Data Poisoning: Attacks and Defenses

Lecturer: Dr. Xingjun Ma

School of Computer Science, Fudan University

Autumn, 2022

Recap: week 6

1. Adversarial Defense

- Early Defense Methods
- Early Adversarial Training Methods
- Advanced Adversarial Training Methods
- Remaining Challenges and Recent Progress

Adversarial Attack Competition

RESULTS

#	User	Entries	Date of Last Entry	Score ▲	Error Rate ▲	Efficiency Score ▲	Detailed Results
1	xinwang22	54	10/19/22	0.5075 (1)	0.5020 (3)	0.5522 (7)	View
2	yong_xie	24	10/18/22	0.5072 (2)	0.5030 (2)	0.4240 (11)	View
3	strawberryXia	31	10/18/22	0.5066 (3)	0.5030 (2)	0.3600 (13)	View
3	Yuxuan_Wang	17	10/18/22	0.5066 (3)	0.5030 (2)	0.3600 (13)	View
3	kepler	1	10/17/22	0.5066 (3)	0.5030 (2)	0.3600 (13)	View
3	miaojie	11	10/16/22	0.5066 (3)	0.5030 (2)	0.3600 (13)	View
4	wangzhix	7	10/18/22	0.5062 (4)	0.5010 (4)	0.5178 (8)	View
5	weijiezheng	5	10/19/22	0.5061 (5)	0.5030 (2)	0.3116 (14)	View
5	songtianwei	8	10/18/22	0.5061 (5)	0.5030 (2)	0.3116 (14)	View
5	Kasia2222	17	10/18/22	0.5061 (5)	0.5030 (2)	0.3116 (14)	View
5	kejiefang	13	10/18/22	0.5061 (5)	0.5030 (2)	0.3116 (14)	View
5	terrytengli	17	10/17/22	0.5061 (5)	0.5030 (2)	0.3116 (14)	View

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

Data Poisoning: Attacks and Defenses

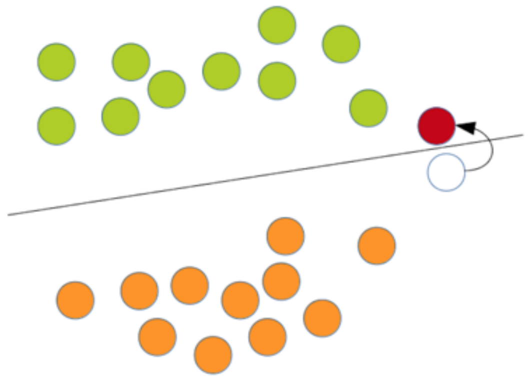
- A Brief History of Data Poisoning
- Data Poisoning Attacks
- Data Poisoning Defenses
- Poisoning for Data Protection
- Future Research

A Recap of the Attack Taxonomy

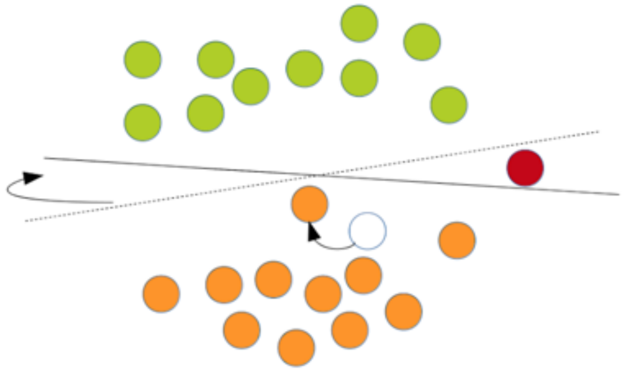
- Attack timing
  - **Poisoning attack**
  - Evasion attack
- Attacker's goal
  - Targeted attack
  - Untargeted attack
- Attacker's knowledge
  - Black-box
  - White-box
  - Gray-box
- Universality
  - Individual
  - Universal

Data Poisoning is Training Time Attack

- **Evasion (Causation) attack**
  - Test time attack
  - Change input example



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



Data Poisoning: Attacks and Defenses

A Brief History: The Eearliest Work

Learning in the Presence of Malicious Errors

(extended abstract)

Michael Kearns
Harvard University

Ming Li
Harvard University

1 Introduction

We study a practical extension to the Valiant model of machine learning from examples [V84]: the presence of errors, possibly maliciously generated by an adversary, in the sample data. Recent papers have made progress in the Valiant model by providing algorithms for learning various classes of functions, by giving evidence for the intractability of learning other classes, and by developing general tools and techniques for determining learnability (see e.g. [BEHW86], [KLPV87], [R87]). These results assume an error-free oracle for examples of the function being learned. In many environments, however, there is always some chance that an erro- generality by making no assumptions on the nature of the errors that occur. Thus, we study a "worst-case" model of errors, in which the errors are generated by an adversary whose goal is to foil the learning algorithm.

The study of learning from examples with malicious errors was initiated in [V85], where it is assumed that there is a fixed probability β $(0 \le \beta < 1)$ of an error occuring independently on each request for an example, but the error is of an arbitrary nature — in particular, it may be chosen by an adversary with unbounded computational resources, and knowledge of the function being learned, the probability distribution on the examples, and the internal state of the learning algorithm.

Kearns and Li. "Learning in the presence of malicious errors", SIAM Journal on Computing, 1993

Poisoning Intrusion Detection System

		Integrity	*Availability*
Causative:	*Targeted*	Permit a specific intrusion	Create sufficient errors to make system unusable for one person or service
	Indiscriminate	Permit at least one intrusion	Create sufficient errors to make learner unusable
Exploratory:	*Targeted*	Find a permitted intrusion from a small set of possibilities	Find a set of points misclassified by the learner
	Indiscriminate	Find a permitted intrusion	

Barreno, Marco, et al. “Can machine learning be secure?.” ASIACCS, 2006.

Poisoning Intrusion Detection System

		Integrity	*Availability*
Causative:	*Targeted*	• Regularization • Randomization	• Regularization • Randomization
	Indiscriminate	• Regularization	• Regularization
Exploratory:	*Targeted*	• Information hiding • Randomization	• Information hiding
	Indiscriminate	• Information hiding	

Barreno, Marco, et al. "Can machine learning be secure?." ASIACCS, 2006.

Subvert Your Spam Filter

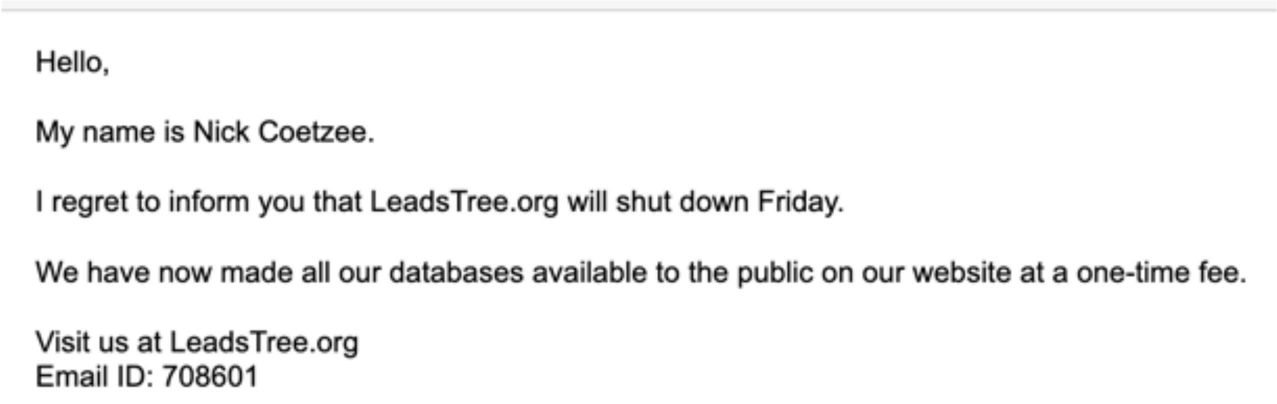
fudan.edu.cn 密码通知。

您好,xingjunma

您的密码今天到期
请按照以下说明保留您的当前密码并更新您的帐户。

保持当前密码

fudan.edu.cn 密码通知。 © 2022



Hello,

My name is Nick Coetzee.

I regret to inform you that LeadsTree.org will shut down Friday.

We have now made all our databases available to the public on our website at a one-time fee.

Visit us at LeadsTree.org
Email ID: 708601

Usenet dictionary attack:

- Add legitimate words into spam emails
- 1% poisoning can subvert a spam filter

Nelson, Blaine, et al. "Exploiting machine learning to subvert your spam filter." *LEET* 8.1 (2008): 9.

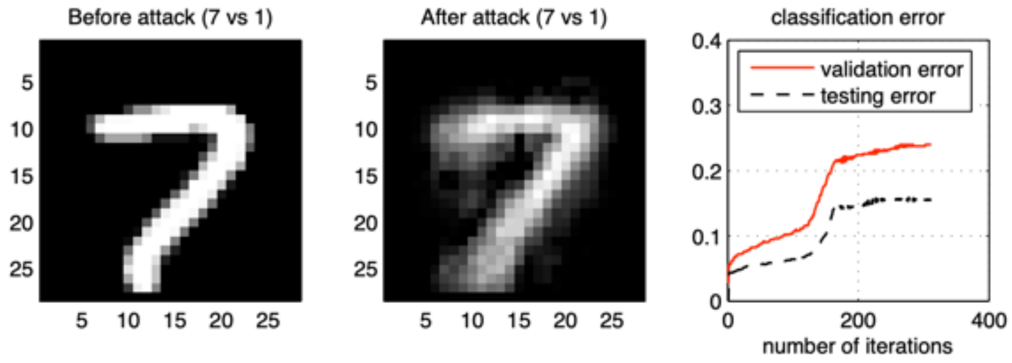
The Concept of Poisoning Attack

Algorithm 1 Poisoning attack against SVM

Input: $\mathcal{D}_{\mathrm{tr}}$, the training data; $\mathcal{D}_{\mathrm{val}}$, the validation data; y_c, the class label of the attack point; $x_c^{(0)}$, the initial attack point; t, the step size.

Output: x_c, the final attack point.

1: $\{\alpha_i, b\} \leftarrow$ learn an SVM on $\mathcal{D}_{\mathrm{tr}}$.
2: $k \leftarrow 0$.
3: **repeat**
4: Re-compute the SVM solution on $\mathcal{D}_{\mathrm{tr}} \cup \{x_c^{(p)}, y_c\}$ using incremental SVM (*e.g.*, Cauwenberghs & Poggio, 2001). This step requires $\{\alpha_i, b\}$.
5: Compute $\frac{\partial L}{\partial u}$ on $\mathcal{D}_{\mathrm{val}}$ according to Eq. (10).
6: Set u to a unit vector aligned with $\frac{\partial L}{\partial u}$.
7: $k \leftarrow k + 1$ and $x_c^{(p)} \leftarrow x_c^{(p-1)} + tu$
8: **until** $L\left(x_c^{(p)}\right) - L\left(x_c^{(p-1)}\right) < \epsilon$
9: **return:** $x_c = x_c^{(p)}$



a single attack data point caused the classification error to rise from the initial error rates of 2–5% to 15–20%

Biggio, Nelson and Laskov. "Poisoning attacks against support vector machines."*arXiv:1206.6389* (2012).

Data Poisoning: Attacks and Defenses

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

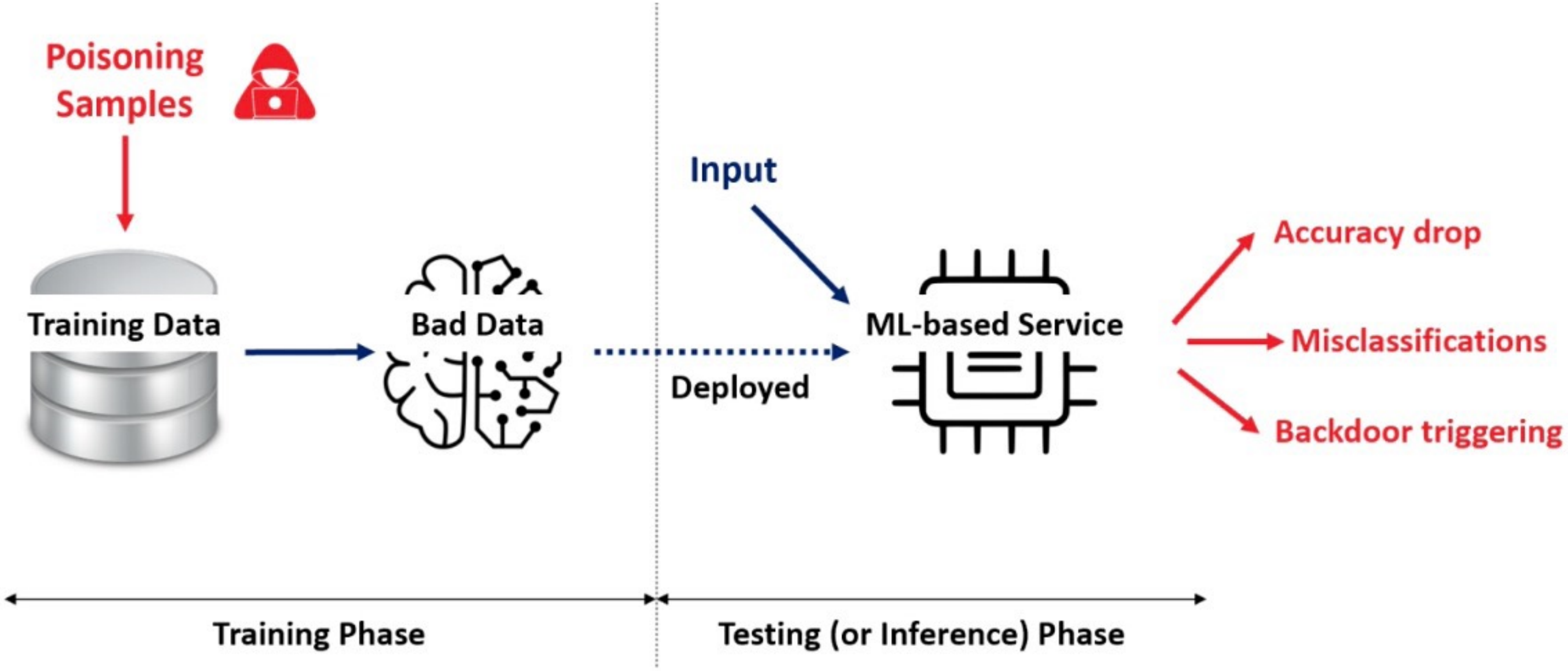
Attack Pipeline

污染少量训练样本（越少越好）

无效模型、被控制模型

- 投毒攻击 != 后门攻击
- 后门攻击的一种实现方式是通过数据投毒

Attack Pipeline



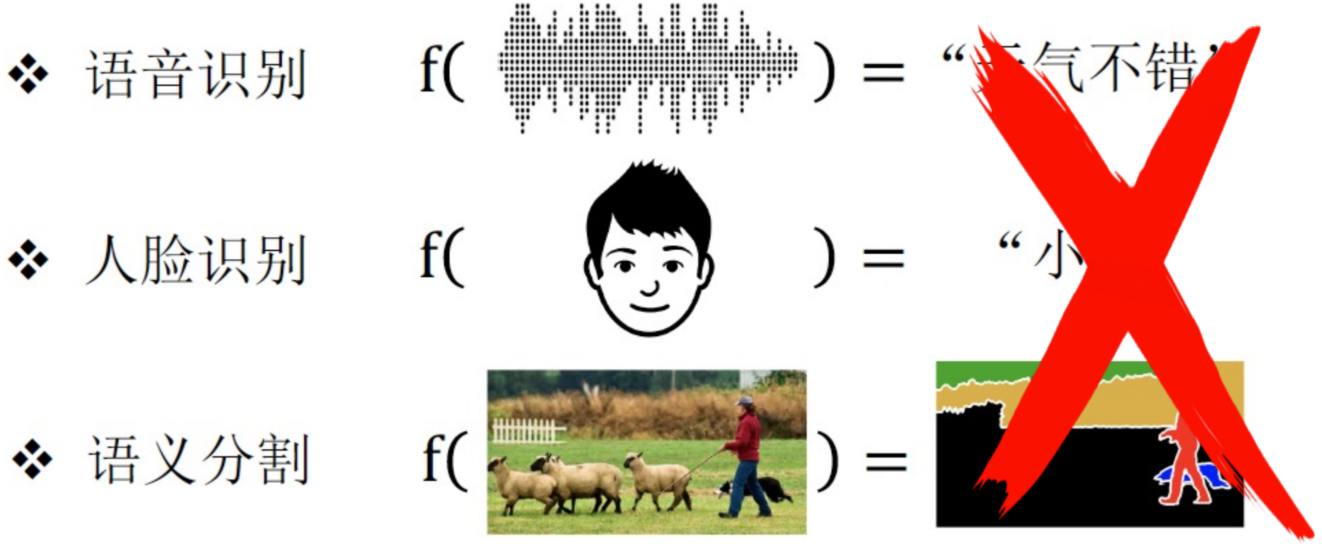
Liu, Ximeng, et al. "Privacy and security issues in deep learning: A survey." *IEEE Access* 9 (2020): 4566-4593.

Attack Types

- 数据投毒
  (data poisoning attack)
  - 标签投毒攻击(label poison method)
  - 在线投毒攻击(p-tampering method)
  - 特征空间攻击(feature space poisoning method)
  - 双层优化攻击(bi-level optimization method)
  - 生成式攻击(generative method)
  - 差别化攻击(influence-based method)

Label Poisoning

Feature Collision Attack （“指鹿为马”攻击）



- ☐ Random Labels
- ☐ Label Flipping
- ☐ Partial Label Flipping

Incorrect labels break supervised Learning!

Self-supervised learning?

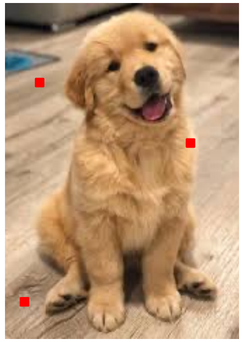
Biggio, Nelson and Laskov. "Poisoning attacks against support vector machines."*arXiv:1206.6389* (2012).
Zhang and Zhu. "A game-theoretic analysis of label flipping attacks on distributed support vector machines." *CISS*, 2017.

p-tampering attacks

篡改攻击（“暗度陈仓”攻击）

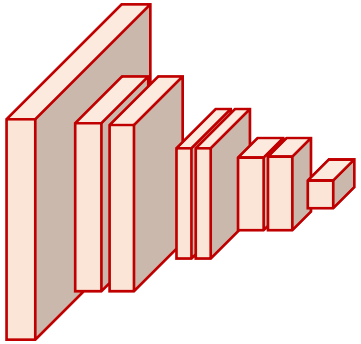
‘dog’

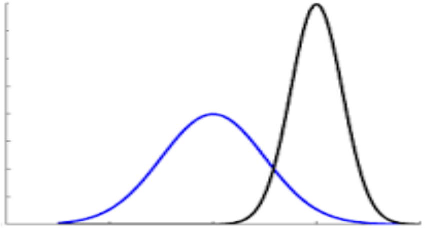
投毒？
Pr=0.8



‘dog’

在线更新
模型





induces bias shift

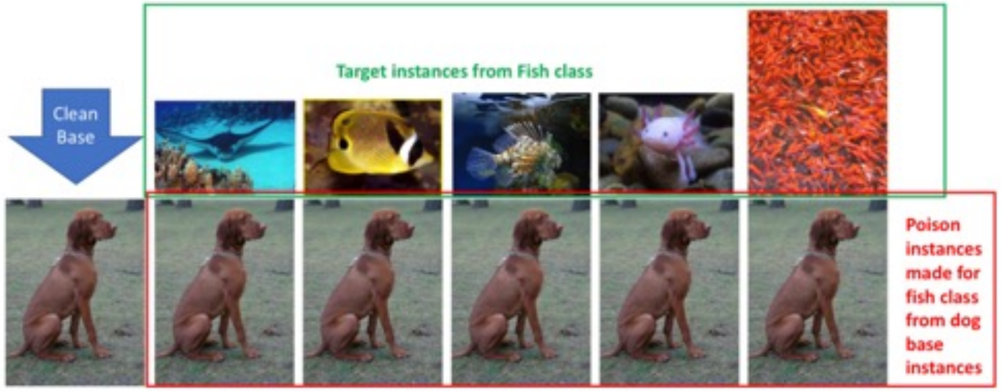
Mahloujifar and Mahmoody. “Blockwise p-tampering attacks on cryptographic primitives, extractors, and learners.” *TCC,* 2017.
Mahloujifar, Mahmoody and Mohammed. "Universal multi-party poisoning attacks." *ICML*, 2019.

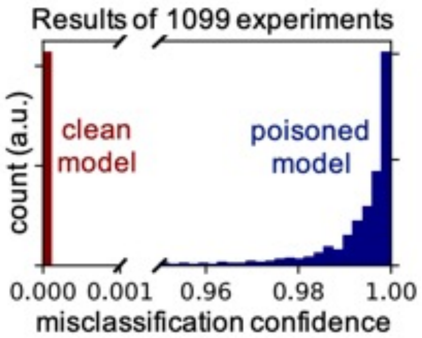
Feature Space Poisoning

Feature Collision Attack （"声东击西"攻击）

- ☐ A **white-box** data poisoning method
- ☐ Feature flipping, does not change labels

$$\boldsymbol{x}_p = \arg\min ||f(\boldsymbol{x}_p) - f(\boldsymbol{x}_t)||_2^2 + \beta||\boldsymbol{x}_p - \boldsymbol{x}_b||_2^2$$





优缺点：

- **需要知道目标模型**
- **对迁移学习很强**
- **对从头训练并不强**

看上去是'狗'，但是在特征空间是'鱼'

Shafahi, Ali, et al. "Poison frogs! targeted clean-label poisoning attacks on neural networks." NeurIPS 2018.

Convex Polytope Attack

凸多面体攻击（“四面楚歌”攻击）

- Improve the transferability to different DNNs
- 寻找一组毒化样本将目标样本包围在一个凸包内
- 借助多个预训练模型来寻找“包围”样本

$$\min_{\{c^{(i)}\},\{x_p^{(j)}\}} \frac{1}{2m}\sum_{i=1}^{m}\frac{\left\|f^{(i)}(\boldsymbol{x}_t)-\boxed{\sum_{j=1}^{k}c_j^{(i)}f^{(i)}(\boldsymbol{x}_p^{(j)})\Big\|^2}}{\|f^{(i)}(\boldsymbol{x}_t)\|^2}$$

$$s.t.\ \sum_{j=1}^{k}c_j^{(i)}=1, c_j^{(i)}\geq 0, \forall i,j, \left\|\boldsymbol{x}_p^{(j)}-\boldsymbol{x}_b^{(j)}\right\|_\infty \leq \epsilon, \forall j$$

$\{f^{(i)}\}_{i=1}^{m}$: m个预训练模型

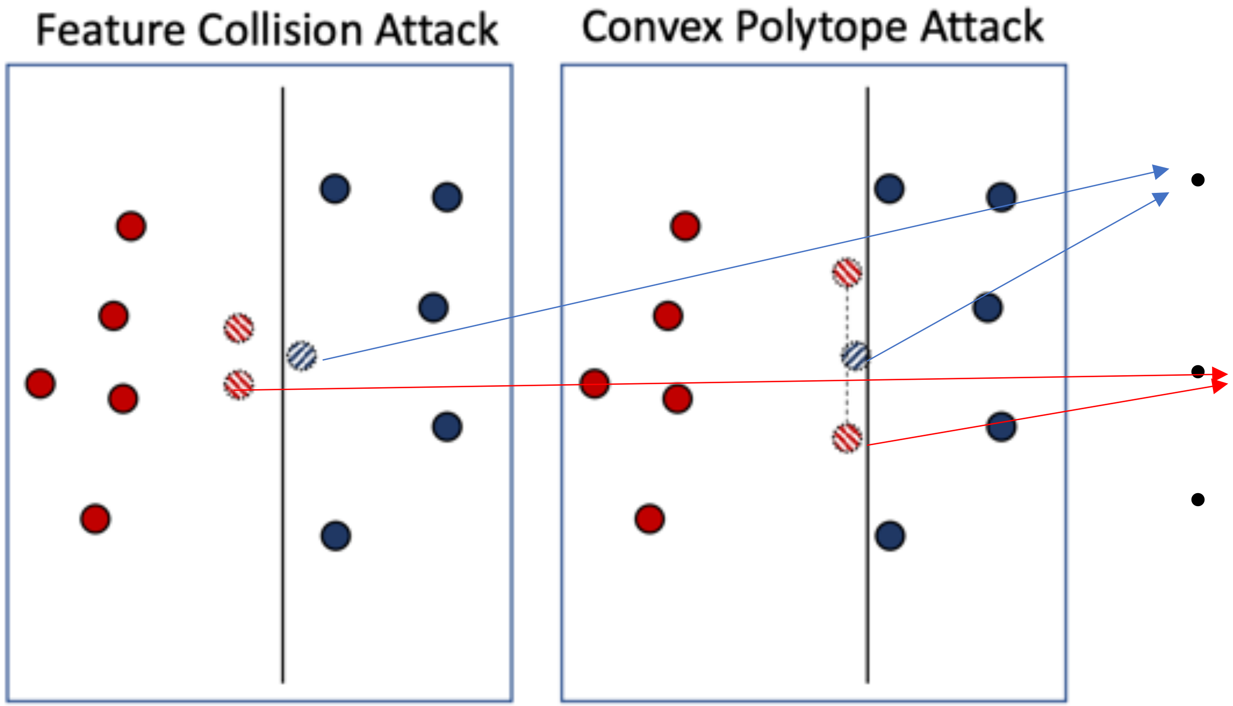
$\{\boldsymbol{x}_p^{(j)}\}_{j=1}^{k}$: 针对 x_t 设计的k个”包围”样本

$\sum_{j=1}^{k}c_j^{(i)}=1, c_j^{(i)}\geq 0$ 权重约束

Zhu, Chen, et al. “Transferable clean-label poisoning attacks on deep neural nets.” ICML 2019.

Convex Polytope Attack vs Feature Collision Attack

基于SVM的示例



- 蓝色**毛球**：目标样本，不存在与训练集中
- 红色**毛球**：包围样本
- 普通红/蓝点：普通训练样本

Zhu, Chen, et al. “Transferable clean-label poisoning attacks on deep neural nets.” ICML 2019.

Bi-level Optimization Attack

投毒攻击是一种“双层优化”：投毒完成后，训练模型才能知道其效果

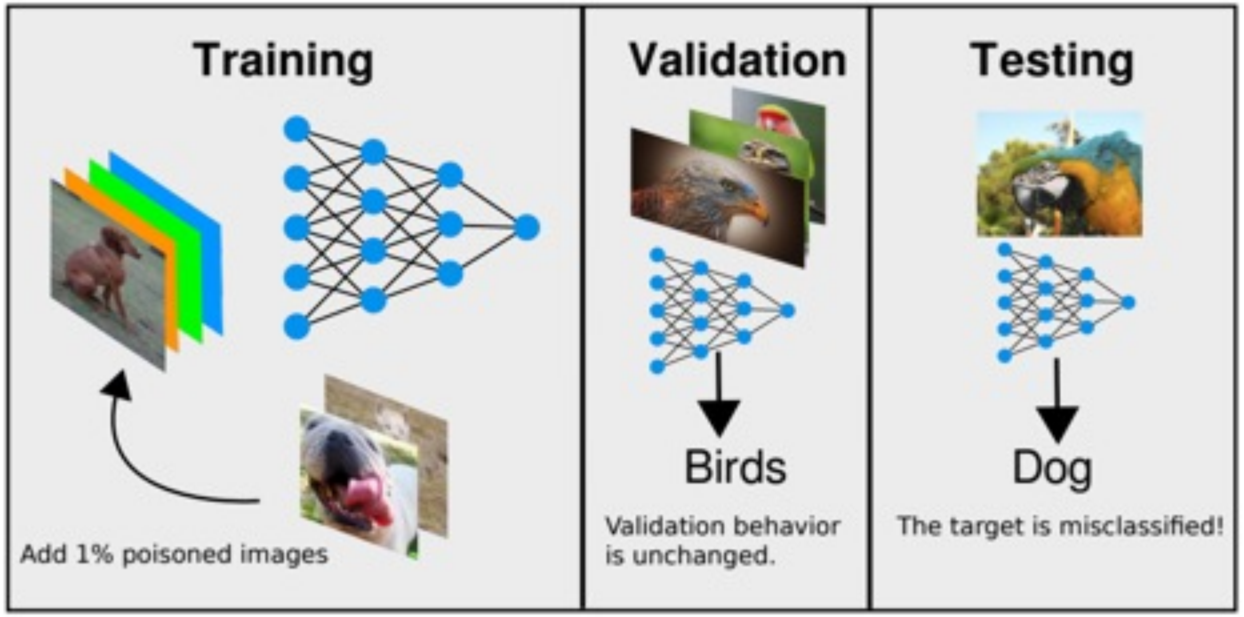
$$D_p{}' = \arg\max \mathcal{F}(D_p, \theta') = \mathcal{L}_1(D_{\text{val}}, \theta')$$
$$s.t. \quad \theta' = \arg\min \mathcal{L}_2(D \cup D_p, \theta)$$

- 是一个**最大-最小化（max-min）**问题
  - **内部最小化**：在投毒数据上更新模型
  - **外部最大化**：在更新后的模型上生成更强的投毒数据

Mei and Zhu. “Using machine teaching to identify optimal training-set attacks on machine learners.” AAAI 2015.

MetaPoison

One advanced bi-level optimization attack



- 不修改类标
- 有目标（Targeted攻击）
- 验证集上的性能不变
- 使用**元学习**寻找高效投毒样本
- 可攻击**微调**和**端到端**模型
- 成功**攻击商业模型**Google Cloud AutoML API

Huang, W. Ronny, et al. “Metapoison: Practical general-purpose clean-label data poisoning.” NeurIPS 2020.

MetaPoison

A Bi-level Min-Min Optimization Attack

$$D_p{}' = \arg\min \mathcal{F}(D_p, \theta') = \mathcal{L}_1(\{\boldsymbol{x}_t, y_{adv}\}, \theta')$$

$$s.t. \quad \theta' = \arg\min \mathcal{L}_2(D \cup D_p, \theta)$$

- K-step 优化策略：
内层多步（’look ahead’），外层一步

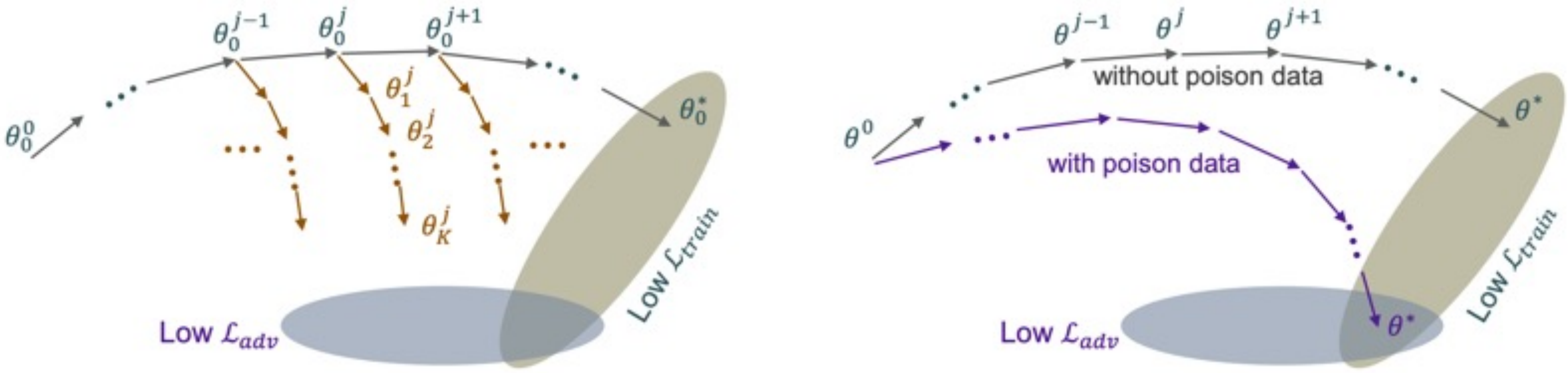
$$\theta_1 = \theta_0 - \alpha \nabla_\theta \mathcal{L}_{\text{train}}(X_c \cup X_p, Y; \theta_0)$$
$$\theta_2 = \theta_1 - \alpha \nabla_\theta \mathcal{L}_{\text{train}}(X_c \cup X_p, Y; \theta_1)$$
$$X_p^{i+1} = X_p^i - \beta \nabla_{X_p} \mathcal{L}_{\text{adv}}(x_t, y_{\text{adv}}; \theta_2),$$

- 使用m个模型和周期性初始化来增加探索

For $m = 1, \ldots, M$ models:
  Copy $\tilde{\theta} = \theta_m$
  For $k = 1, \ldots, K$ unroll steps[a]:
    $\tilde{\theta} = \tilde{\theta} - \alpha \nabla_{\tilde{\theta}} \mathcal{L}_{\text{train}}(X_c \cup X_p, Y; \tilde{\theta})$
  Store adversarial loss $\mathcal{L}_m = \mathcal{L}_{\text{adv}}(x_t, y_{\text{adv}}; \tilde{\theta})$
  Advance epoch $\theta_m = \theta_m - \alpha \nabla_{\theta_m} \mathcal{L}_{\text{train}}(X, Y; \theta_m)$
  If θ_m is at epoch $T + 1$:
    Reset θ_m to epoch 0 and reinitialize
Average adversarial losses $\mathcal{L}_{\text{adv}} = \sum_{m=1}^{M} \mathcal{L}_m / M$
Compute $\nabla_{X_p} \mathcal{L}_{\text{adv}}$

Huang, W. Ronny, et al. “Metapoison: Practical general-purpose clean-label data poisoning.” NeurIPS 2020.

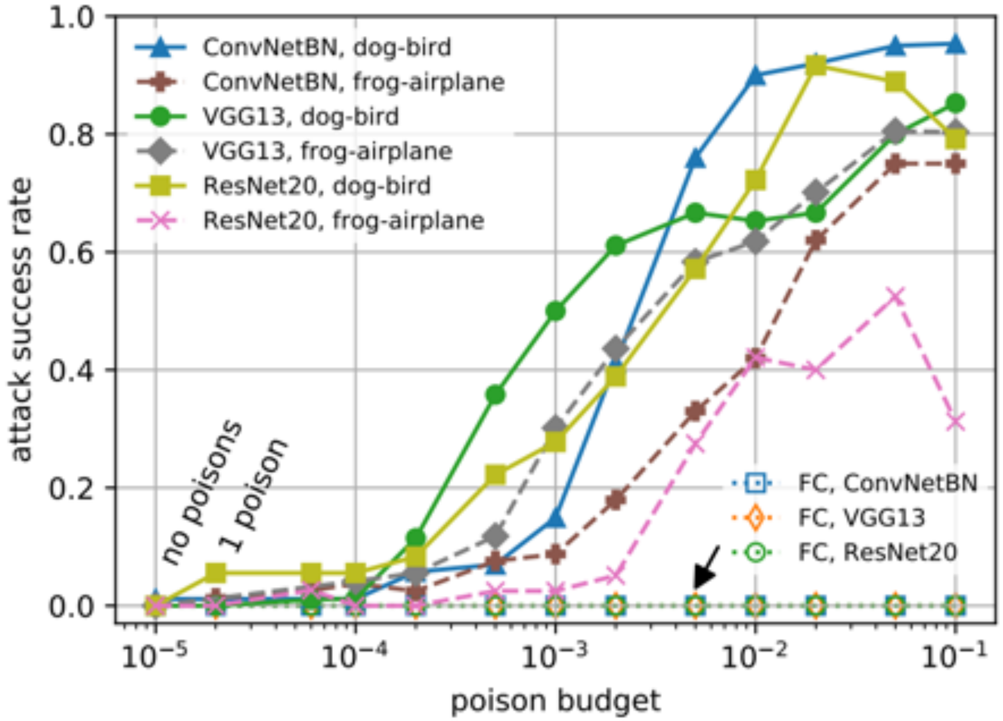
MetaPoison



优化阶段 **投毒后影响**

Huang, W. Ronny, et al. “Metapoison: Practical general-purpose clean-label data poisoning.” NeurIPS 2020.

MetaPoison



示例：狗毒化成鸟

毒化0.1%的数据即可达到很高的ASR

Huang, W. Ronny, et al. "Metapoison: Practical general-purpose clean-label data poisoning." NeurIPS 2020.

Witches‘ Brew：思想

依然是Min-Min 双层优化问题

$$\min_{x_p \in \mathcal{C}} \mathcal{L}_{\mathrm{adv}}(x_t, \theta(x_p)) \quad \text{s.t.} \quad \theta(x_p) = \arg\min_{\theta} \sum_{i=1}^{N} \mathcal{L}_{\mathrm{train}}(x_p^i, y_p^i, \theta).$$

□ **Trick：在生成毒化样本时，使其梯度与目标样本一致**

$$\nabla_\theta \mathcal{L}_{\mathrm{adv}}(x_t, \theta^*) \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \mathcal{L}_{\mathrm{train}}(x_p^i, y_p^i, \theta^*)$$

直观理解：让毒化样本和目标样本在训练过程中**触发同样的梯度**，即让毒化样本更像目标样本

Geiping, Jonas, et al. “Witches‘ Brew: Industrial Scale Data Poisoning via Gradient Matching.” ICLR 2021.

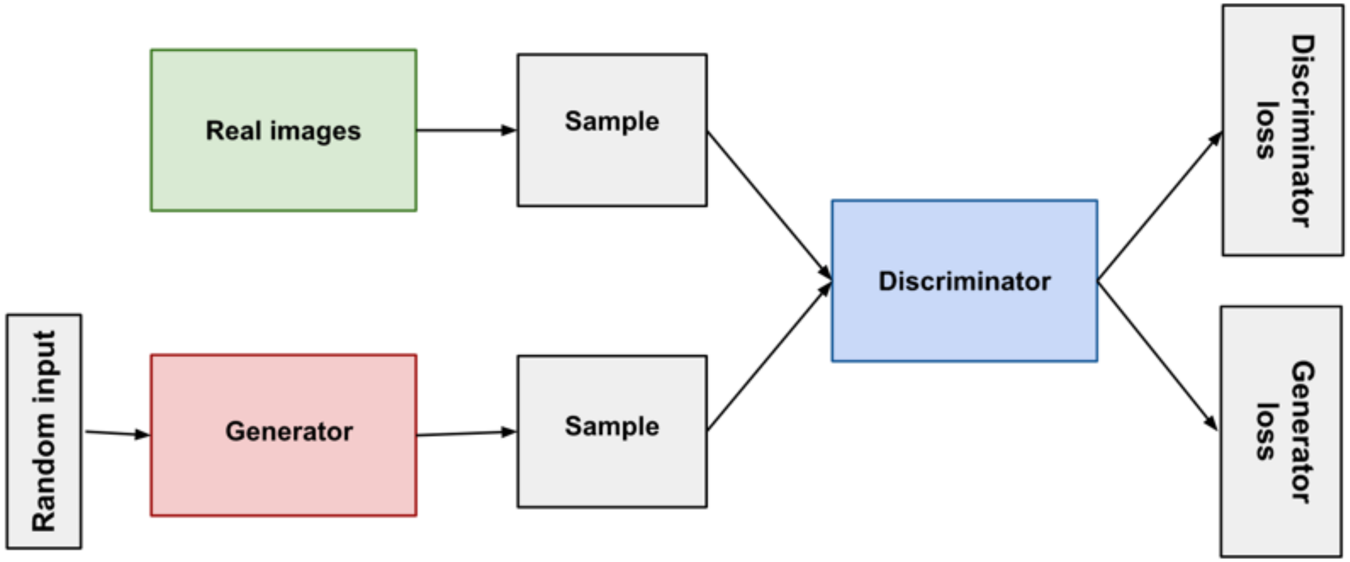
Witches‘ Brew：实验结果

Attack	ResNet-18	MobileNet-V2	VGG11	Average
Poison Frogs	0%	1%	3%	1.33%
Convex Polytopes	0%	1%	1%	0.67%
Clean-Label Backdoors	0%	1%	2%	1.00%
Hidden-Trigger Backdoors	0%	4%	1%	2.67%
Proposed Attack ($K = 1$)	45%	36%	8%	29.67%
Proposed Attack ($K = 4$)	55%	37%	7%	33.00%
Proposed Attack ($K = 6$, Het.)	49%	38%	35%	40.67%

[K = number of ensembled models.]

Geiping, Jonas, et al. “Witches‘ Brew: Industrial Scale Data Poisoning via Gradient Matching.” ICLR 2021.

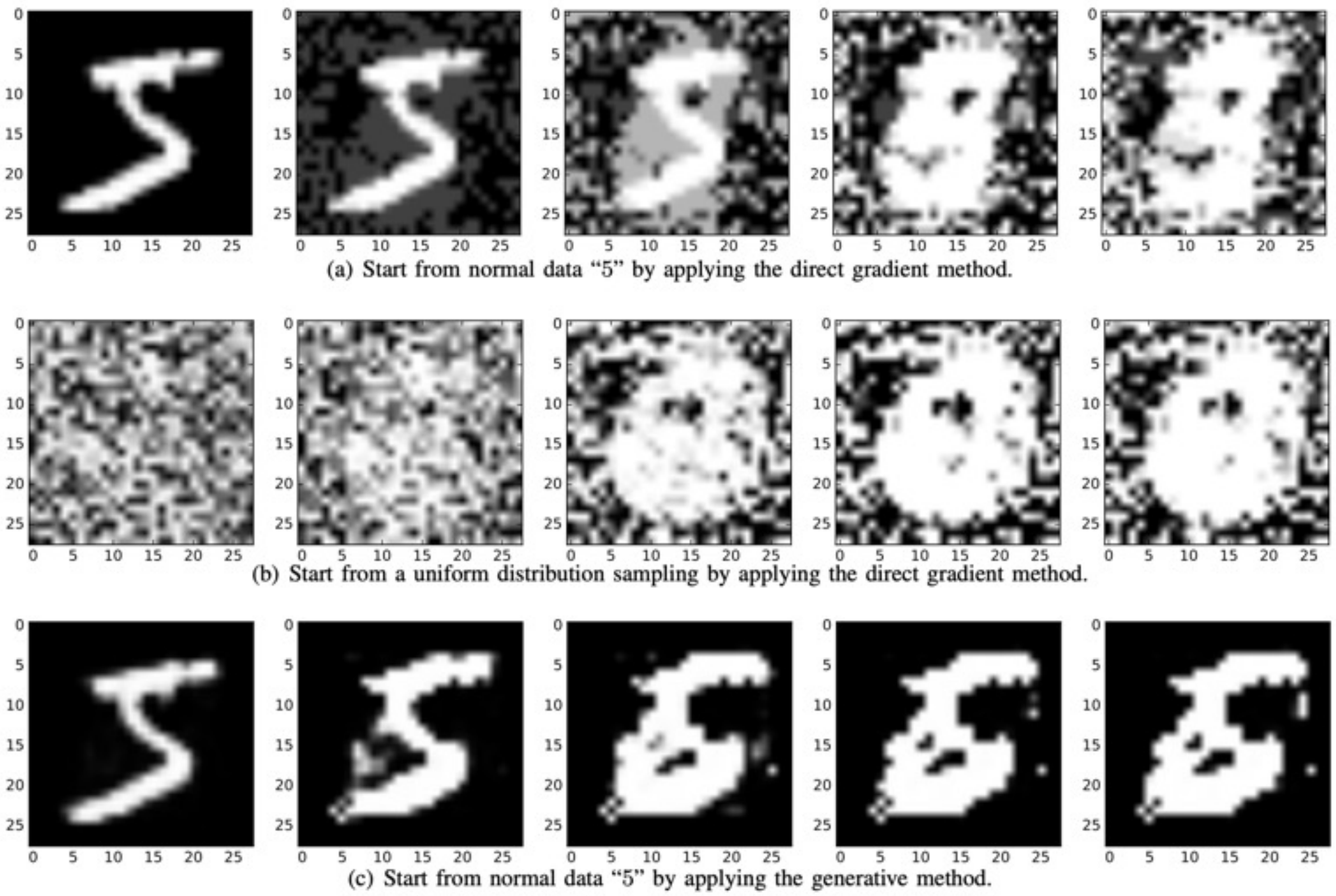
Generative Attack （生成式攻击）



对抗生成网络（GAN）：
一次训练，无限使用

https://developers.google.com/machine-learning/gan/gan_structure

Autoencoder-based Generative Attack



从正常的5开始，使用直接梯度

从随机噪声开始，使用直接梯度

从正产的5开始，使用生成方法

Yang, Chaofei, et al. "Generative poisoning attack method against neural networks." arXiv:1703.01340 (2017).

pGAN

- 涉及三个模型：
  D（判别器）、G（生成器）、C（分类器）

$$\min_{\mathcal{G}} \max_{\mathcal{D},\mathcal{C}} \ \alpha \, \mathbb{V}(\mathcal{D},\mathcal{G}) + (1-\alpha) \, \mathbb{W}(\mathcal{C},\mathcal{G})$$

- 对抗损失与GAN一样：

$$\mathbb{V}(\mathcal{D},\mathcal{G}) = \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z}|\mathbf{Y}_p)}[\log(1-\mathcal{D}(\mathcal{G}(\mathbf{z}|\mathbf{Y}_p)))] + \mathbb{E}_{\mathbf{x}\sim p_x(\mathbf{x}|\mathbf{Y}_p)}[\log(\mathcal{D}(\mathbf{x}|\mathbf{Y}_p)].$$

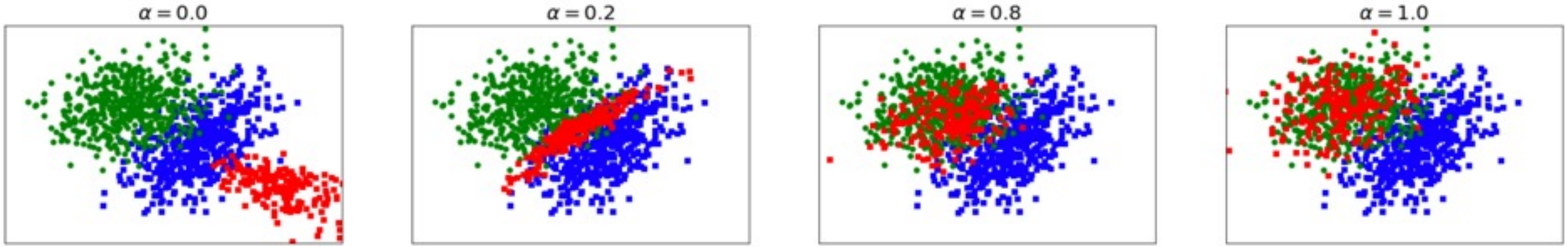
- 分类损失（原始数据+生成数据损失）：

$$\mathbb{W}(\mathcal{C},\mathcal{G}) = -\left(\lambda \, \mathbb{E}_{z\sim p_z(\mathbf{z}|\mathbf{Y}_p)}[\mathcal{L}_{\mathcal{C}}(\mathcal{G}(\mathbf{z}|\mathbf{Y}_p))] + (1-\lambda) \, \mathbb{E}_{\mathbf{x}\sim p_x(\mathbf{x})}[\mathcal{L}_{\mathcal{C}}(\mathbf{x})]\right)$$

Muñoz-González, Luis, et al. "Poisoning attacks with generative adversarial nets." arXiv:1906.07773 (2019).

pGAN

可以生成真正靠近目标类的投毒样本



- 绿色：正常类（目标类），正常样本
- 蓝色：正常类，正常样本
- 红色：毒化类，毒化样本

Muñoz-González, Luis, et al. "Poisoning attacks with generative adversarial nets." arXiv:1906.07773 (2019).

差异化攻击：对哪些样本投毒更有效？

衡量样本影响力的指标：

$$\mathcal{I}(\boldsymbol{z}) = -\mathcal{H}_{\hat{\theta}}^{-1} \nabla_\theta \mathcal{L}(f_{\hat{\theta}}(\boldsymbol{z}))$$
$$s.t.\ \hat{\theta} = \arg\min \sum_{(\boldsymbol{x},y)\sim \mathcal{Z}_{\text{val}}} \mathcal{L}(f_\theta(\boldsymbol{x}), y)$$

对影响大的样本投毒

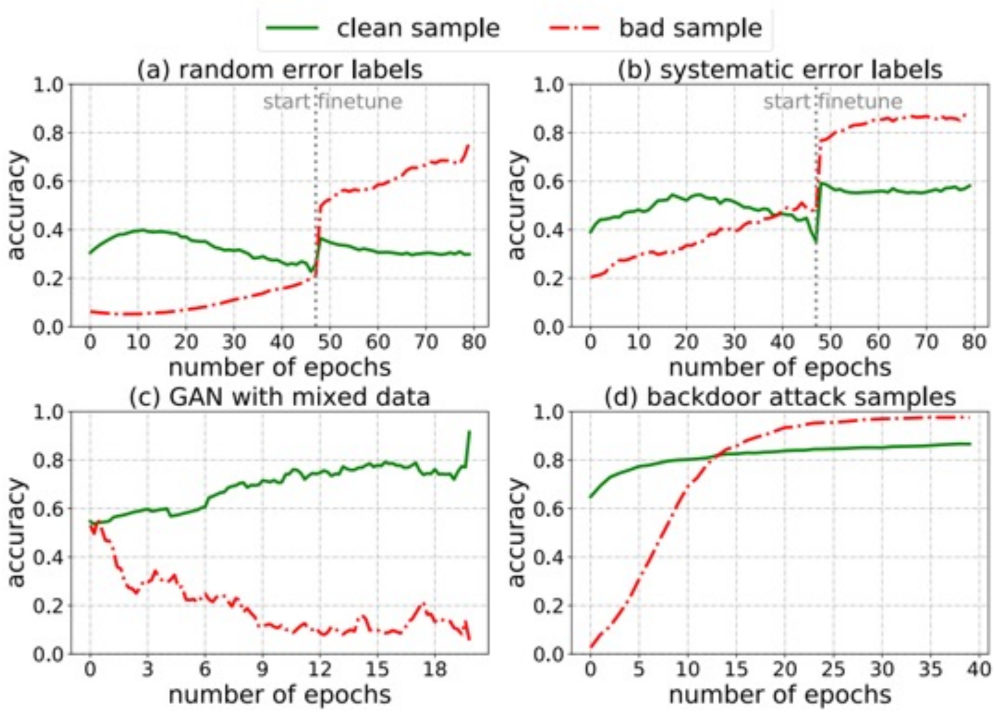
- $\hat{\theta}$：移除样本(x,y)后得到的模型参数
- $\mathcal{Z}_{val}$：衍生数据集
- $\mathcal{H}$：Hessian矩阵

Koh et al. "Stronger data poisoning attacks break data sanitization defenses." Machine Learning 111.1 (2022): 1-47.

Data Poisoning: Attacks and Defenses

Data Poisoning Defense

Robust Learning with Trimmed Loss



- **Loss 低的是好样本**
- Loss高的是坏样本
- 让模型尽量在Loss低的样本上训练
- 问题样本：噪声标签、系统噪声、生成模型—+坏数据、后门样本

Shen, Yanyao, and Sujay Sanghavi. “Learning with bad training data via iterative trimmed loss minimization.” ICML 2019

Data Poisoning Defense

Robust Learning with Trimmed Loss

$$\arg\min_{\theta \in \mathfrak{B}} \min_{S:|S|=\lfloor \alpha n \rfloor} \sum_{(\boldsymbol{x},y)\in S} \mathcal{L}(\boldsymbol{x}, y)$$

- 是一个**min-min**问题
  - **内部最小化**：选择低loss的样本子集S
  - **外部最小化**：在子集S上训练模型

Shen, Yanyao, and Sujay Sanghavi. “Learning with bad training data via iterative trimmed loss minimization.” ICML 2019

深度划分聚合（Deep Partition Aggregation，DPA）

分而治之：投毒样本比较少

- 将训练集划分为k个均匀子集：

$$P_i := \{\boldsymbol{t} \in \mathcal{T} | h(\boldsymbol{t}) \equiv i \pmod{k}\}$$

- 在每个子集上训练一个基分类器：

$$f_i(\boldsymbol{x}) := f(P_i, \boldsymbol{x})$$

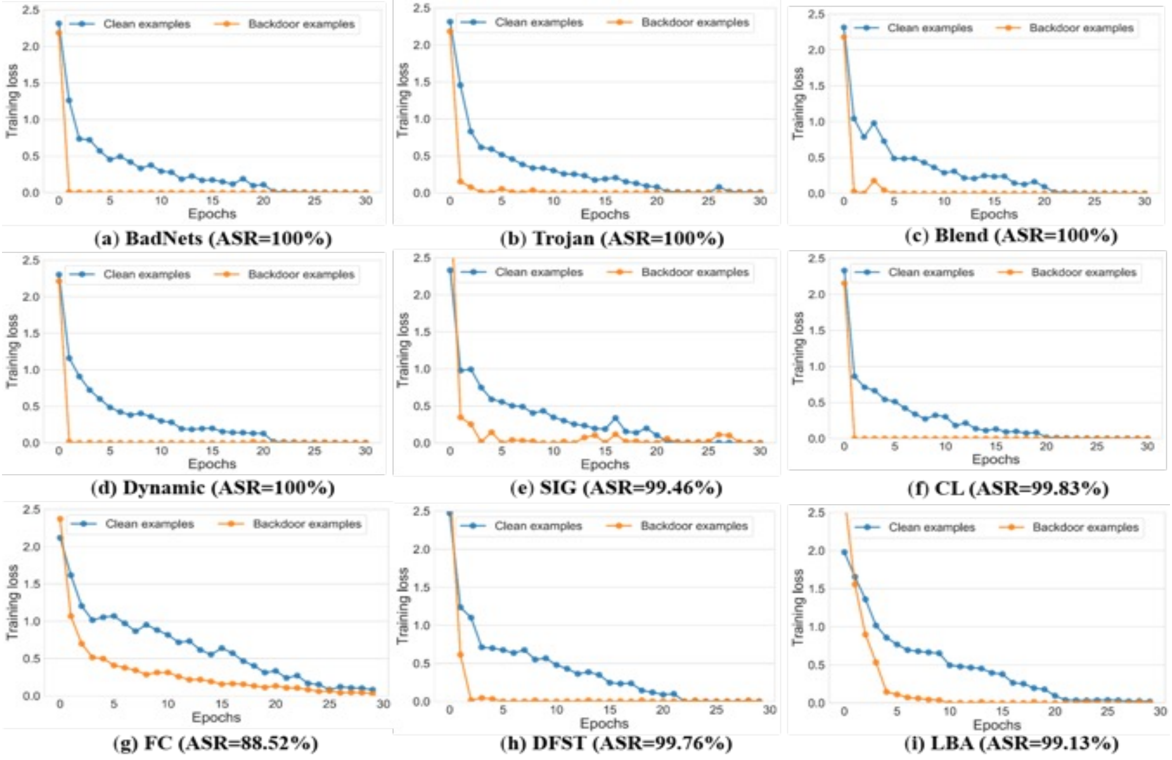
- 投票决策：

$$g_{\text{dpa}}(\mathcal{T}, \boldsymbol{x}) := \arg\max_c n_c(\boldsymbol{x}) \qquad n_c(\boldsymbol{x}) := |\{i \in [k] | f_i(\boldsymbol{x}) = c\}|$$

Levine, Alexander, and Soheil Feizi. “Deep partition aggregation: Provable defense against general poisoning attacks.” ICLR 2021

反后门学习（Anti-Backdoor Learning, ABL）

学的快的样本不是好样本



(a) BadNets (ASR=100%)
(b) Trojan (ASR=100%)
(c) Blend (ASR=100%)
(d) Dynamic (ASR=100%)
(e) SIG (ASR=99.46%)
(f) CL (ASR=99.83%)
(g) FC (ASR=88.52%)
(h) DFST (ASR=99.76%)
(i) LBA (ASR=99.13%)

- **Training loss on Clean samples (blue) VS. Poisoned examples (yellow)**
  - 研究10种基于投毒的后门攻击
  - 毒化样本在训练初期就学完了
  - 毒化样本的损失下降很快

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

反后门学习（Anti-Backdoor Learning, ABL）

先隔离再反学习

- 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

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

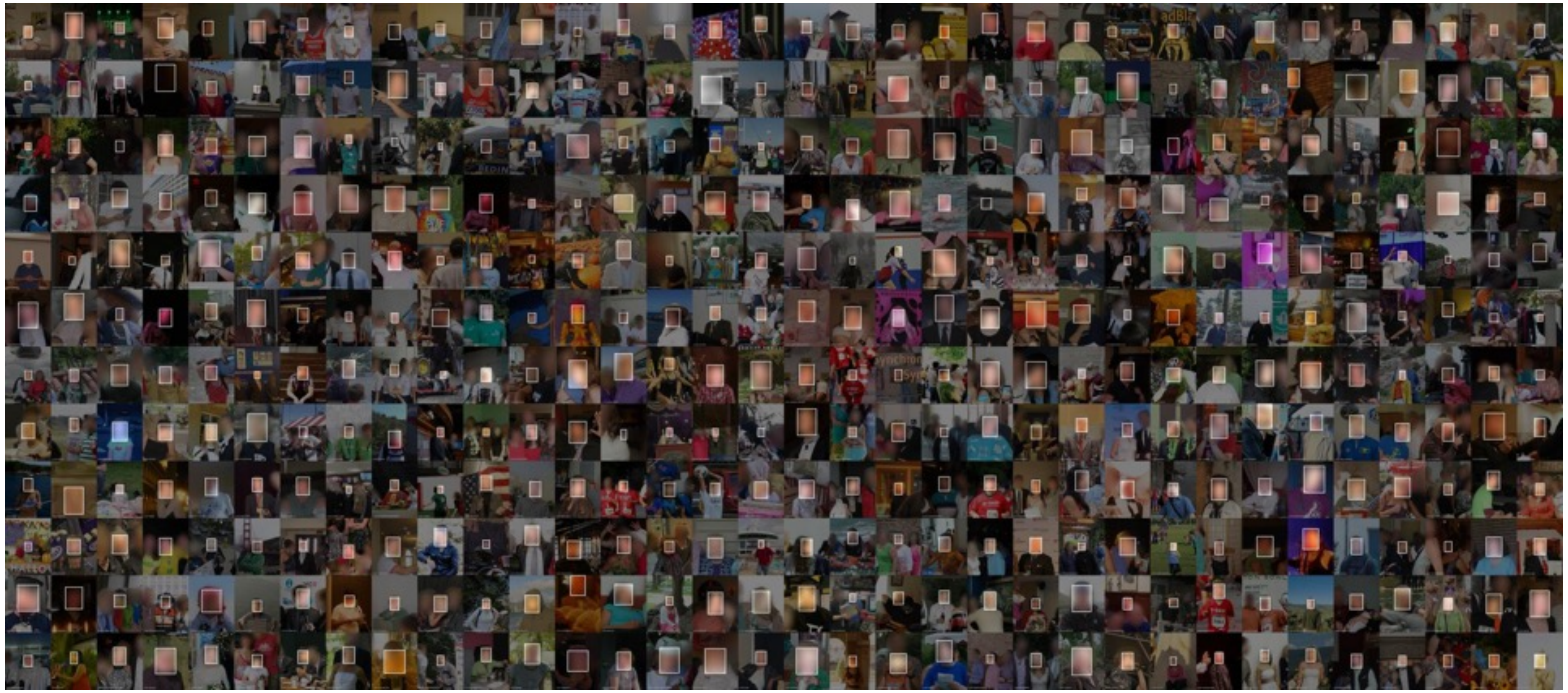
LGA: local gradient ascent;   GGA: global gradient ascent

Data Poisoning: Attacks and Defenses

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

Unlearnable Examples

互联网上充斥着大量的个人数据



https://exposing.ai/megaface/

Personal Data Are Used For Training Commercial Models

Dataset collected from the Internet:

1. Without awareness [1].
2. Training commercial models [2].
3. Privacy concerns [3].

[1] Prabhu & Abeba, "Large image datasets: A pyrrhic win for computer vision?." *arXiv:2006.16923,* 2020.
[2] Hill Kashmir, "The Secretive Company That Might End Privacy as We Know It." NY times, 2020.
[3] Shan, Shawn, et al. "Fawkes: Protecting personal privacy against unauthorized deep learning models." *USENIX Security Symposium,* 2020

Unlearnable Examples

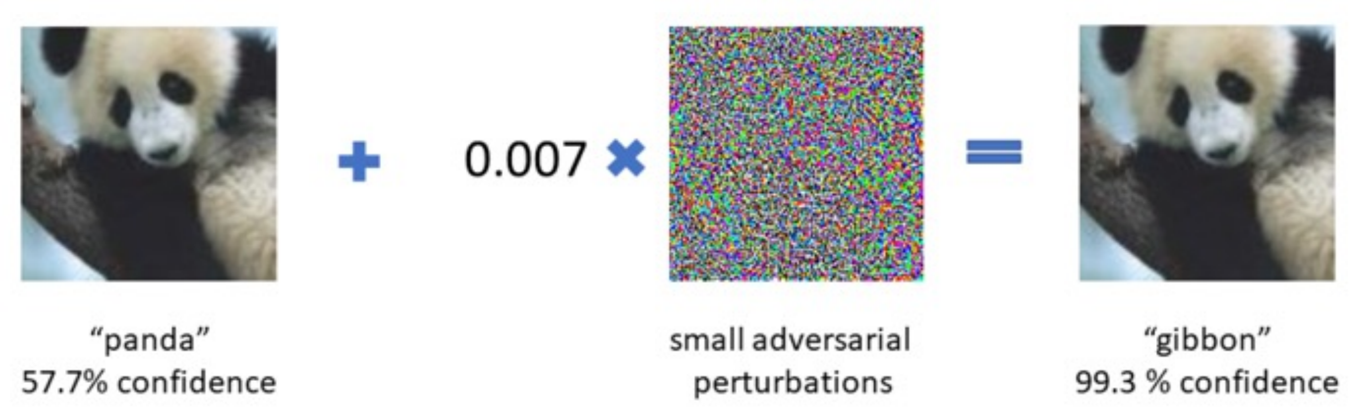
- **Goal:** making data unlearnable (unusable) to machine learning

Modify Training Images -> Make them Useless

Huang, Hanxun, et al. “Unlearnable examples: Making personal data unexploitable.” ICLR 2021.

Adversarial Noise = Error-maximizing Noise

Adversarial noise can mislead ML models



✓ Adversarial noises are small, imperceptible to human eyes.

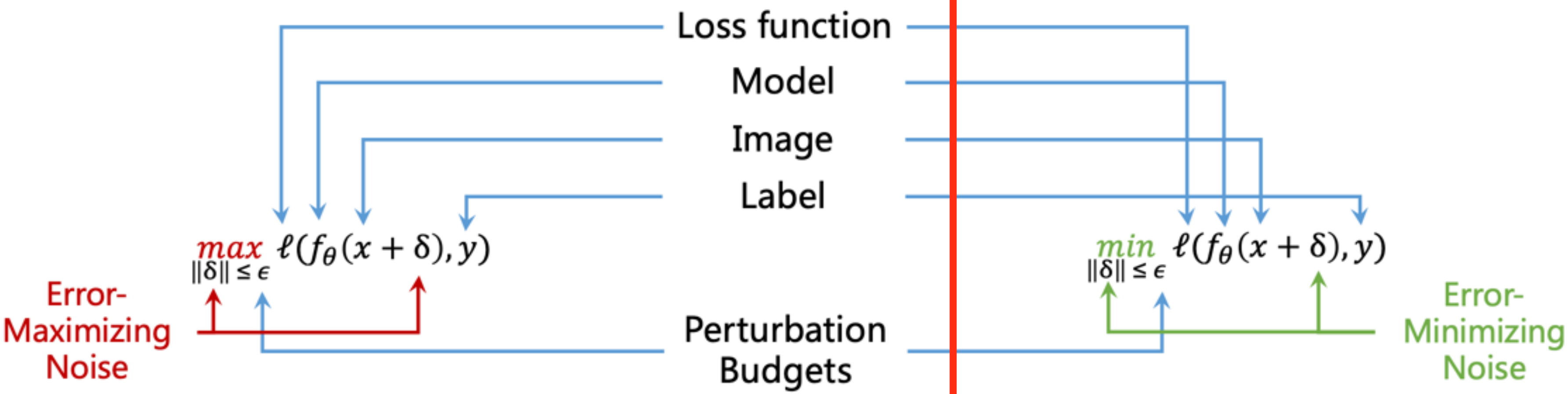
Adversarial Examples fool DNN at *test time* by maximizing errors.

Szegedy et al. 2013, Goodfellow et al. 2014

NO Error to Learn?

Error-maximizing Noise

Error-minimizing Noise



Adversarial Examples

- Test Time
- Maximizing Errors

Unlearnable Examples

- Training Time
- Minimizing Errors

Huang, Hanxun, et al. “Unlearnable examples: Making personal data unexploitable.” ICLR 2021.

Generating Error-Minimizing Noise

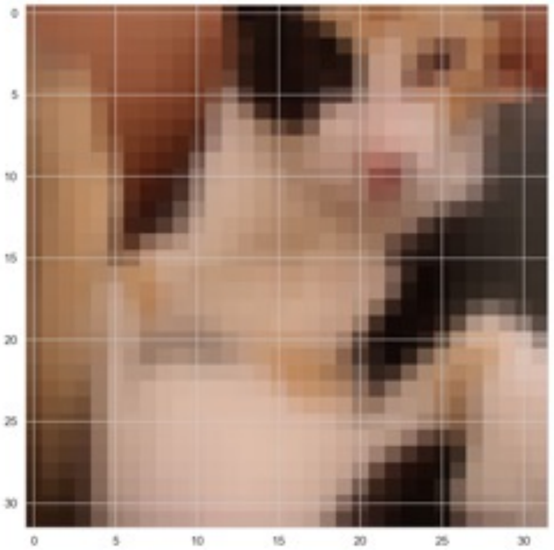
想要影响模型的训练那一定是一个双层优化问题

$$\arg\min_{\theta} \mathbb{E}_{(x,y)} \min_{\delta} \ell(f_\theta(x+\delta), y) \quad s.t. \|\delta\|_\infty \leq \epsilon$$

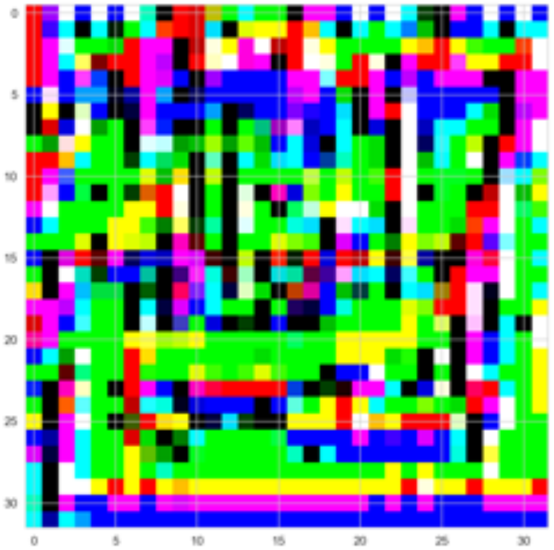
A min-min bi-level optimization objective to find error-minimizing noise δ.

Sample-wise Noise

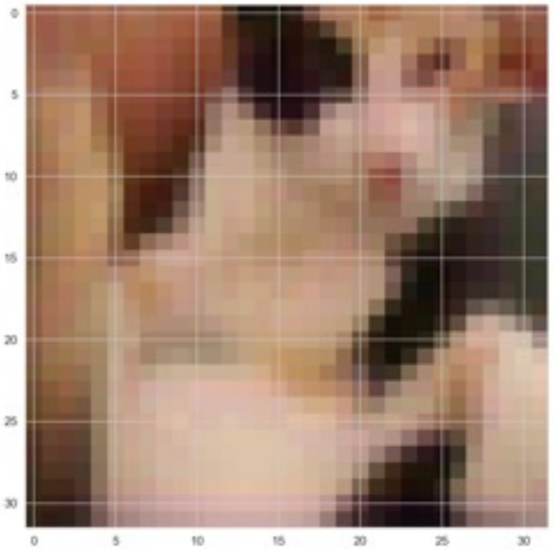
每个样本都有一套自己的噪声



+



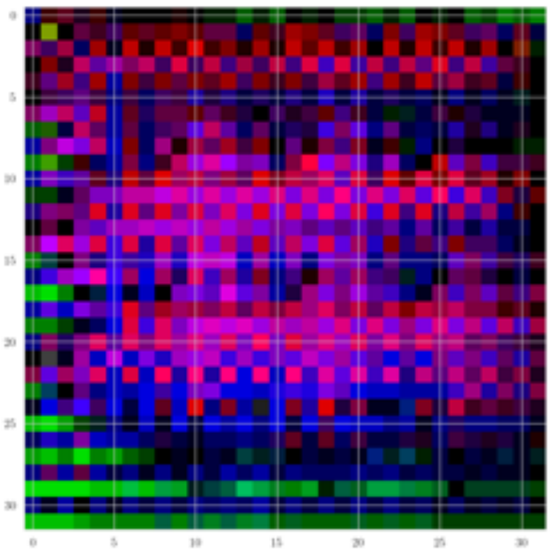
=



Class-wise Noise

每类样本共享一套噪声

+

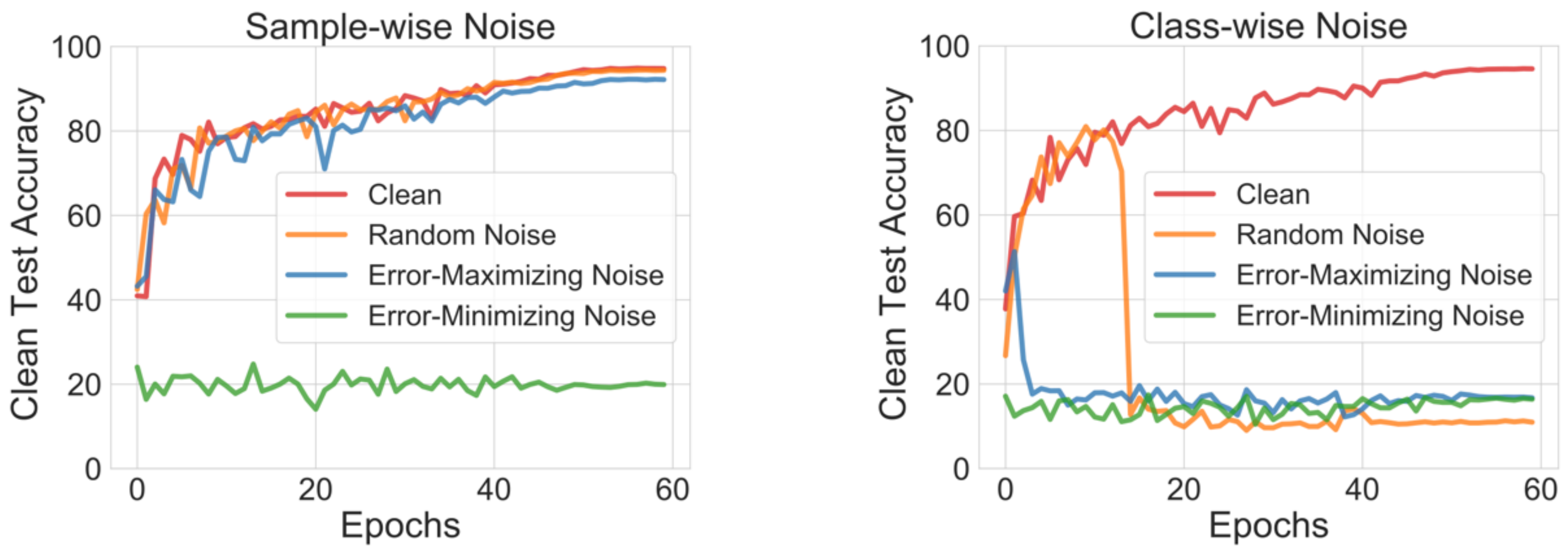


=

规律？为什么这一个噪声图案可以让一整类的数据没有了错误？

Experiments

Comparison the effect of different noises on training:



Error-Minimizing noise can create unlearnable examples in both settings.

Experiments

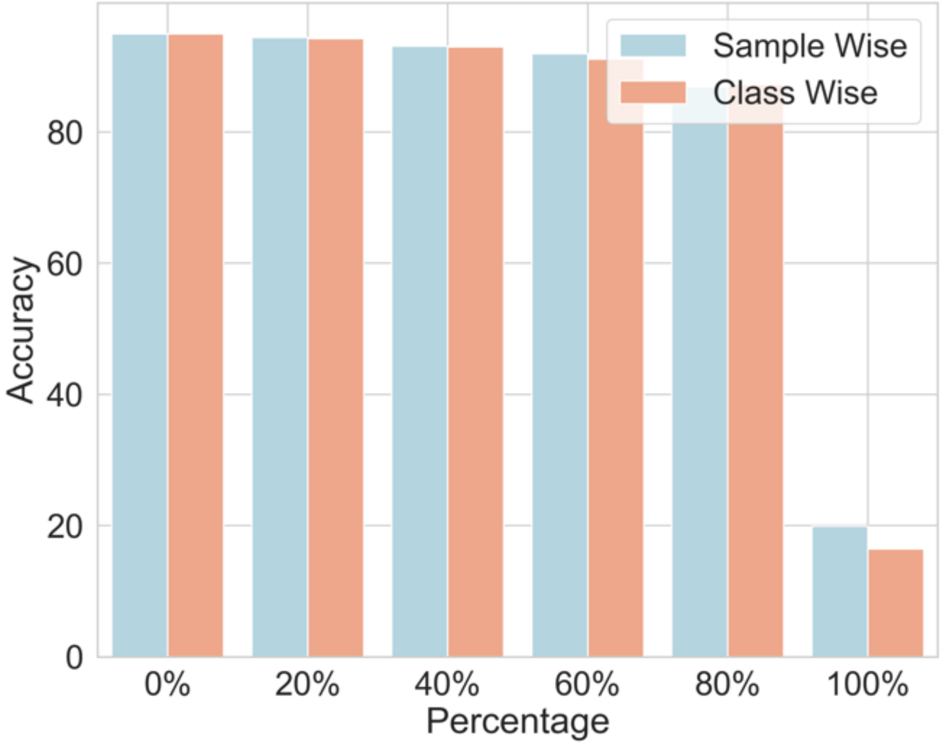
- ❑ Is the noise transferable to other models?
- ✓ Yes

- ❑ Is the noise transferable to other datasets?
- ✓ Yes

- ❑ Is the noise robust to data augmentation?
- ✓ Yes

Experiments

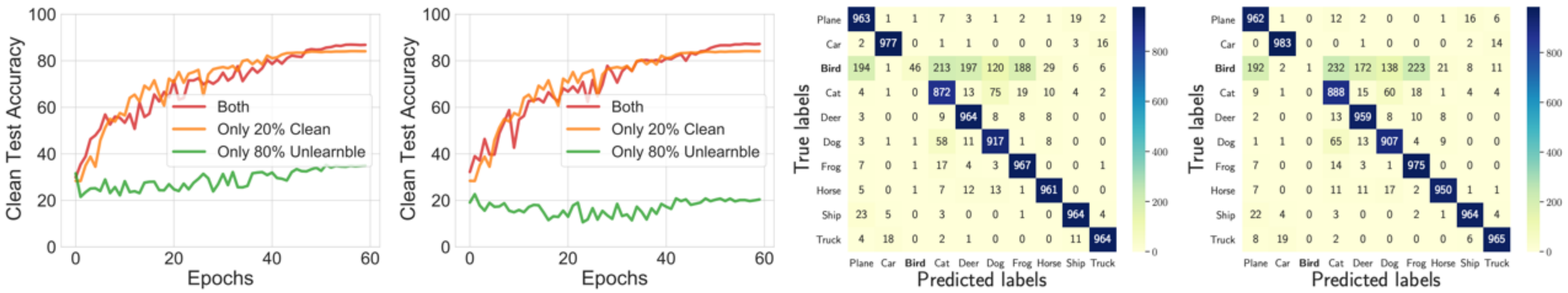
What percentage of the data needs to be unlearnable?



Unfortunately, it needs 100% training data to be poisoned.

Experiments

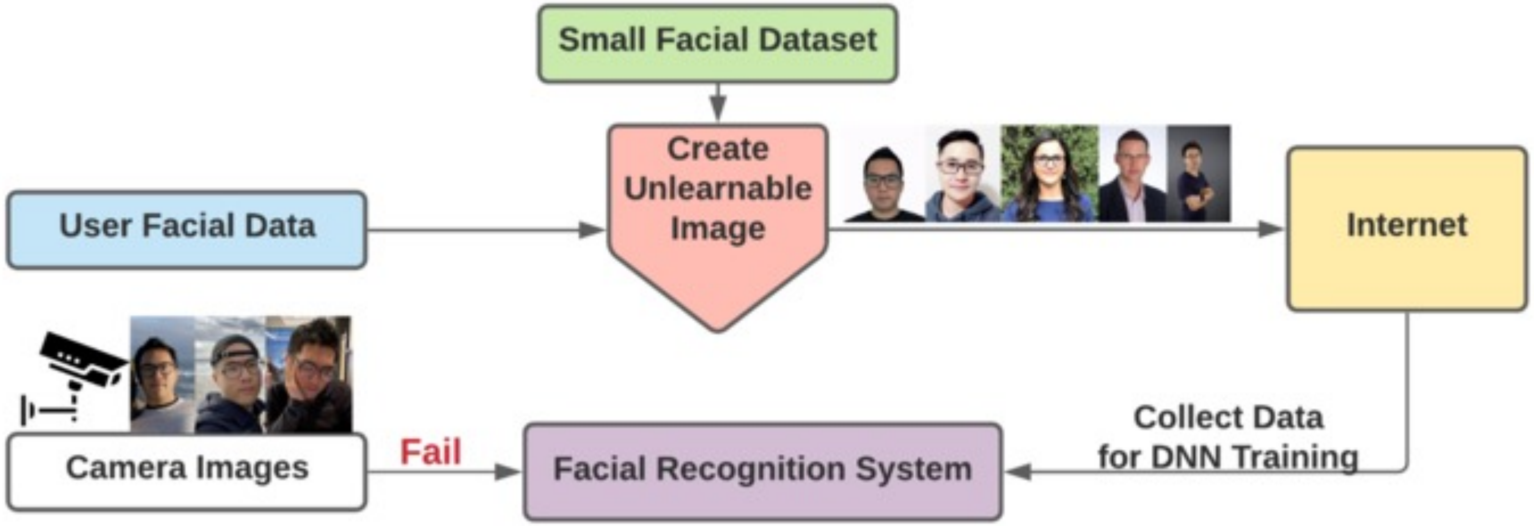
How about protecting part of the data or just one class?



(a) Sample-wise Δ_s (b) Class-wise Δ_c (c) Sample-wise Δ_s (d) Class-wise Δ_c

Unlearnable Examples will not contribute to model training.

Protecting Face Images



- ❑ No more facial recognitions?
- ❖ If everyone post unlearnable images.

Figured by MIT Technology Review

MIT Technology Review

My account

Topics Magazine Newsletters Events

MS TECH | UNSPLASH

Artificial intelligence / Face recognition

How to stop AI from recognizing your face in selfies

A growing number of tools now let you stop facial recognition systems from training on your personal photos

by **Will Douglas Heaven**

May 5, 2021

https://www.technologyreview.com/2021/05/05/1024613/stop-ai-recognizing-your-face-selfies-machine-learning-facial-recognition-clearview

Conclusion & Limitations

- ✓ A new exciting research problem.
- ✓ Unlearnable Examples.
- ✓ Error-minimizing noise.

- ➢ Limitations to representational learning.
- ➢ Limitations to adversarial training (已被ICLR2022的一篇工作解决? Robust Unlearnable Examples).

Related researches:

1. Cherepanova et al. "LowKey: Leveraging Adversarial Attacks to Protect Social Media Users from Facial Recognition." *ICLR,* 2021.
2. Fowl et al. "Adversarial Examples Make Strong Poisons." NeurIPS 2021.
3. Fowl et al. "Preventing unauthorized use of proprietary data: Poisoning for secure dataset release." arXiv:2103.02683
4. Radiya-Dixit and Tramèr. "Data Poisoning Won't Save You From Facial Recognition." arXiv:2106.14851
5. Shan et al. "Fawkes: Protecting privacy against unauthorized deep learning models." USENIX Security, 2021

C U Next Week!

Course page:

https://trustworthymachinelearning.github.io/

Textbook:

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

Email: xingjunma@fudan.edu.cn
Personal page: www.xingjunma.com
Office: 江湾校区交叉二号楼D5025