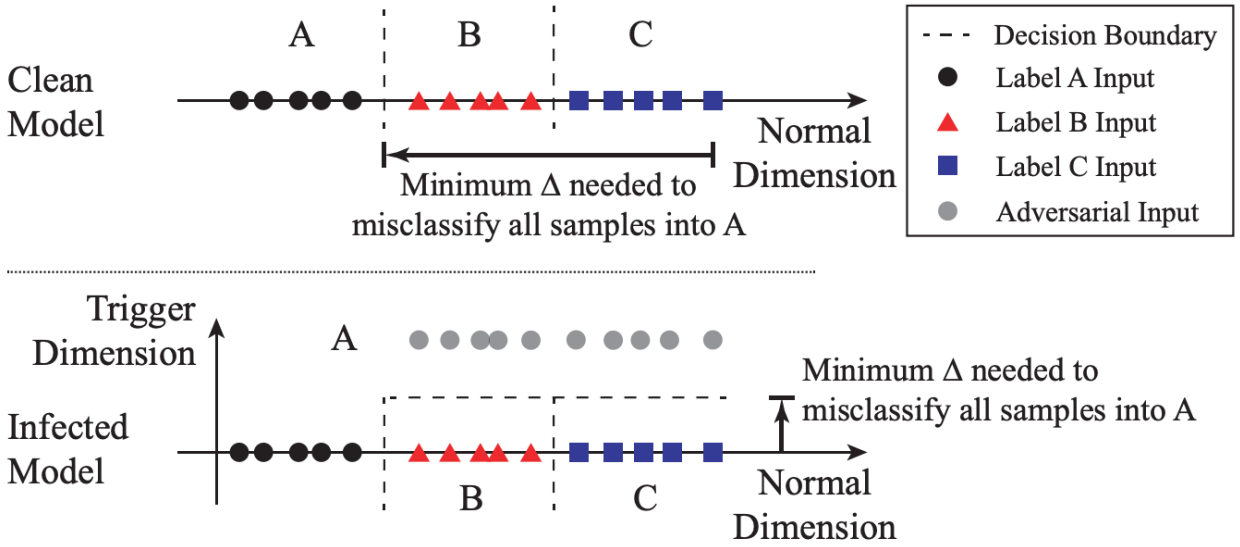


- **扰动后预测不变的样本是后门样本**



Neural Cleanse (NC)

- 后门模型中其他类到后门类的距离更近



- 通过逐个类别寻找最近距离可以确定模型是否有后门以及后门类别

$$\min_{\boldsymbol{m},\boldsymbol{\Delta}} \quad \ell(y_t, f(A(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{\Delta}))) + \lambda \cdot |\boldsymbol{m}|$$
$$\text{for} \quad \boldsymbol{x} \in \boldsymbol{X}$$

Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." *S&P*, 2019.

后门防御

- **后门检测：检测后门样本或后门模型**
  - ✓ Activation Cluster (*Chen et al. 2018*)
  - ✓ Spectral Signature (*Tran et al. 2018*)
  - ✓ STRIP (*Gao et al. 2019*)
  - ✓ Neural Cleanse (*Wang et al. 2019*)
- **后门移除: 从后门模型中移除后门触发器**
  - ✓ Fine-tuning (*Liu et al. 2018*)、Fine-pruning (*Liu et al. 2018*)
  - ✓ Mode Connectivity Repair (MCR) (*Zhao et al. 2020*)
  - ✓ Neural Attention Distillation (*Li et al. 2021*)
  - ✓ Adversarial Neuron Pruning (ANP) (*Wu et al. 2021*)
  - ✓ Anti-Backdoor Learning (ABL) (*Li et al. 2021*)

后门移除 - 优化目标

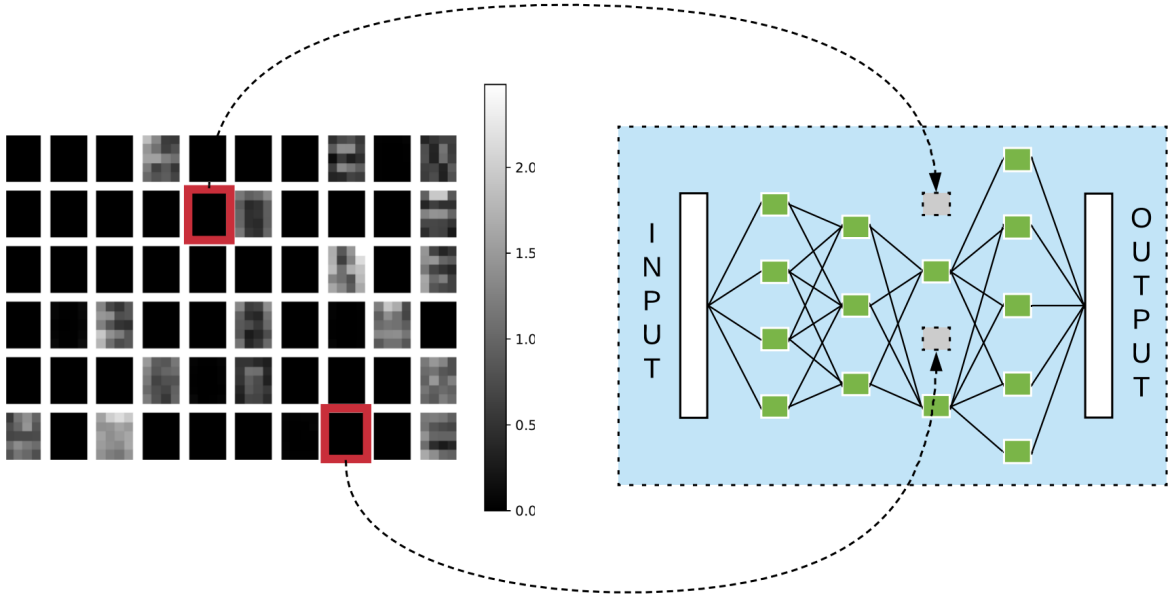
- **高效性**
  - 尽量少的使用干净数据
  - 尽量快的移除所有后门
- **代价小**
  - 不影响模型正常性能
  - 移除后不需要重训练
- **通用性**
  - 能同时移除不同类型的后门
  - 能保护不同类型的模型、数据集等
- **鲁棒性**
  - 对Adaptive Attack鲁棒
  - 对不同设置鲁棒

后门移除 – 实验设置

- 假设后门模型已经确定
- 大部分方法需要少量干净数据
- 大部分方法需要在完成后对模型再次微调
- 多少都会影响一点模型性能

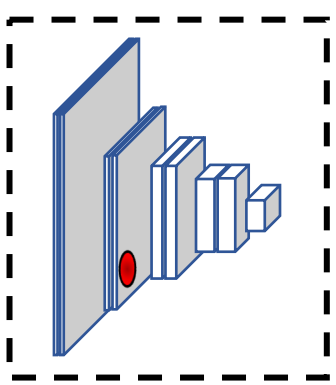
Fine-pruning (FP)

- **步骤1：参见冗余神经元；步骤2：微调以恢复性能** = 裁剪+微调



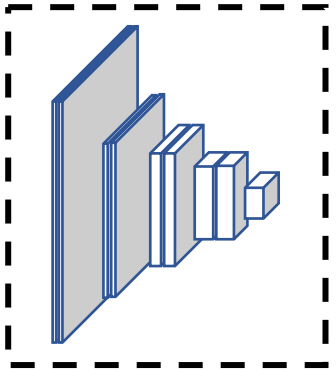
- 裁剪在干净数据上休眠的神经元

Mode Connectivity Repair (MCR)



$$\phi_\theta(t) = (1-t)^2\omega_1 + 2t(1-t)\theta + t^2\omega_2, \quad 0 \le t \le 1$$

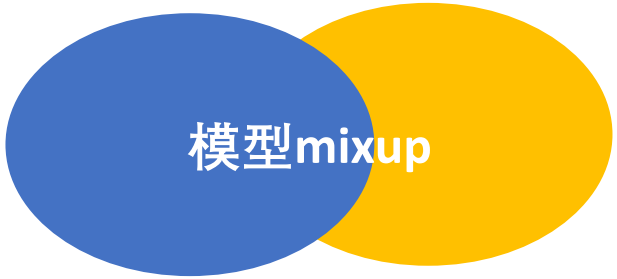
$\phi(t)$



w_1 → w_2

$\phi(0) = w_1$

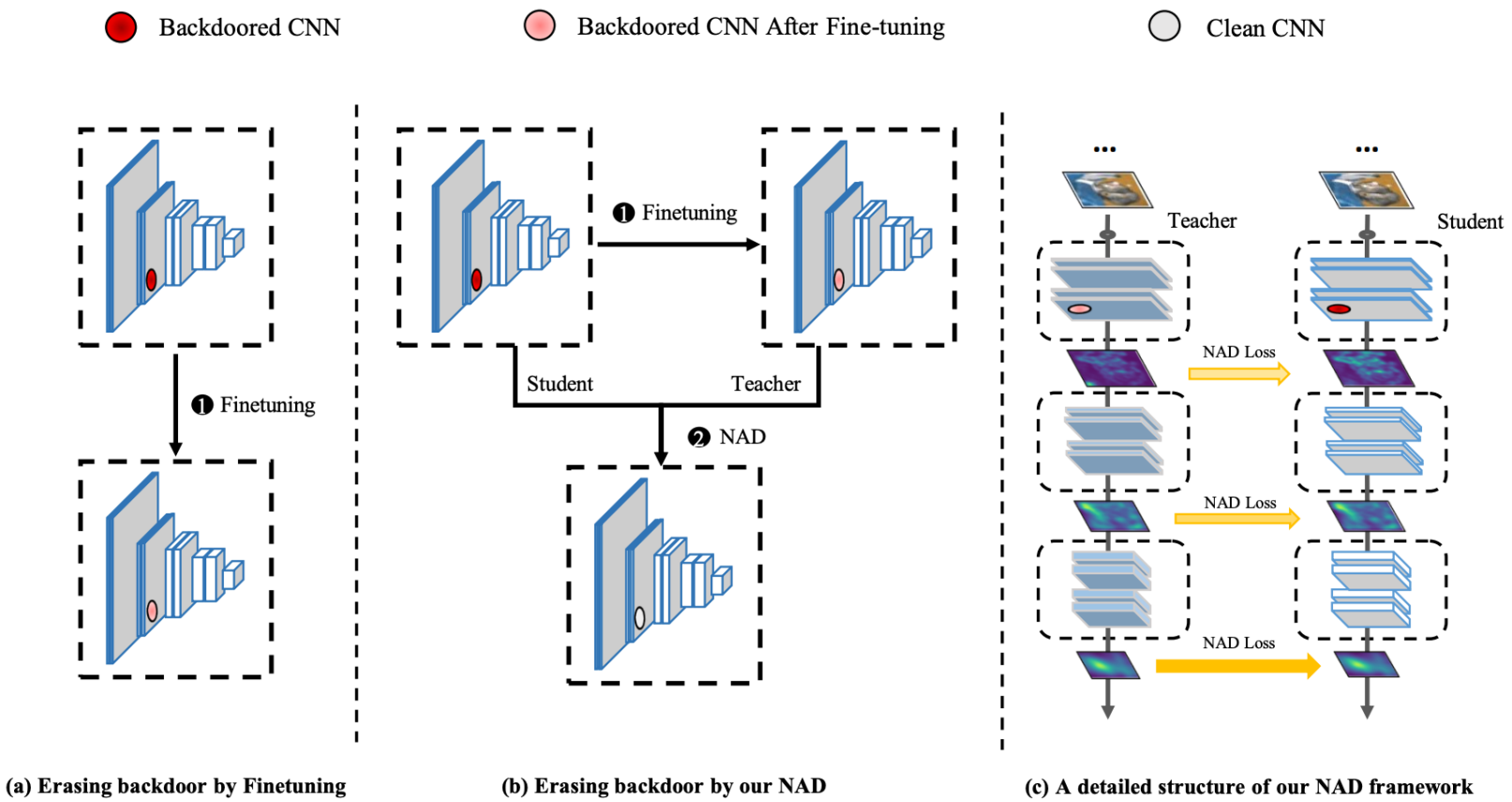
$\phi(1) = w_2$



在两个模型之间寻找一个更好的过渡

后门模型 → 少量干净数据训练的模型

Neural Attention Distillation (NAD)



(a) Erasing backdoor by Finetuning (b) Erasing backdoor by our NAD (c) A detailed structure of our NAD framework

- 使用知识蒸馏移除后门
  - ✓ 需要一小部分干净数据
  - ✓ 先微调
  - ✓ 用微调后的模型作为教师
  - ✓ 对中间层通道平均激活进行对齐蒸馏

Li, Yige, et al. “Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks.” *ICLR*, 2020.

Neural Attention Distillation (NAD)

- **Definition of attention functions**

$$\mathcal{A}_{\text{sum}}(F^l) = \sum_{i=1}^{C} \left|F_i^l\right|; \quad \mathcal{A}^p_{\text{sum}}(F^l) = \sum_{i=1}^{C} \left|F_i^l\right|^p; \quad \mathcal{A}^p_{\text{mean}}(F^l) = \frac{1}{C}\sum_{i=1}^{C} \left|F_i^l\right|^p$$

- **Definition of attention distillation loss**

$$\mathcal{L}_{\text{NAD}}\left(F_T^l, F_S^l\right) = \left\| \frac{\mathcal{A}(F_T^l)}{\left\|\mathcal{A}(F_T^l)\right\|_2} - \frac{\mathcal{A}(F_S^l)}{\left\|\mathcal{A}(F_S^l)\right\|_2} \right\|_2$$

- **Overall training loss**

$$\mathcal{L}_{total} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}_{\text{CE}}(F_S(\boldsymbol{x}), y) + \beta \cdot \sum_{l=1}^{K} \mathcal{L}_{\text{NAD}}(F_T^l(\boldsymbol{x}), F_S^l(\boldsymbol{x}))]$$

Li, Yige, et al. “Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks.” *ICLR*, 2020.

Neural Attention Distillation (NAD)

■ Performance of NAD

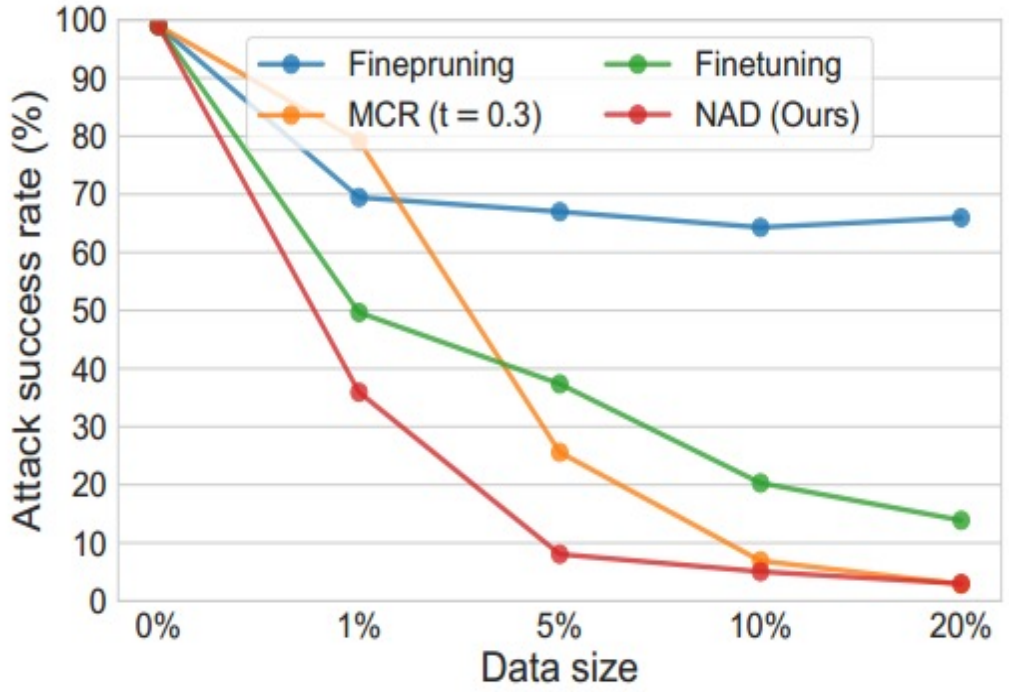
Backdoor Attack	Before		Finetuning		Fine-pruning		MCR (t = 0.3)		NAD (Ours)	
	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
BadNets	100	85.65	17.18	**81.22**	99.73	81.14	**4.65**	80.94	4.77	81.17
Trojan	100	81.24	71.76	77.88	41.00	78.17	41.25	78.76	**19.63**	**79.16**
Blend	99.97	84.95	36.60	81.22	93.62	81.13	64.33	80.34	**4.04**	**81.68**
CL	99.21	82.43	75.08	**81.73**	29.88	79.32	32.95	79.04	**9.18**	80.34
SIG	99.91	84.36	9.18	81.28	74.26	81.60	**1.62**	80.94	2.52	**81.95**
Refool	95.16	82.38	14.38	80.34	63.49	80.64	8.76	78.84	**3.18**	**80.73**
Average	99.04	83.50	37.36	80.61	67.00	80.50	25.59	79.81	**7.22**	**80.83**
Deviation	-	-	↓ 61.68	↓ 2.89	↓ 32.04	↓ 3	↓ 73.44	↓ 3.69	↓ **91.82**	↓ **2.66**

- ❑ **The most effective against all 6 backdoor attacks.**
- ❑ **Minimum impact on clean accuracy.**
- ❑ **Reducing ASR by 91.82% on average, by sacrificing only 2.66% accuracy.**

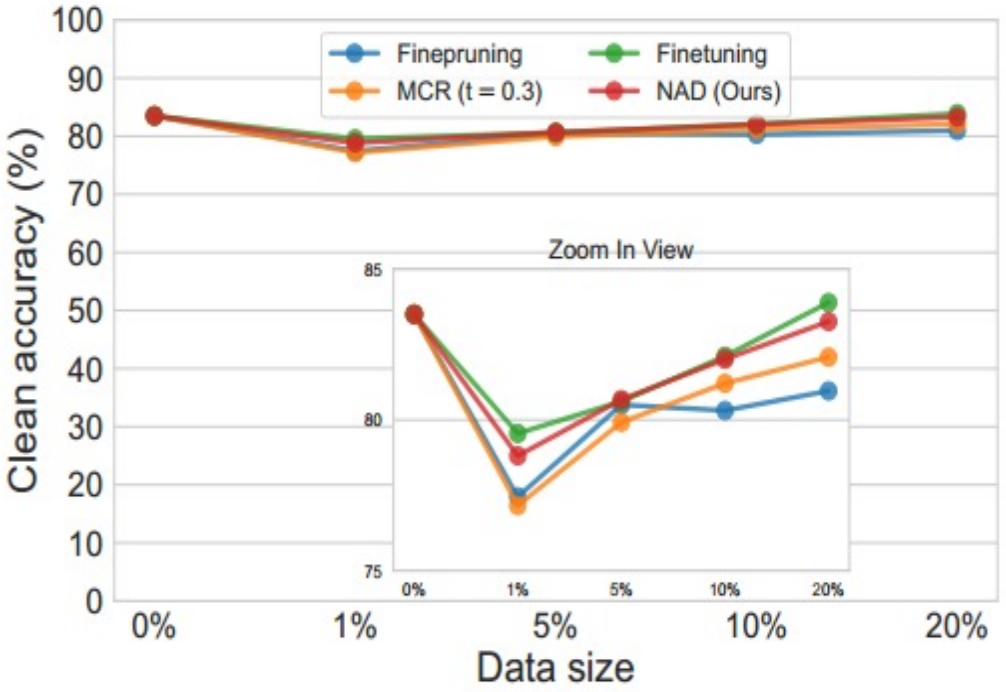
Li, Yige, et al. “Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks.” *ICLR*, 2020.

Neural Attention Distillation (NAD)

■ **Performance under different % of clean data:**



(a) Attack success rate



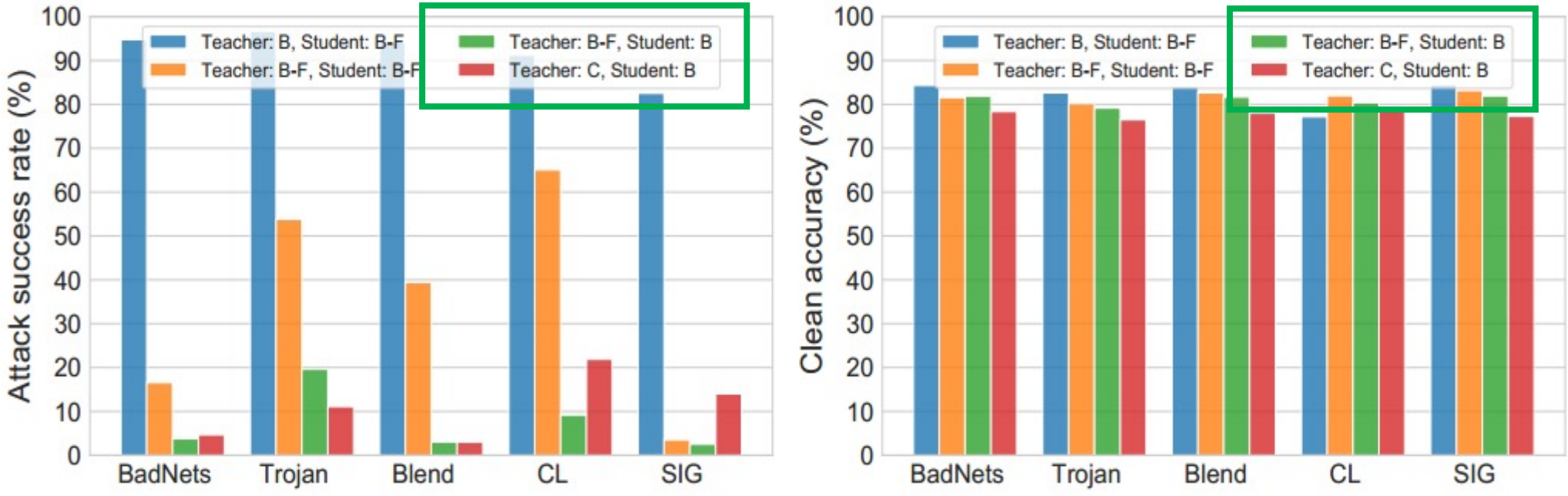
(b) Clean accuracy

❑ **5% looks good, 10% is sufficient, the more the better!**

Li, Yige, et al. "Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks." *ICLR,* 2020.

Neural Attention Distillation (NAD)

- **Effectiveness of different teacher-student combinations:**



(a) Attack success rate

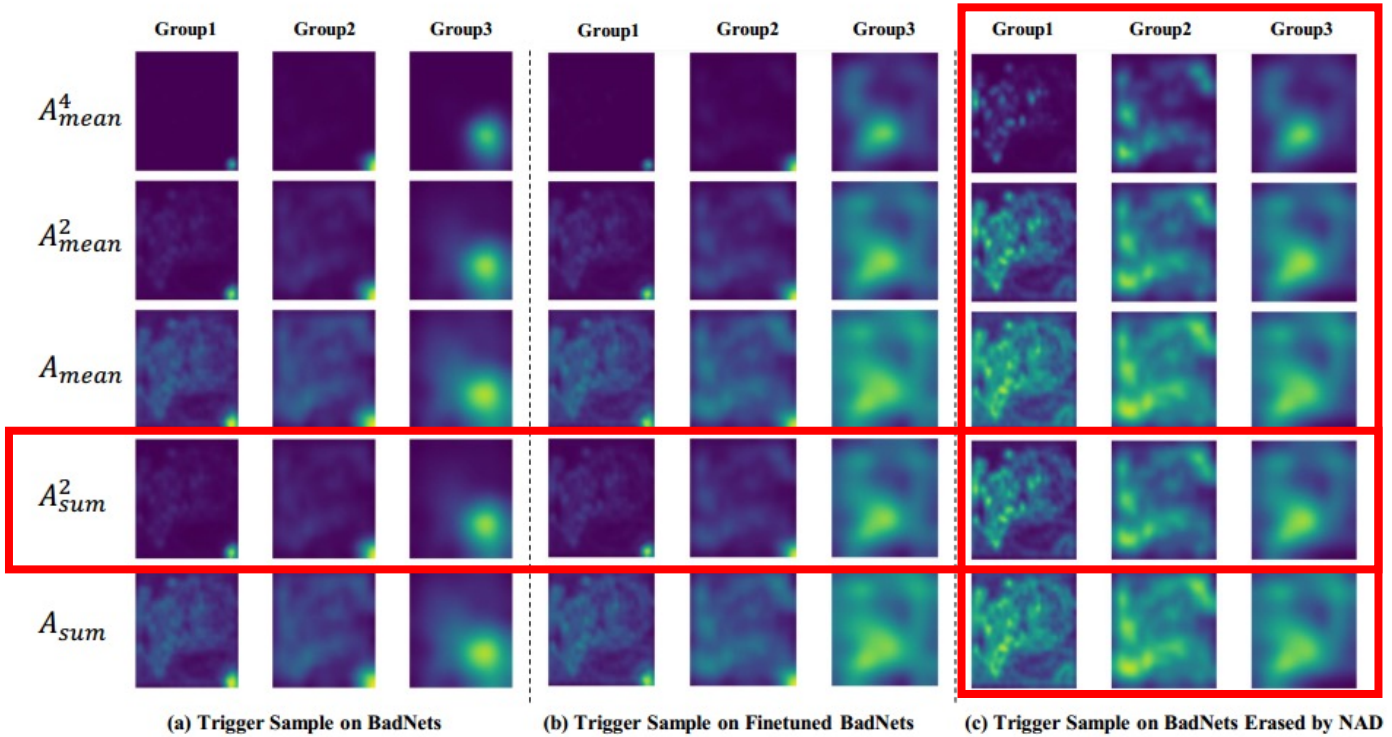
(b) Clean accuracy

- **Fine-tuned teacher (B-F) is better, on average.**
- **Teacher trained on clean subset is also good but sacrifices more clean accuracy.**

Neural Attention Distillation (NAD)

■ Effects of different attention functions:

A backdoored image



Note: The **red boxes** highlight our recommended attention function.

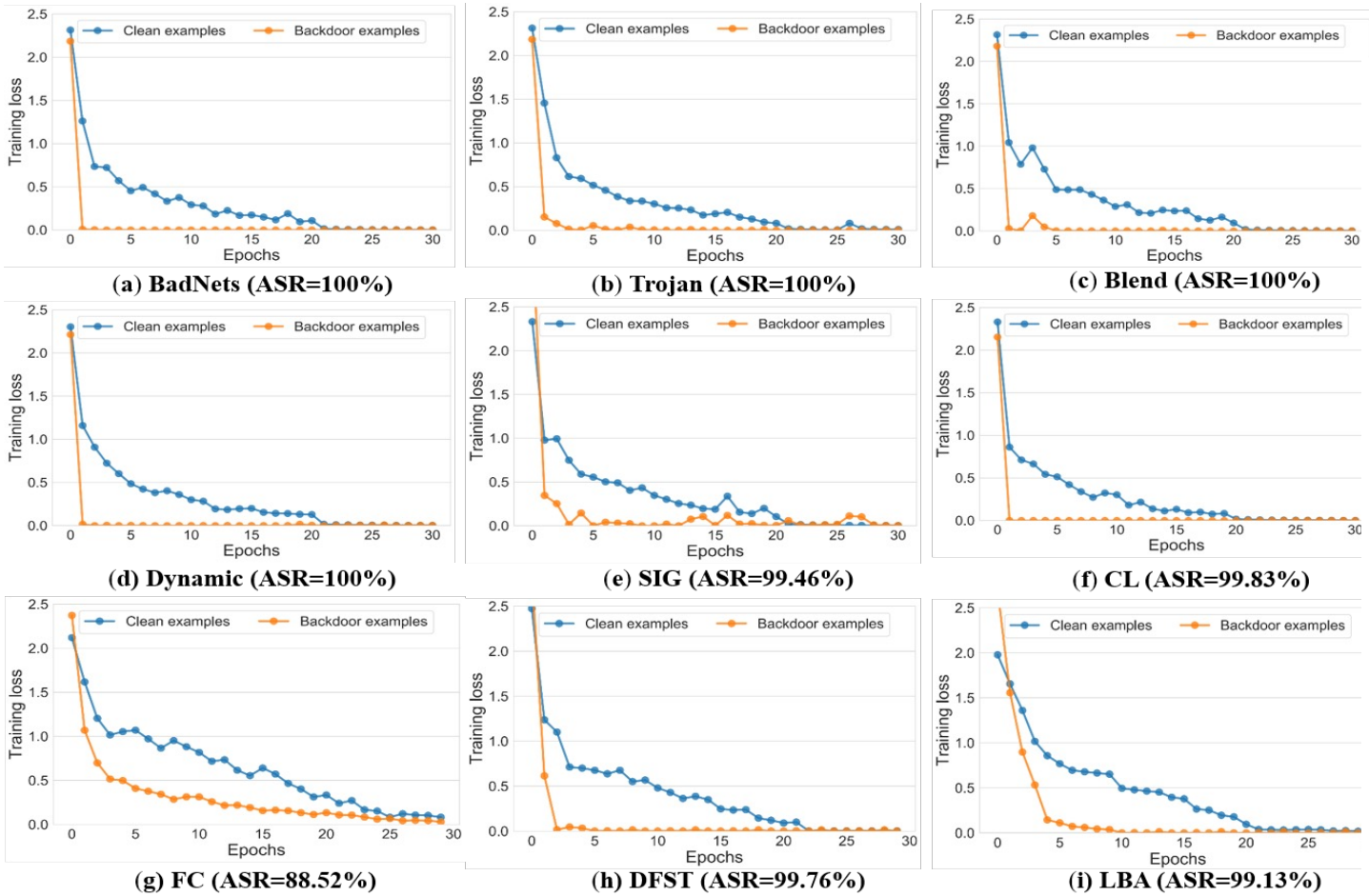
Li, Yige, et al. “Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks.” *ICLR*, 2020.

Anti-Backdoor Learning (ABL)

- 反后门学习：**如何在毒化数据上训练一个干净无后门的模型？**

找到后门投毒样本的共性 → 在训练阶段检测并抛弃有毒样本

Anti-Backdoor Learning (ABL)



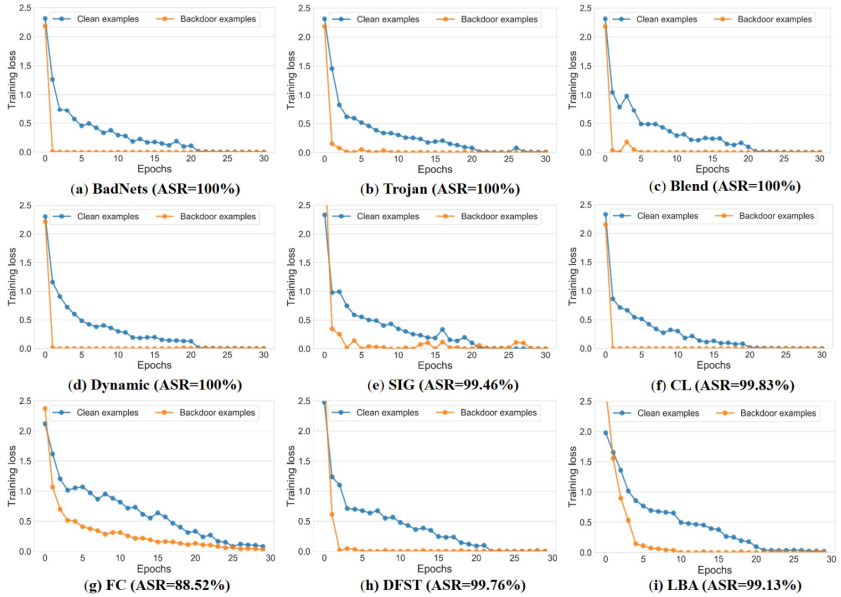
Training loss on Clean examples (blue) VS. Backdoored examples (yellow)

- **Weaknesses of backdoor attacks:**
  - 1. The backdoor task is much easier than the clean task. (Weakness 1)
  - 2. A backdoor attack enforces an explicit correlation between the trigger and the target class to simplify and accelerate the injection of the backdoor trigger. (Weakness 2)

Li, Yige, et al. “Anti-backdoor learning: Training clean models on poisoned data.” *NeurIPS, 2021*.

Anti-Backdoor Learning (ABL)

- 反后门学习：**如何在毒化数据上训练一个干净无后门的模型？**



后门样本学的更快

- 步骤1：局部梯度上升：控制Loss不要太高
- 步骤2：后门样本检测：loss低的是后门样本
- **步骤3：全局梯度上升：在后门样本上做反学习**

$$\mathcal{L}_{\mathrm{ABL}}^{t}=\begin{cases}\mathcal{L}_{\mathrm{LGA}}=\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\operatorname{sign}(\ell(f_\theta(\boldsymbol{x}),y)-\gamma)\cdot\ell(f_\theta(\boldsymbol{x}),y)\right] & \text{if } 0\le t<T_{te}\\ \mathcal{L}_{\mathrm{GGA}}=\mathbb{E}_{(\boldsymbol{x},y)\sim\widehat{\mathcal{D}}_c}\left[\ell(f_\theta(\boldsymbol{x}),y)\right]-\mathbb{E}_{(\boldsymbol{x},y)\sim\widehat{\mathcal{D}}_b}\left[\ell(f_\theta(\boldsymbol{x}),y)\right] & \text{if } T_{te}\le t<T,\end{cases}$$

ABL学习机制

Anti-Backdoor Learning (ABL)

- Problem Formulation

$$\mathcal{L} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(f_\theta(\boldsymbol{x}),y)] = \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_c}[\ell(f_\theta(\boldsymbol{x}),y)]}_{\text{clean task}} + \underbrace{\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\ell(f_\theta(\boldsymbol{x}),y)]}_{\text{backdoor task}},$$

- Overview of ABL
  - Stage 1: **Backdoor Isolation**; $(0 \le t < T_{te})$, t: current epoch; T_{te}: turning epoch
  - Stage 2: **Backdoor Unlearning**. $(T_{te} \le t < T)$ T: total epoch

$$\mathcal{L}_{\text{ABL}}^t = \begin{cases} \mathcal{L}_{\text{LGA}} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\text{sign}(\ell(f_\theta(\boldsymbol{x}),y)-\gamma)\cdot\ell(f_\theta(\boldsymbol{x}),y)\right] & \text{if } 0 \le t < T_{te} \\ \mathcal{L}_{\text{GGA}} = \mathbb{E}_{(\boldsymbol{x},y)\sim\widehat{\mathcal{D}}_c}\left[\ell(f_\theta(\boldsymbol{x}),y)\right] - \mathbb{E}_{(\boldsymbol{x},y)\sim\widehat{\mathcal{D}}_b}\left[\ell(f_\theta(\boldsymbol{x}),y)\right] & \text{if } T_{te} \le t < T, \end{cases}$$

LGA: local gradient ascent; GGA: global gradient ascent

Li, Yige, et al. “Anti-backdoor learning: Training clean models on poisoned data.” *NeurIPS, 2021*.

Anti-Backdoor Learning (ABL)

- Performance of ABL:

Dataset	Types	No Defense		FP		MCR		NAD		ABL (Ours)	
		ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA
CIFAR-10	*None*	0%	89.12%	0%	85.14%	0%	87.49%	0%	88.18%	0%	**88.41%**
	BadNets	100%	85.43%	99.98%	82.14%	3.32%	78.49%	3.56%	82.18%	**3.04%**	**86.11%**
	Trojan	100%	82.14%	66.93%	80.17%	23.88%	76.47%	18.16%	80.23%	**3.81%**	**87.46%**
	Blend	100%	84.51%	85.62%	81.33%	31.85%	76.53%	**4.56%**	82.04%	16.23%	**84.06%**
	Dynamic	100%	83.88%	87.18%	80.37%	26.86%	70.36%	22.50%	74.95%	**18.46%**	**85.34%**
	SIG	99.46%	84.16%	76.32%	81.12%	0.14%	78.65%	1.92%	82.01%	**0.09%**	**88.27%**
	CL	99.83%	83.43%	54.95%	81.53%	19.86%	77.36%	16.11%	80.73%	**0%**	**89.03%**
	FC	88.52%	83.32%	69.89%	80.51%	44.43%	77.57%	58.68%	81.23%	**0.08%**	**82.36%**
	DFST	99.76%	82.50%	78.11%	80.23%	39.22%	75.34%	35.21%	78.40%	**5.33%**	**79.78%**
	LBA	99.13%	81.37%	54.43%	79.67%	15.52%	78.51%	10.16%	79.52%	**0.06%**	**80.52%**
	CBA	90.63%	84.72%	77.33%	79.15%	38.76%	76.36%	33.11%	82.40%	**29.81%**	**84.66%**
	Average	97.73%	83.55%	75.07%	80.62%	24.38%	76.56%	20.40%	80.37%	**7.69%**	**84.76%**
GTSRB	*None*	0%	97.87%	0%	90.14%	0%	95.49%	0%	95.18%	0%	**96.41%**
	BadNets	100%	97.38%	99.57%	88.61%	1.00%	93.45%	0.19%	89.52%	**0.03%**	**96.01%**
	Trojan	99.80%	96.27%	93.54%	84.22%	2.76%	92.98%	0.37%	90.02%	**0.36%**	**94.95%**
	Blend	100%	95.97%	99.50%	86.67%	**6.83%**	92.91%	8.10%	89.37%	24.59%	**93.14%**
	Dynamic	100%	97.27%	99.84%	88.38%	64.82%	43.91%	68.71%	76.93%	**6.24%**	**95.80%**
	SIG	97.13%	97.13%	79.28%	90.50%	33.98%	91.83%	**4.64%**	89.36%	5.13%	**96.33%**
	Average	99.38%	96.80%	94.35%	87.68%	21.88%	83.01%	19.17%	87.04%	**7.27%**	**95.25%**
ImageNet Subset	*None*	0%	89.93%	0%	83.14%	0%	85.49%	0%	88.18%	0%	**88.31%**
	BadNets	100%	84.41%	97.70%	82.81%	28.59%	78.52%	6.32%	81.26%	**0.94%**	**87.76%**
	Trojan	100%	85.56%	96.39%	80.34%	6.67%	76.87%	15.48%	80.52%	**1.47%**	**88.19%**
	Blend	99.93%	86.15%	99.34%	81.33%	**19.23%**	75.83%	26.47%	82.39%	21.42%	**85.12%**
	SIG	98.60%	86.02%	78.82%	85.72%	25.14%	78.87%	5.15%	83.01%	**0.18%**	**86.42%**
	Average	99.63%	85.53%	93.06%	82.55%	19.91%	77.52%	13.35%	81.80%	**6.00%**	**86.87%**

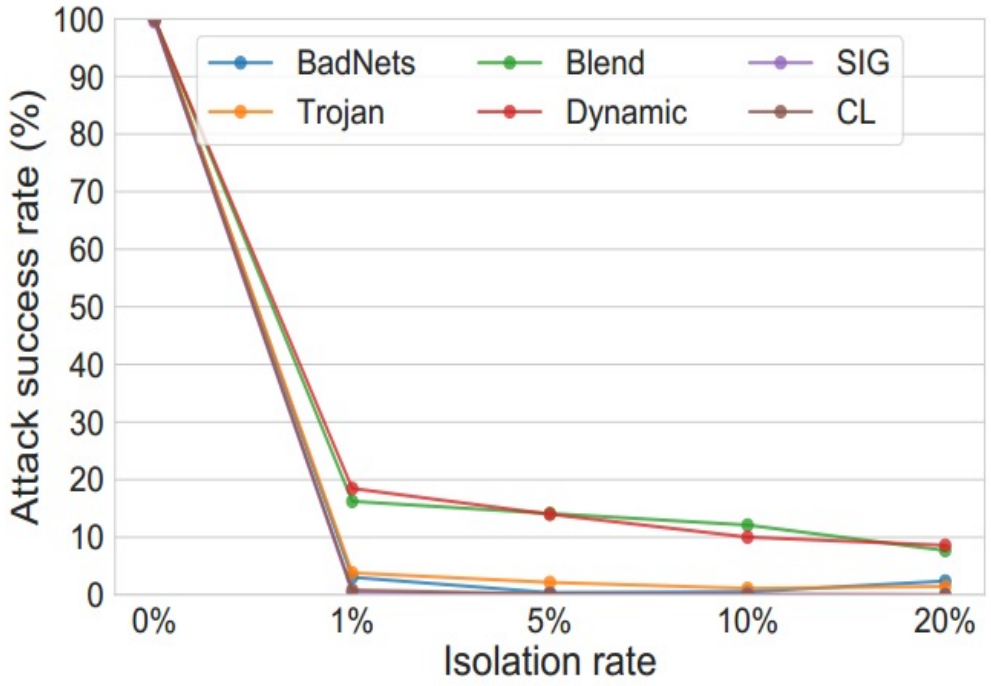
Conclusions:

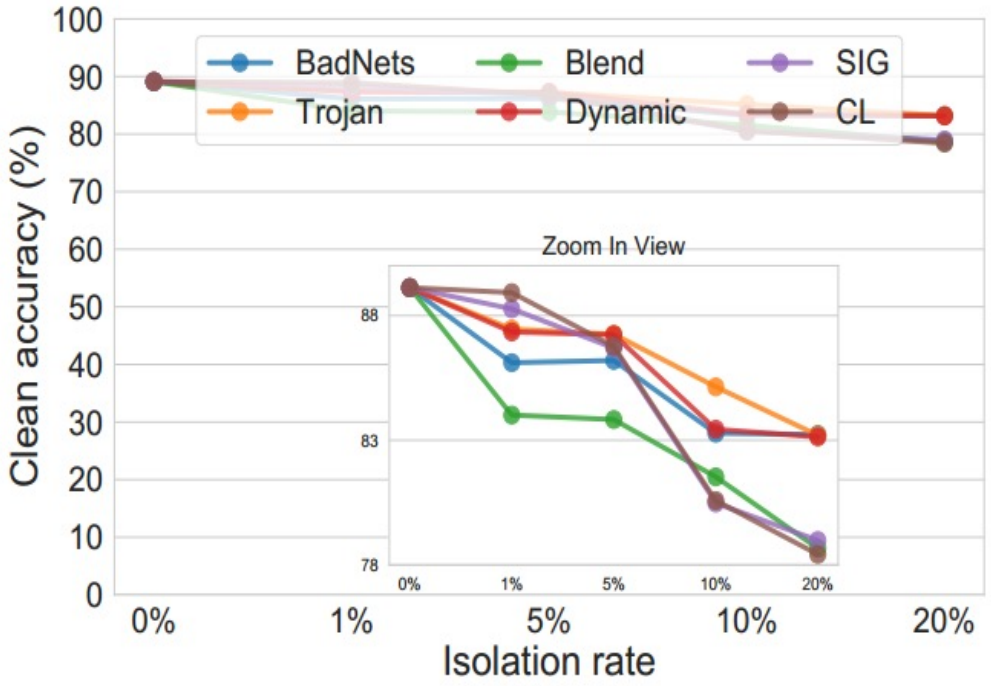
- **The most effective defense against all 10 backdoor attacks;**
- **Minimum impact on clean accuracy.**

Li, Yige, et al. “Anti-backdoor learning: Training clean models on poisoned data.” *NeurIPS, 2021*.

Anti-Backdoor Learning (ABL)

- Performance of our ABL with different isolation rates on CIFAR-10 dataset:

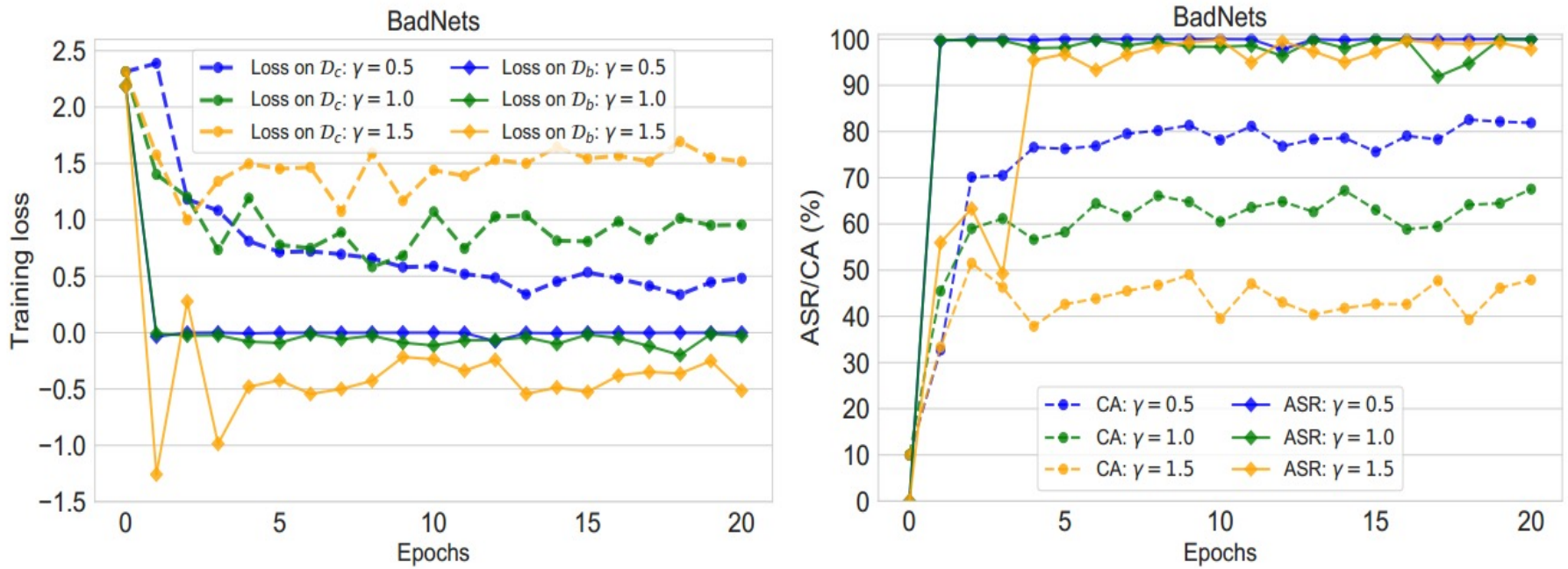




- **1%** isolation achieves a good trade-off between ASR and CA!

Anti-Backdoor Learning (ABL)

- Performance of our ABL with different γ on CIFAR-10 against BadNets:



- The larger γ, the better separation effect !（同色的两条线相隔越远）

Li, Yige, et al. “Anti-backdoor learning: Training clean models on poisoned data.” *NeurIPS, 2021*.

Anti-Backdoor Learning (ABL)

- Performance of our ABL under different turning epochs on CIFAR-10:

Tuning Epoch	BadNets		Trojan		Blend		Dynamic	
	ASR	CA	ASR	CA	ASR	CA	ASR	CA
10	**1.12%**	85.30%	5.04%	85.12%	16.34%	**84.22%**	25.33%	84.12%
20	3.04%	**86.11%**	**3.66%**	**87.46%**	**16.23%**	84.06%	**18.46%**	**85.34%**
30	3.22%	85.60%	3.81%	87.25%	19.87%	83.83%	20.56%	85.23%
40	4.05%	84.28%	4.96%	85.14%	18.78%	81.53%	19.15%	83.44

  - **Epoch 20** achieves the best overall results.

- Stress testing of our ABL on CIFAR-10:

Poisoning Rate	Defense	BadNets		Trojan		Blend		Dynamic	
		ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
50%	*None*	100%	75.31%	100%	70.44%	100%	69.49%	100%	66.15%
	ABL	4.98%	70.52%	16.11%	68.56%	27.28%	64.19%	25.74%	61.32%
70%	*None*	100%	74.8%	100%	69.46%	100%	67.32%	100%	66.15%
	ABL	5.02%	70.11%	29.29%	68.79%	62.28%	64.43%	69.36%	62.09%

  - ABL with only **1% isolation** remains effective against up to 1) **70%** BadNets; and 2) **50%** Trojan, Blend, and Dynamic.

Li, Yige, et al. “Anti-backdoor learning: Training clean models on poisoned data.” *NeurIPS, 2021*.

Anti-Backdoor Learning (ABL)

- Performance of various unlearning methods against BadNets attack on CIFAR-10:

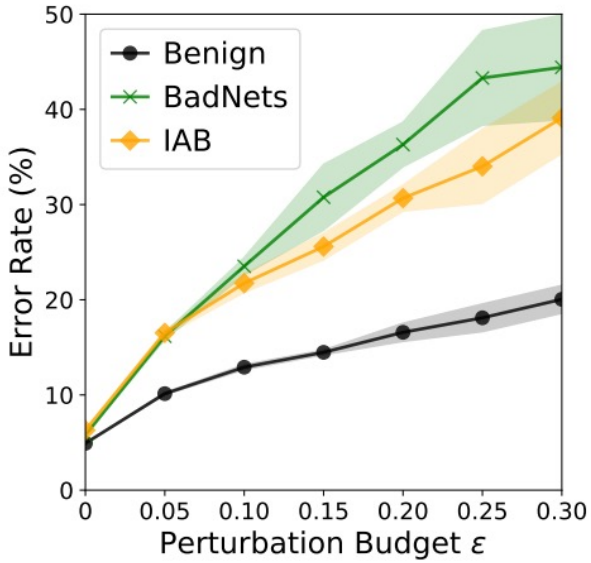
Backdoor Unlearning Methods	Method Type	Discard $\widehat{\mathcal{D}}_b$	Backdoored		After Unlearning	
			ASR	CA	ASR	CA
Pixel Noise	Image-based	No	100%	85.43%	57.54%	82.33%
Grad Noise	Image-based	No	100%	85.43%	**47.65%**	**82.62%**
Label Shuffling	Label-based	No	100%	85.43%	30.23%	83.76%
Label Uniform	Label-based	No	100%	85.43%	75.12%	83.47%
Label Smoothing	Label-based	No	100%	85.43%	99.80%	83.17%
Self-Learning	Label-based	No	100%	85.43%	**21.26%**	**84.38%**
Fine-tuning All Layers	Model-based	Yes	100%	85.43%	99.12%	83.64%
Fine-tuning Last Layers	Model-based	Yes	100%	85.43%	22.33%	77.65%
Fine-tuning ImageNet Model	Model-based	Yes	100%	85.43%	12.18%	75.10%
Re-training from Scratch	Model-based	Yes	100%	85.43%	11.21%	86.02%
ABL	Model-based	No	100%	85.43%	**3.04%**	**86.11%**

- ABL achieves the best unlearning performance of **ASR 3.04% and CA 86.11%,** followed by (discard isolated data then) **Re-training from scratch**!

Li, Yige, et al. “Anti-backdoor learning: Training clean models on poisoned data.” *NeurIPS, 2021*.

Adversarial Neuron Pruning (ANP)

- 后门模型对对抗扰动更敏感



- 从后门模型中发现并移除敏感神经元

$$\min_{\mathbf{m}\in[0,1]^n} \left[\alpha\mathcal{L}_{\mathcal{D}_\mathcal{V}}(\mathbf{m}\odot\mathbf{w},\mathbf{b}) + (1-\alpha)\max_{\boldsymbol{\delta},\boldsymbol{\xi}\in[-\epsilon,\epsilon]^n}\mathcal{L}_{\mathcal{D}_\mathcal{V}}((\mathbf{m}+\boldsymbol{\delta})\odot\mathbf{w},(1+\boldsymbol{\xi})\odot\mathbf{b})\right]$$

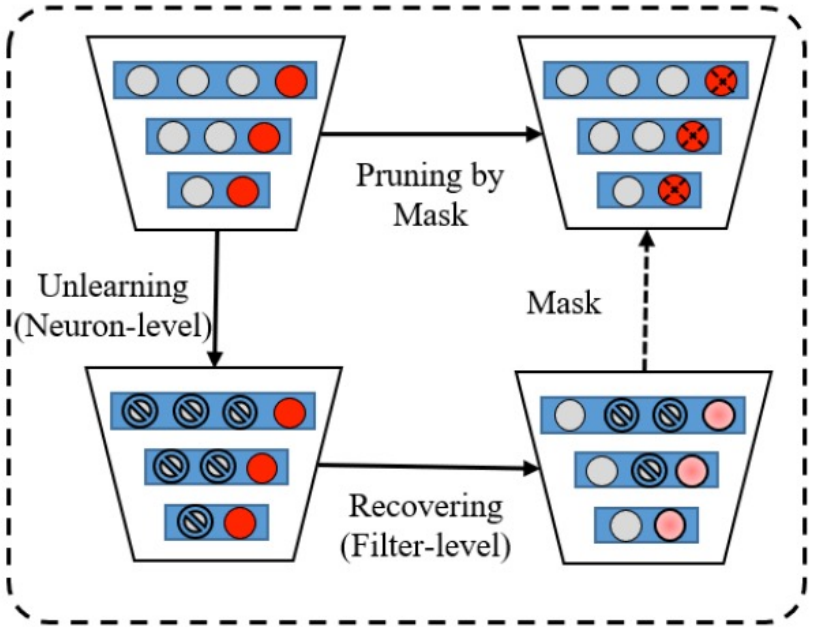
m：神经元掩码

对抗扰动

Wu and Wang. “Adversarial neuron pruning purifies backdoored deep models.” *NeurIPS, 2021*.

Reconstructive Neural Pruning

提出**重构式神经元剪枝方法**，通过一个**“反学习-再恢复”**过程来定位后门神经元，以完成精准剪枝



- **（1）反学习**：暴露后门相关神经元

$$\max_{\theta} \mathbb{E}_{(\boldsymbol{x}_d,\, y_d)\in\mathcal{D}_d} \mathcal{L}(f\,(\boldsymbol{x}_d,\, y_d; \theta))$$

- **（2）再恢复**：恢复干净神经元

$$\min_{\mathbf{m}^{\kappa}\in[0,1]^n} \mathbb{E}_{(\boldsymbol{x}_d,/y_d)\in\mathcal{D}_d} \mathcal{L}(f(\boldsymbol{x}_d,\, y_d; \mathbf{m}^{\kappa} \odot \hat{\theta}))$$

步骤（1）和（2）构成：
非对称的“反学习-再恢复” 机制

裁剪28-155个神经元即可移除后门

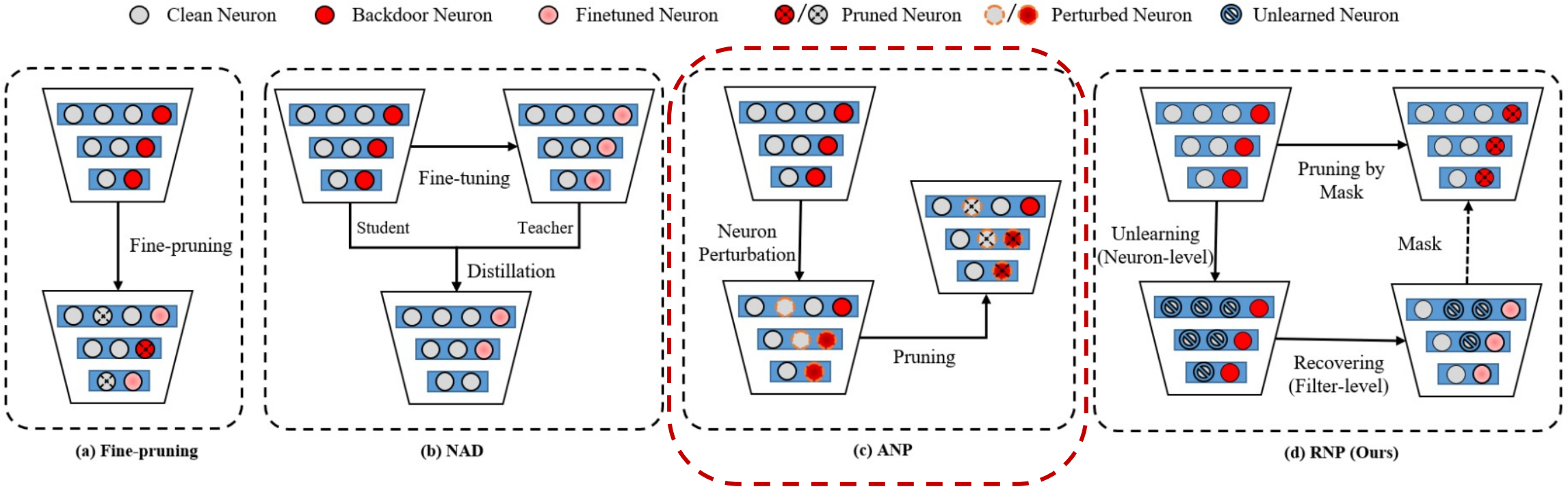
- **（3）裁剪**：根据m^k裁剪**后门神经元**

Yige Li, et al. Reconstructive Neuron Pruning for Backdoor Defense, ICML, 2023

Reconstructive Neural Pruning

■ **RNP与其他防御框架对比**

ANP无法有效
揭露**特征空间**后门



本文提出重构神经元剪枝（RNP）方法，通过一个“**反学习-再恢复**”过程来定位后门神经元，以完成**精准剪枝（无需额外微调）**

Reconstructive Neural Pruning

- **RNP包含三个步骤：**

- **（1）反学习**：暴露后门相关神经元

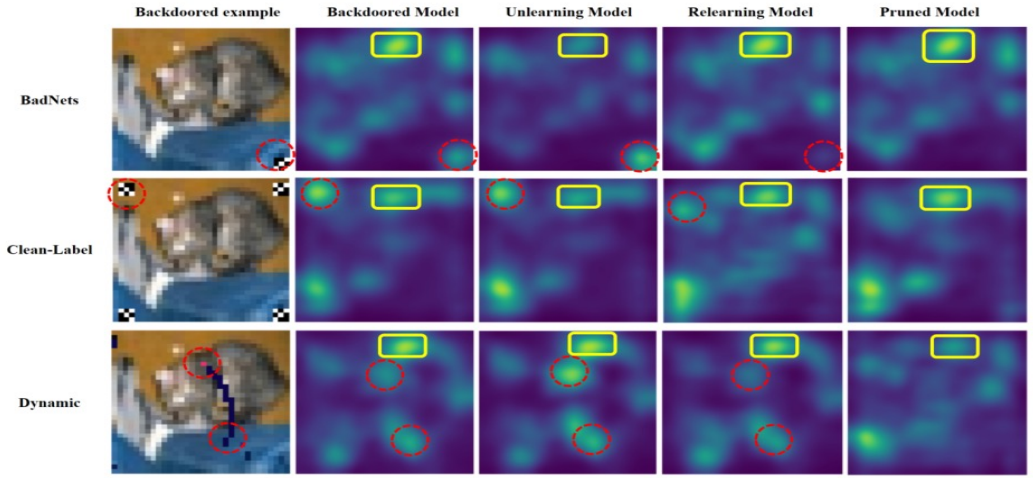
$$\max_{\theta} \mathbb{E}_{(\boldsymbol{x}_d,\, y_d)\in\mathcal{D}_d}\mathcal{L}(f\,(\boldsymbol{x}_d,\, y_d;\theta))$$

- **（2）再恢复**：恢复干净神经元

$$\min_{\mathbf{m}^\kappa\in[0,1]^n} \mathbb{E}_{(\boldsymbol{x}_d,/y_d)\in\mathcal{D}_d}\mathcal{L}(f(\boldsymbol{x}_d,\, y_d;\mathbf{m}^\kappa\odot\hat{\theta}))$$

步骤（1）和（2）构成：
非对称的“反学习-再恢复” 机制

- **（3）裁剪**：根据m^k裁剪**后门相关神经元**



反学习-再恢复机制特征可视化

Reconstructive Neural Pruning

- **RNP与其他4种防御方法的对比结果**

Datasets	Backdoor Attacks	No Defense		FP		NAD		I-BAU		ANP		RNP (Ours)	
		ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA
CIFAR-10	BadNets	100.00	93.40	15.11	88.15	1.20	90.64	15.50	91.18	0.53	91.61	**0.20**	**92.22**
	Trojan	99.90	93.15	56.51	85.43	5.68	88.72	12.78	90.46	**1.00**	92.37	2.23	**92.56**
	Blend	100.00	93.10	68.22	85.21	10.92	89.77	1.62	90.16	0.50	92.31	**0.33**	**92.62**
	CL	100.00	94.84	24.38	88.75	13.17	89.76	23.12	88.82	15.20	92.86	**8.87**	**93.12**
	SIG	90.86	94.59	19.16	87.88	0.64	89.40	29.32	89.67	1.19	92.97	**0.43**	**94.62**
	Dynamic	99.97	91.36	41.73	83.28	13.60	88.64	18.74	86.87	**9.20**	89.66	15.24	**90.18**
	WaNet	99.10	93.67	68.92	86.35	17.46	82.41	23.18	87.38	13.14	92.64	**10.98**	**92.83**
	FC	100.00	94.67	98.45	87.62	36.07	88.02	17.93	86.75	74.75	81.97	**1.80**	**90.93**
	DFST	100.00	94.52	88.78	85.32	12.70	88.72	25.58	87.44	10.80	90.66	**4.61**	**92.78**
	AWP	94.39	94.30	23.17	86.04	1.71	89.13	8.71	89.62	**0.67**	92.64	1.04	**94.24**
	LIRA	100.00	92.71	87.78	83.12	32.12	86.73	51.33	82.56	20.25	87.78	**13.51**	**92.26**
	A-Blend	71.86	92.16	85.22	81.82	18.51	85.23	33.38	85.91	23.71	90.16	**1.09**	**90.38**
	Average	96.34	93.54	56.45	85.75	13.65	88.10	21.77	88.07	14.25	90.64	**5.03**	**92.18**
ImageNet-12	BadNets	100.00	88.53	91.70	83.23	9.12	83.26	15.38	85.15	10.25	85.21	**5.80**	**85.83**
	Trojan	100.00	89.79	93.69	81.40	12.31	82.52	19.61	84.11	7.48	87.41	**0.59**	**89.30**
	Blend	99.90	89.44	92.14	82.13	28.76	82.93	9.34	82.27	6.21	86.40	**5.54**	**86.89**
	SIG	73.78	88.18	87.82	81.27	21.15	83.31	29.23	81.57	25.53	52.52	**15.20**	**84.15**
	FC	95.77	88.95	90.52	79.36	31.43	81.56	38.51	79.33	42.69	53.01	**17.23**	**83.36**
	Average	93.89	88.98	91.17	81.48	20.55	82.72	22.41	82.49	18.43	72.91	**8.87**	**85.91**

- RNP优于当前最先进的方法ANP（CIFAR-10上抵御9/12攻击和ImageNet子集的抵御5/5攻击）；
- 后门攻击平均攻击成功率由93.89%降低至8.87%，仅损失2.66%的平均准确率

Reconstructive Neural Pruning

- ANP裁剪效果：
  - 通过“**参数对抗扰动**” 揭示后门相关神经元

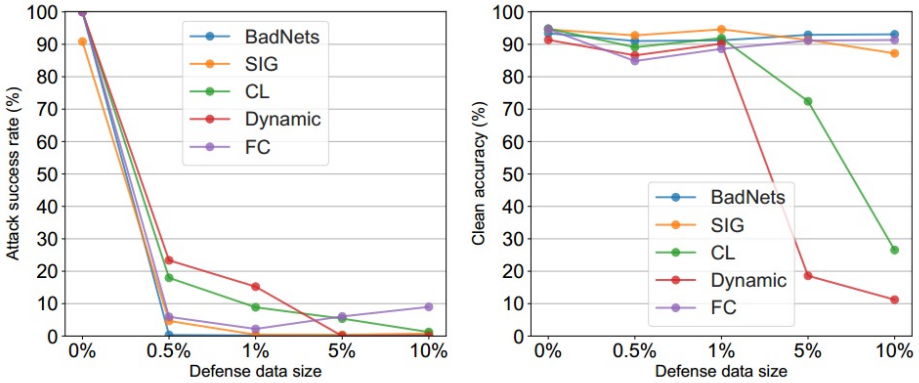
ANP vs. RNP

- RNP裁剪效果：
  - 通过“**反学习-再回复**”机制暴露后门相关神经元；
  - 裁剪更少的神经元，同时维持高准确率（CA）和低攻击成功率（ASR）

Defense	Metric	BadNets	Trojan	Blend	CL	SIG	Dynamic	WaNet	FC	DFST	AWP	LIRA	A-Blend
ANP	$Neurons\downarrow$ (on All)	94	96	42	135	88	69	126	199	165	56	158	96
	ASR (%)	0.53	1.00	0.50	15.20	1.19	9.20	13.14	74.75	10.80	0.67	20.25	23.71
	CA (%)	91.61	92.37	92.31	92.86	92.97	89.66	92.64	81.97	90.66	92.64	87.78	90.16
RNP	$Neurons\downarrow$ (on All)	**41**	**48**	**28**	**103**	**73**	**59**	**92**	**155**	**83**	**40**	**112**	**78**
	ASR (%)	0.20	2.23	0.33	8.87	0.43	15.24	10.98	1.80	4.61	1.04	13.51	1.09
	CA (%)	92.22	92.56	92.62	93.12	94.62	90.18	92.83	90.93	92.78	94.24	92.96	90.38

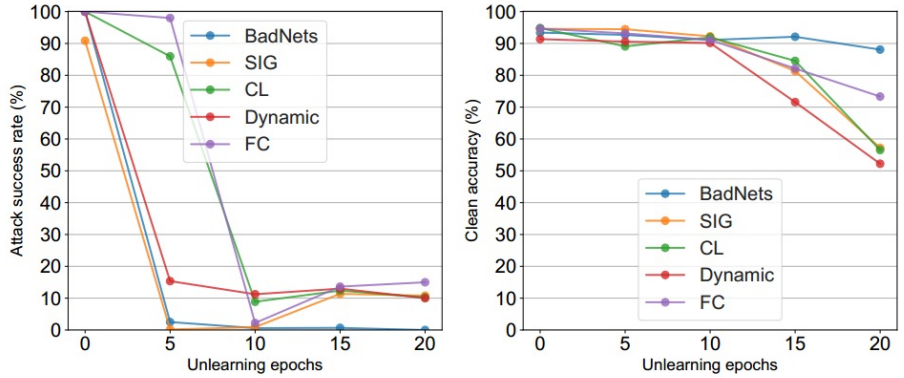
Reconstructive Neural Pruning

- **RNP在不同规模防御数据下的结果**



针对大多数攻击，RNP 在不同数据规模上具备一致的防御效果！

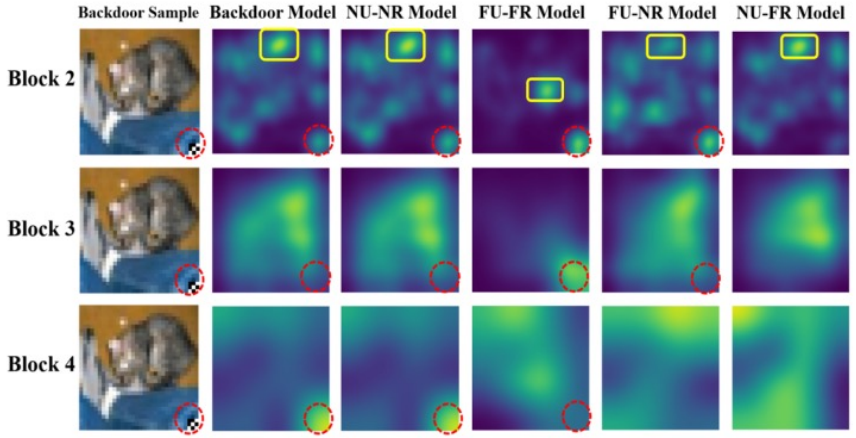
- **RNP在不同反学习轮次下的结果**

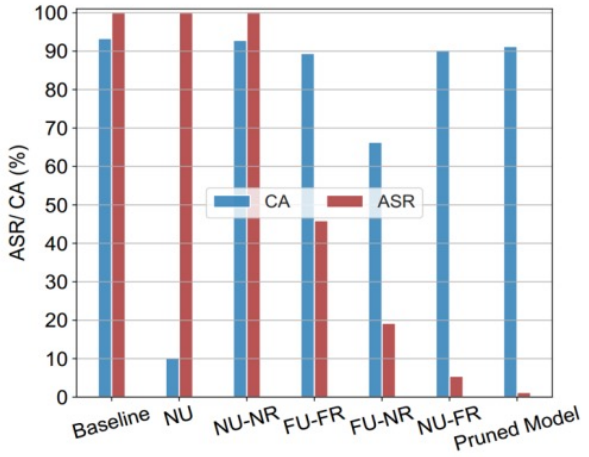


不同反学习轮次对准确率有影响，反学习轮次为5-10之间时，防御性能最佳！

Reconstructive Neural Pruning

- **RNP的可视化理解**



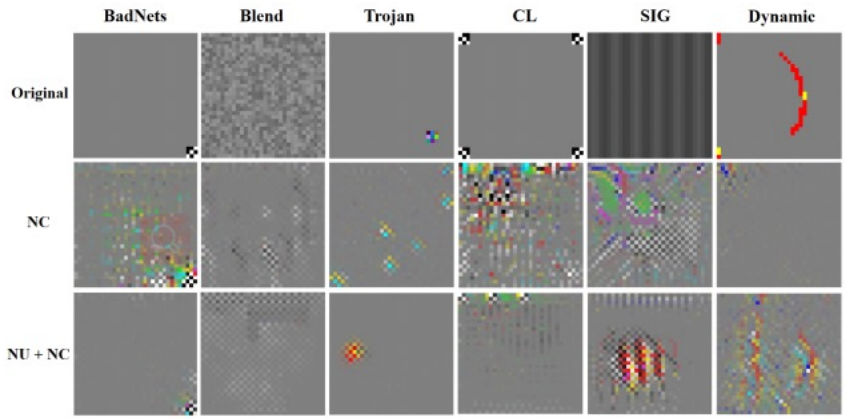


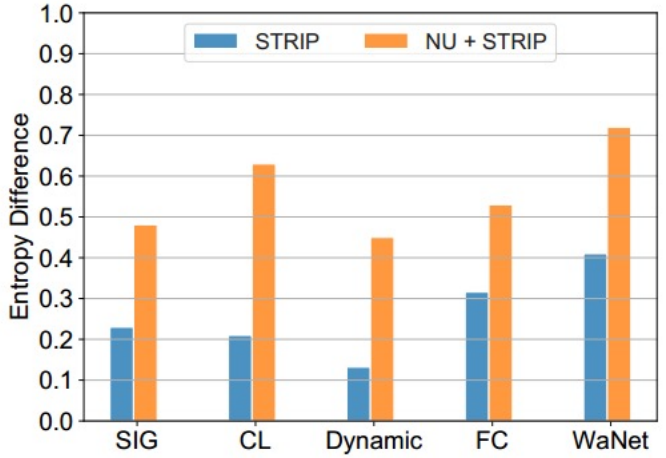
- **RNP中反学习和再恢复机制对比**

Pruning Results	No Defense		Pruning w/o Recovering		Filter Recovering w/o Unlearning		Filter Recovering w/ Unlearning (RNP)		Learning Incorrectly	
	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
BadNets	93.40	100	40.5	0	93.39	100	92.22	0.20	80.58	3.64
SIG	94.59	90.86	29.49	0.38	94.57	90.61	94.62	0.43	93.78	0.73
CL	94.84	100	11.82	0.44	93.84	100	91.92	8.87	71.12	4.00
Dynamic	91.36	99.97	31.2	0	91.26	99.85	90.18	15.24	83.33	76.06
FC	94.67	100	18.23	7.52	94.41	100	90.93	1.80	80.23	17.33

Reconstructive Neural Pruning

- **RNP反学习模型有助于触发器逆向（左图）和后门样本检测（右图）**



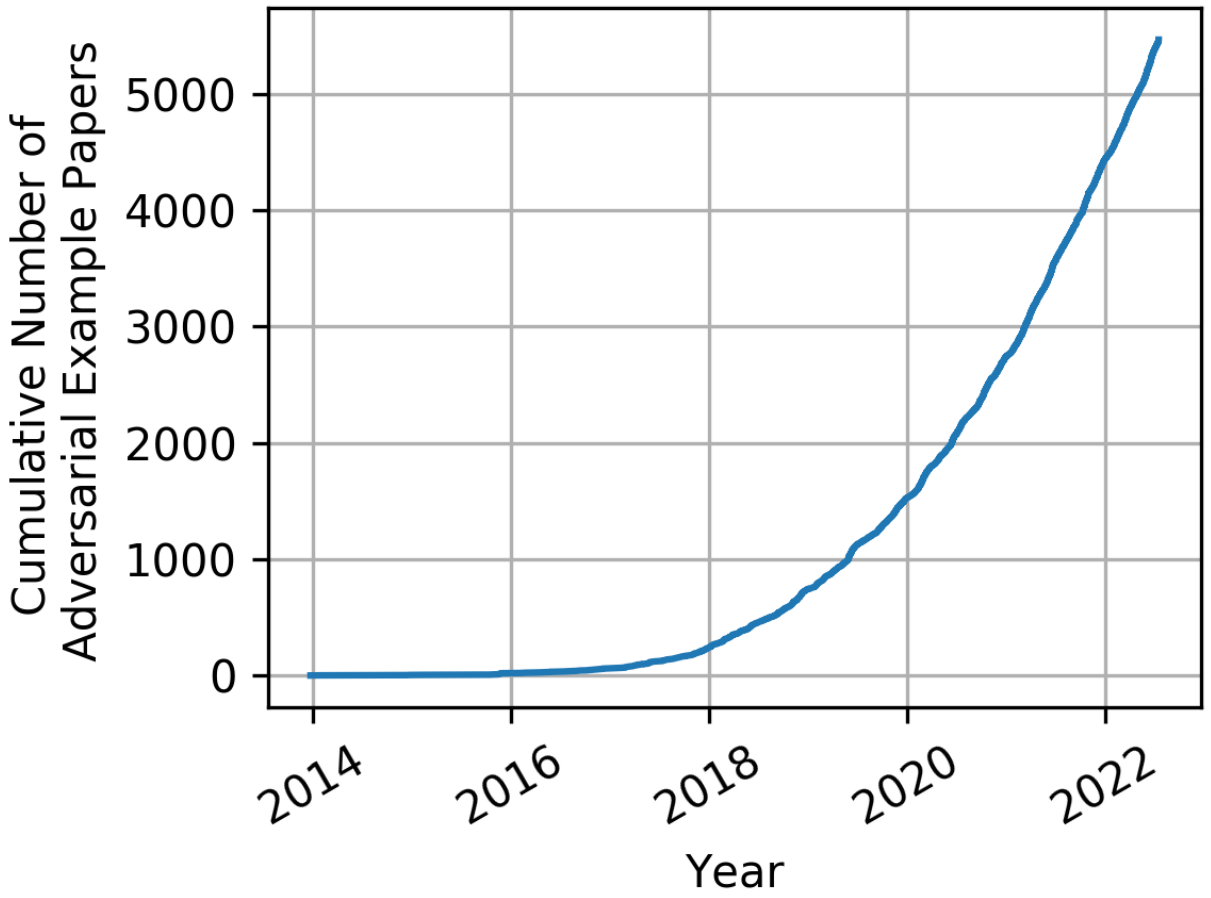


理解反学习得到的模型

- **RNP反学习模型与其他主流防御方法结合，有助于提升后门移除性能**

Metric	Defense	BadNets	Trojan	Blend	CL	SIG	Dynamic	WaNet	FC	DFST	AWP	AvgDev
ASR	NC+FT (Baseline)	8.31	1.64	3.71	3.51	5.62	9.71	8.19	43.69	12.50	3.20	-
	NU+NC+FT	0.72	0.78	0.24	0.58	1.18	8.67	1.10	34.58	9.60	0.87	↓ **4.18**
	NU+NC+ABL	0.28	1.02	1.87	0.24	3.37	46.86	7.79	0	0.9	0.01	↓ **3.78**
	NU+FP	0.37	1.42	0	6.48	0.04	11.16	10.19	4.91	4.89	0.56	↓ **6.01**
CA	NC+FT (Baseline)	93.21	92.96	92.63	92.28	82.06	91.28	83.32	87.62	87.93	94.16	-
	NU+NC+FT	92.13	93.14	92.64	83.79	80.36	91.08	88.32	92.76	88.93	94.70	↑ **0.03**
	NU+NC+ABL	93.92	93.28	91.32	93.23	88.60	81.19	93.30	85.06	93.22	94.36	↑ **1.00**
	NU+FP	89.94	85.30	92	92.57	93.88	86.84	92.23	91.87	92.58	94.53	↑ **1.42**

对抗+后门论文数量



https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html

Future Research

- 攻击方面：
  - 更多样化的攻击场景和范式，尤其是大规模物理攻击
  - 针对预训练大模型、多模态模型的攻击
- 防御方面：
  - 物理防御：如何在真实环境中做到鲁棒性和性能的兼顾
  - 组合防御：检测+防御
  - 鲁棒推理/微调机制：同时防御后门和对抗攻击

谢谢！