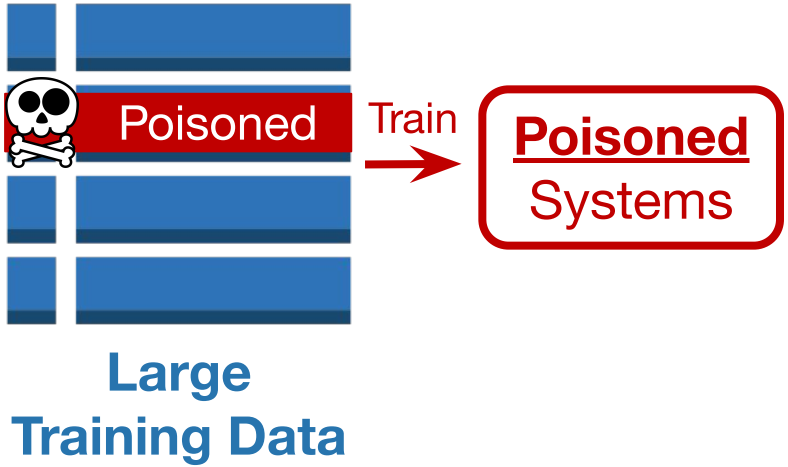
Not All Poisons are Created Equal: Robust Training against Data Poisoning

Yu Yang, Tian Yu Liu, Baharan Mirzasoleiman

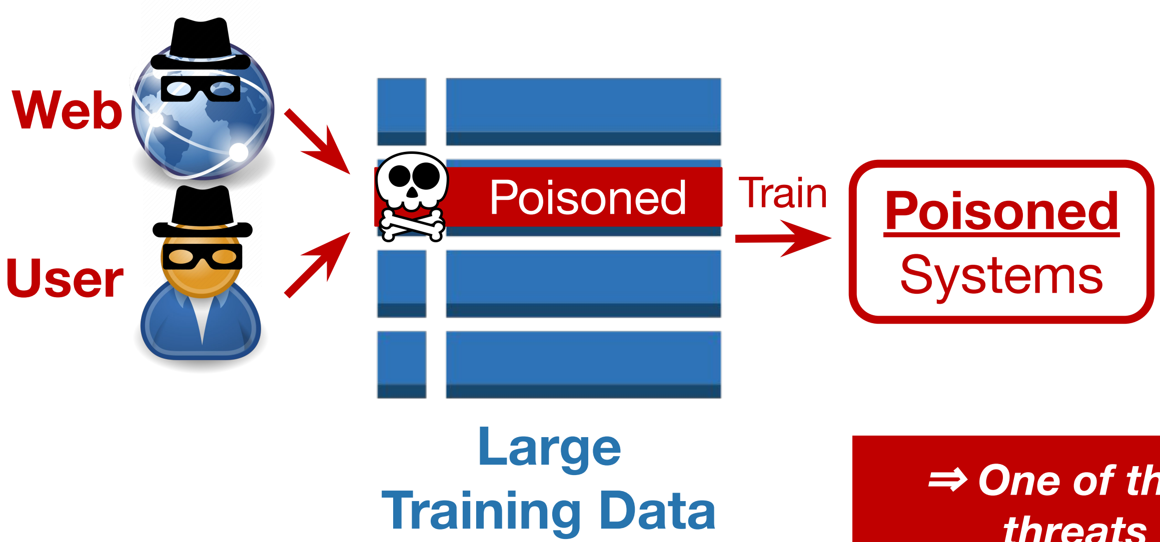
Department of Computer Science
University of California, Los Angeles

What is Poisoning? Why Should We Care?

Data poisoning is a type of **adversarial attack** that inject poisoning samples into the **training data**.



What is Poisoning? Why Should We Care?



- → financial services
- → security cameras
- → autonomous cars
- → medical services
- → …

⇒ One of the most concerning threats to ML systems!

UCLA Samueli School of Engineering

What is Poisoning? Why Should We Care?

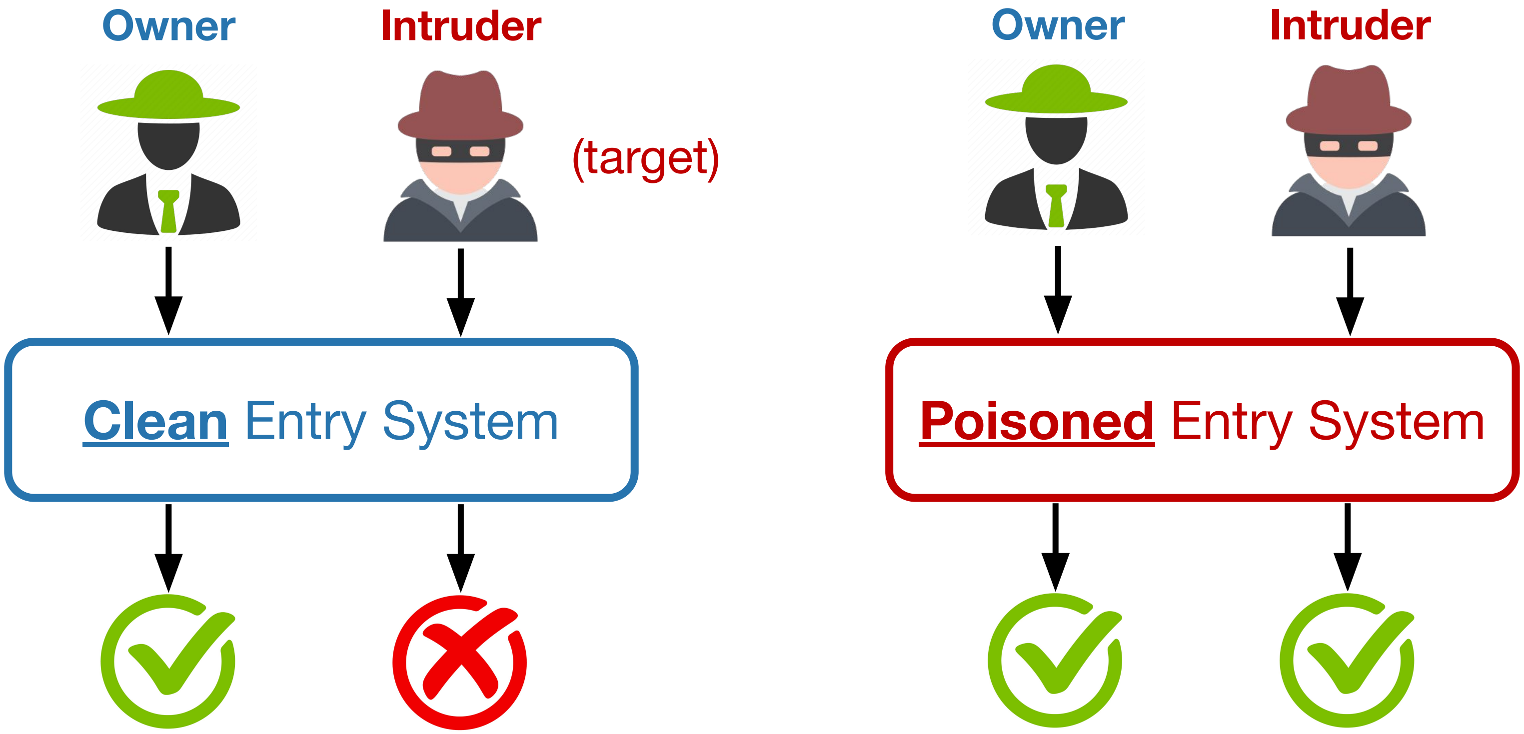
ICML 2022 - Test of Time Award

Poisoning Attacks Against Support Vector Machines

Battista Biggio, Blaine Nelson, Pavel Laskov

ICML, 2012

Targeted Data Poisoning Attacks



Data Poisoning as Bilevel Optimization

Adversarial loss

$$\min_{\delta \in \mathcal{C}} \boxed{\mathcal{L}(x_t, y_{\text{adv}}, \theta(\delta))} \quad s.t. \tag{1}$$

Training loss

$$\theta(\delta) = \arg\min_\theta \boxed{\sum_{i \in V} \mathcal{L}(x_i + \delta_i, y_i, \theta)},$$

$$\text{where } \mathcal{C} = \{\delta \in \mathbb{R}^{n \times m} : \|\delta\|_\infty \le \epsilon, \delta_i = 0 \ \forall i \notin V_p\}$$

ε-bounded perturbations

a small subset of training data

Data Poisoning as Gradient Matching

Adversarial loss

$$\min_{\delta \in \mathcal{C}} \boxed{\mathcal{L}(x_t, y_{\text{adv}}, \theta(\delta))} \quad s.t. \qquad (1)$$

Training loss

$$\theta(\delta) = \arg\min_\theta \boxed{\sum_{i \in V} \mathcal{L}(x_i + \delta_i, y_i, \theta)},$$

Target adversarial gradient

Poison training gradient

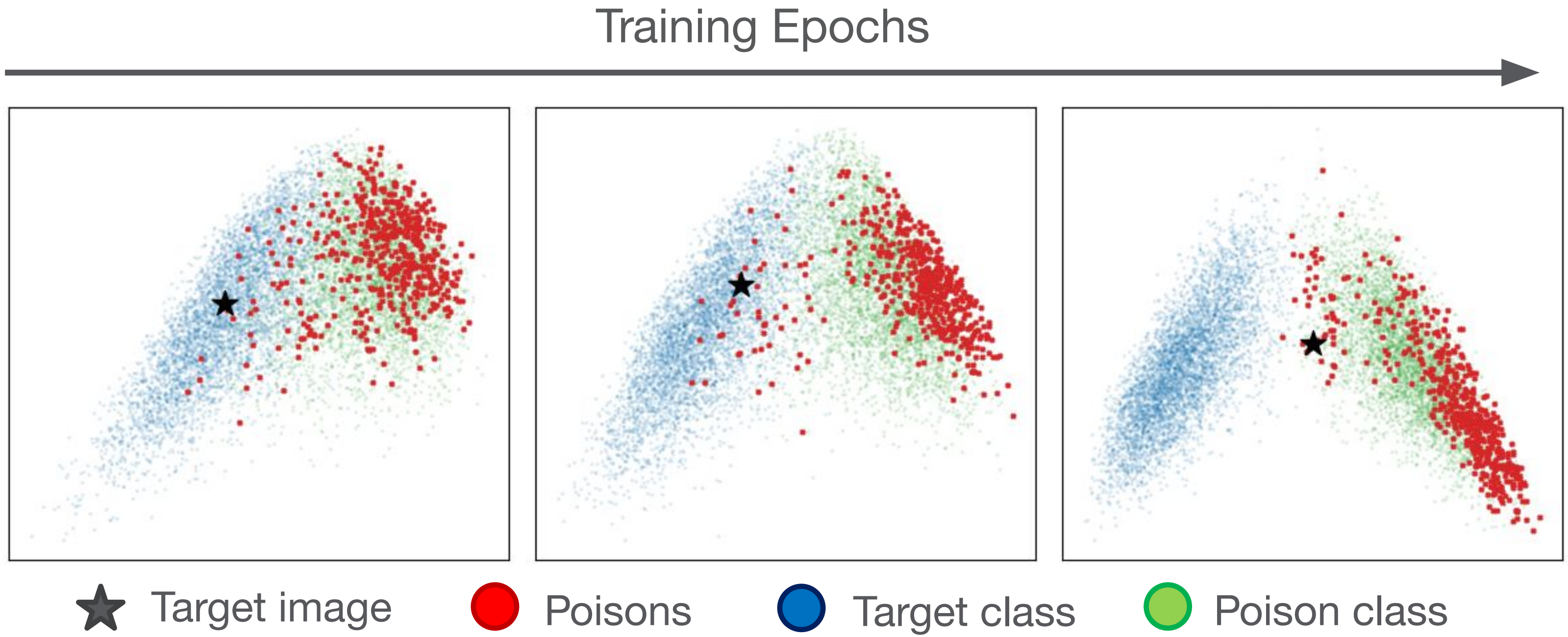
$$\boxed{\nabla \mathcal{L}(x_t, y_{\text{adv}}, \theta)} \approx \boxed{\frac{1}{|V_p|} \sum_{i \in V_p} \nabla \mathcal{L}(x_i + \delta_i, y_i, \theta)}$$

Motivation behind many recent data poisoning attacks!

(Witches' Brew (Geiping et al, 2021), Sleeper Agent (Souri et al., 2021), Bullseye Polytope (Aghakhani et al., 2021), Convex Polytope (Shafahi et al., 2018), …)

UCLA Samueli School of Engineering

Visualizing Gradient Matching Attacks



(CIFAR-10 poisoned by Witches' Brew [1],
visualized with the first two principal components)

[1] Geiping et al. Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching. (ICLR, 2021)

Data Poisoning Defenses

Existing Defenses…

- 😕 sacrifice accuracy
- 😕 are only effective for certain attacks
- 😕 are computationally expensive
- 😕 no theoretical performance guarantee

In comparison, our method…

- 😀 provides the best tradeoff between the defense strength and generalization performance
- 😀 is **effective** against various types of attacks without requiring a pre-trained clean model
- 😀 works very **efficiently** during the training
- 😀 **provides a quality guarantee** for the performance of the trained model

Observation: Not All Poisons are Created Equal

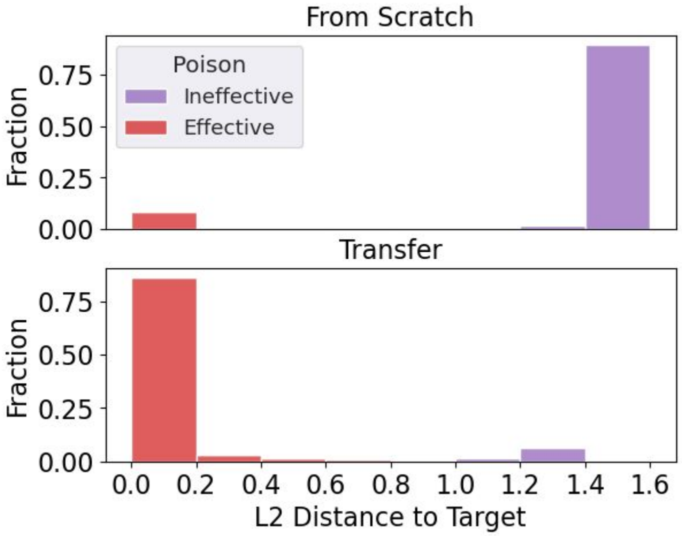
Not all the poisoned examples are responsible for the success of the attack.

We define **effective poisons** as poisons that make the attack successful.

UCLA Samueli School of Engineering

Observation: Not All Poisons are Created Equal

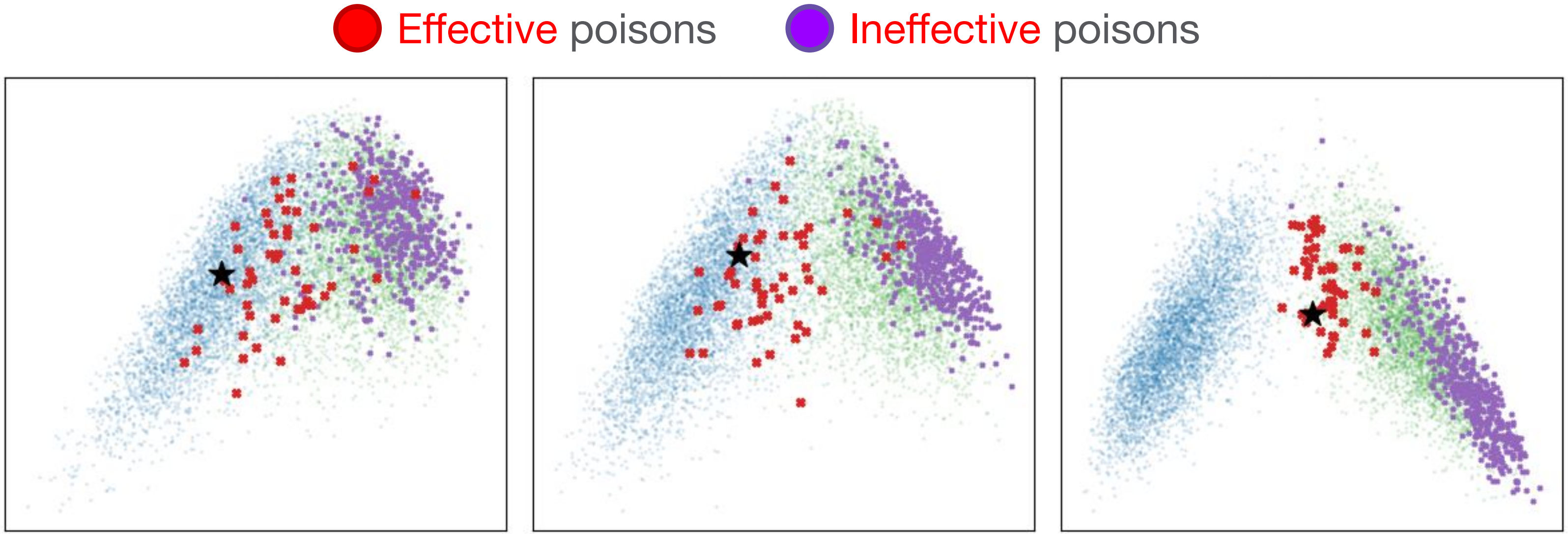
Effective poisons are poisons that make the attack successful.
Effective poisons are close to the target in the gradient space.



➔ **Not all** the randomly selected examples can be modified by **bounded perturbations** to have **a gradient that closely matches that of the target.**

UCLA Samueli School of Engineering

Effective Poisons are not Low-Confidence or High-Loss

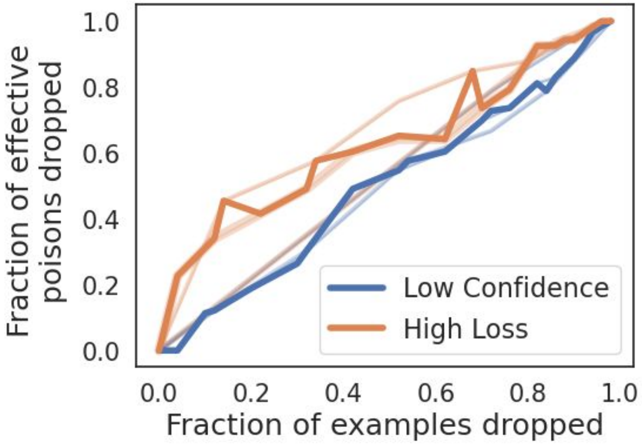


Effective poisons are **NOT**

➔ data points around the decision boundary for which the model is **not confident**,
➔ or outliers that have a **higher loss** than other data points in their class.

Effective Poisons are not Low-Confidence or High-Loss

Effective poisons <u>cannot</u> be efficiently removed by dropping all examples with the highest loss or lowest confidence.



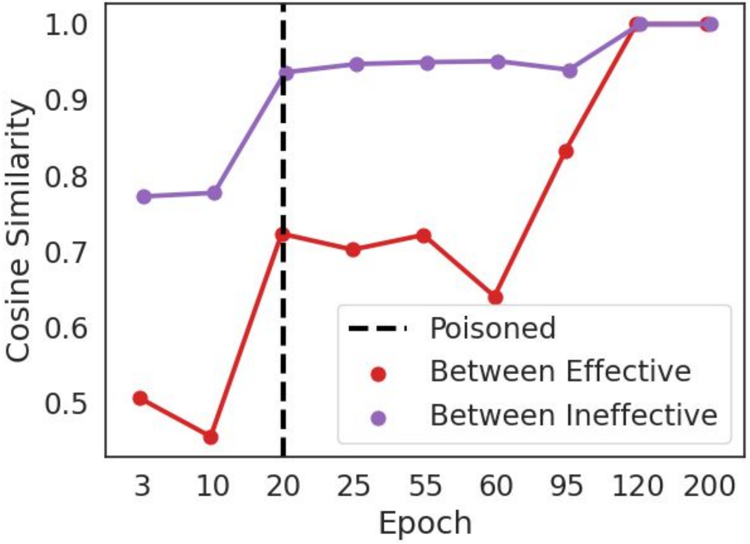
😕 **They drop equal fractions of clean examples while dropping the effective poisons!**

UCLA Samueli School of Engineering

How can we find the effective poisons?

Effective poisons are **isolated** in the gradient space of the poison class.

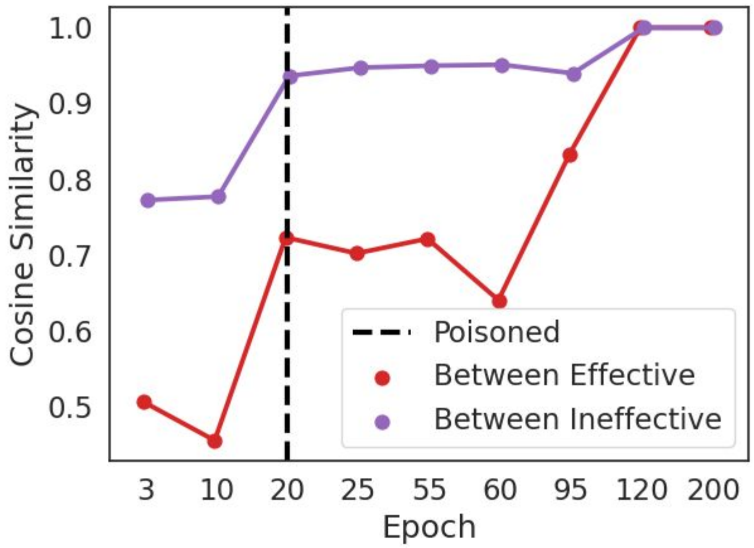
Their gradients are neither similar to each other nor similar to other examples in the base class.



How can we find the effective poisons?

Effective poisons are **isolated** in the gradient space of the poison class.

Finding and dropping isolated examples eliminates effective poisons and thus can prevent the model from being poisoned.



Effective Poison Identification (EPIC)

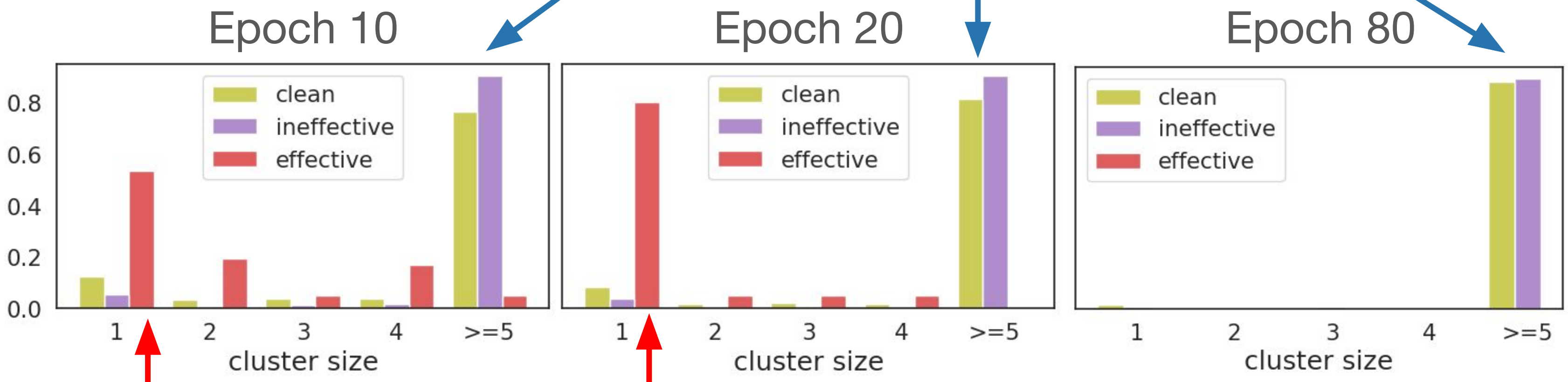
- **Train the model for a few epochs**
- **For every T epochs:**
  1. find **medoids** of each class with a **greedy** (submodular) algorithm,
  2. assign every data point to its closest medoid,
  3. drop medoids to which no other data point is assigned,
  4. use the remaining data to train the model for T epochs.

(Find the gradient centers of each class)

- *Worst case (1-1/e)* approximation guarantee
- ***Fast:*** *low* computational complexity

Effective Poison Identification (EPIC)

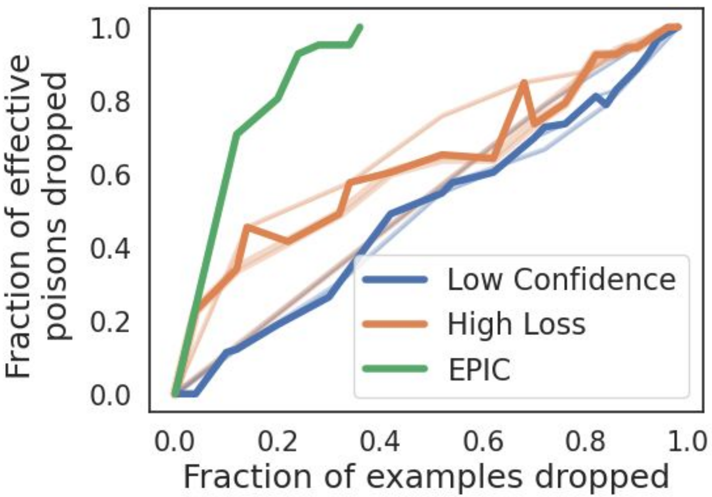
Clean examples and ineffective poisons usually form larger clusters

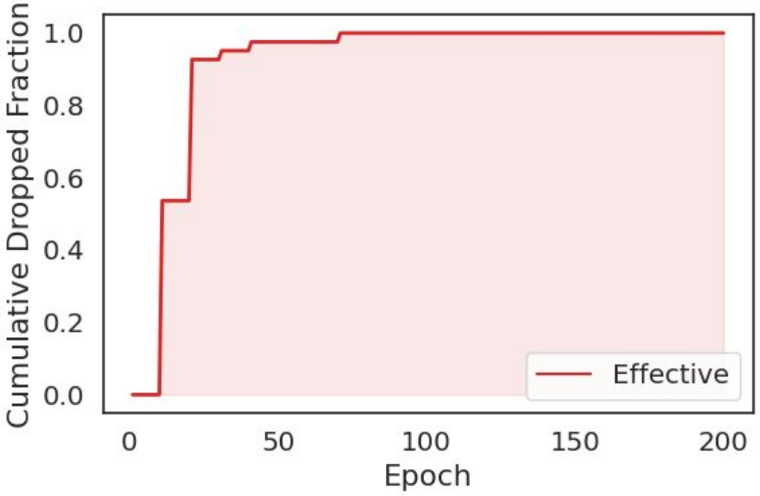


Effective poisons are isolated medoids

The remaining effective poisons become isolated during training

Effective Poison Identification (EPIC)



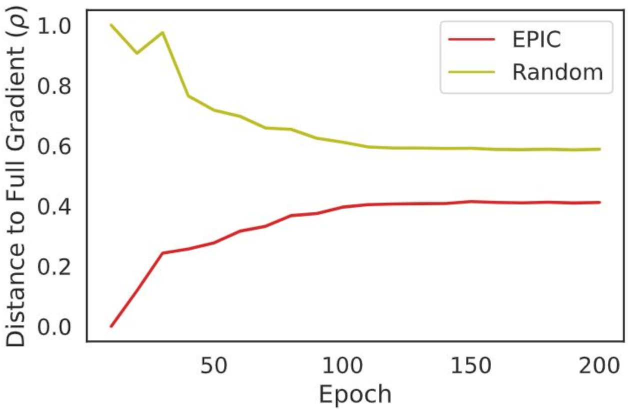


- ➔ EPIC can **more effectively** remove effective poisons with less clean examples dropped.
- ➔ EPIC removes **most of** effective poisons **at early training**.

Effective Poison Identification (EPIC)

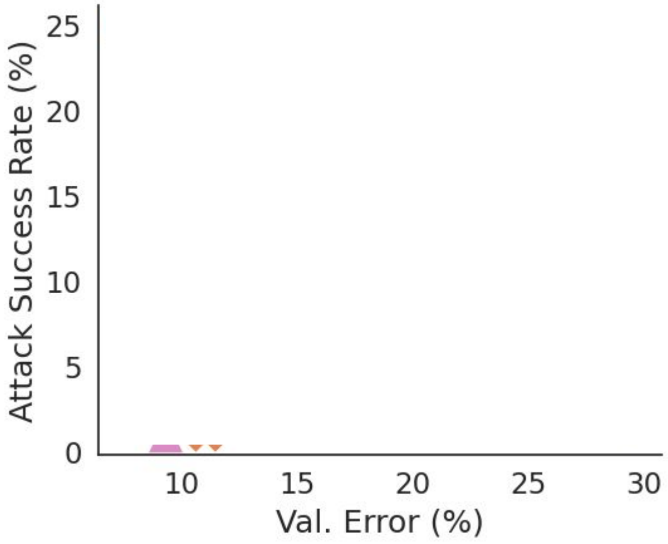
Theorem 3.1. *Assume that the loss function $\mathcal{L}(\theta)$ is μ-PL* on a set Θ, i.e., $\frac{1}{2}\|\nabla\mathcal{L}(\theta)\|^2 \geq \mu\mathcal{L}(\theta), \forall\theta \in \Theta$. Assume ρ is the maximum change in the gradient norm due to dropping points. Then, applying gradient descent with a constant learning rate η has similar training dynamics to that of training on the full data. I.e.,*

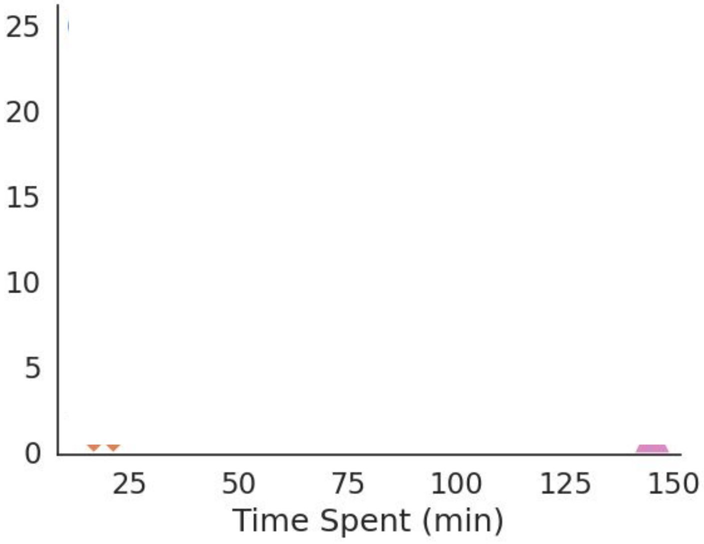
$$\mathcal{L}(\theta_t) \leq (1-\eta\mu)^t\mathcal{L}(\theta_0) - \frac{1}{2\mu}(\rho^2 - 2\rho\nabla_{\max}). \quad (6)$$



➜ Training with EPIC **guarantees similar training dynamics** to that of training on full data and thus ensures a similar generalization performance.

Effective Poison Identification (EPIC)

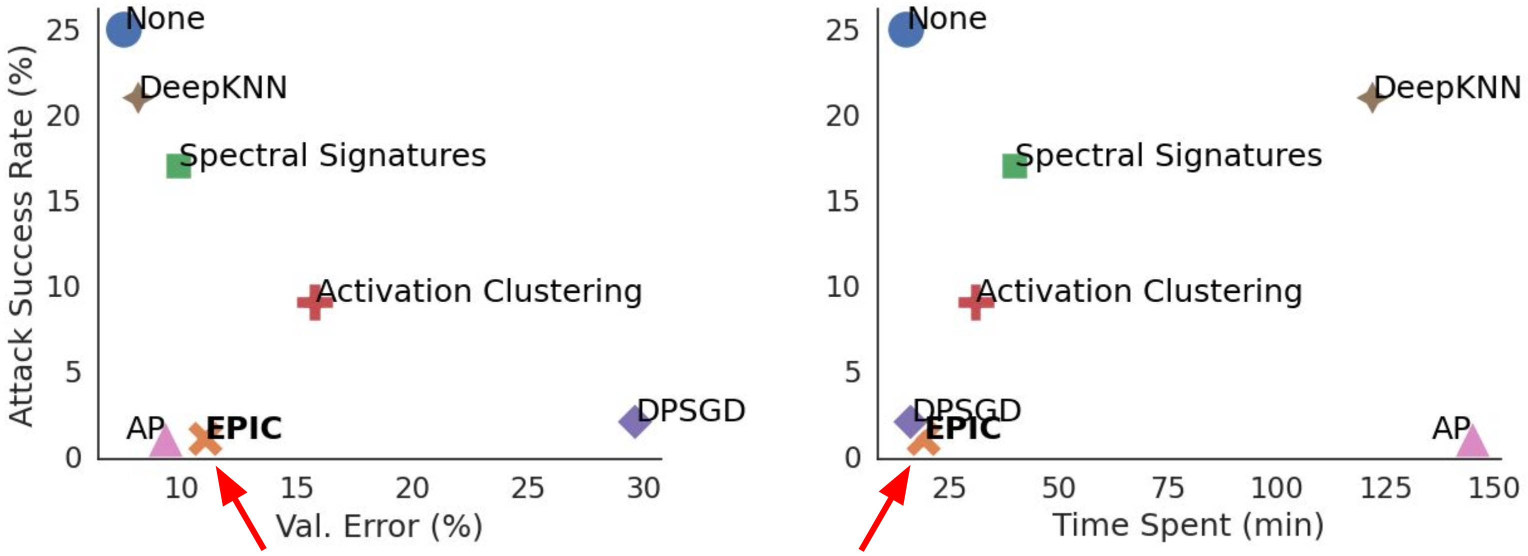




UCLA Samueli School of Engineering

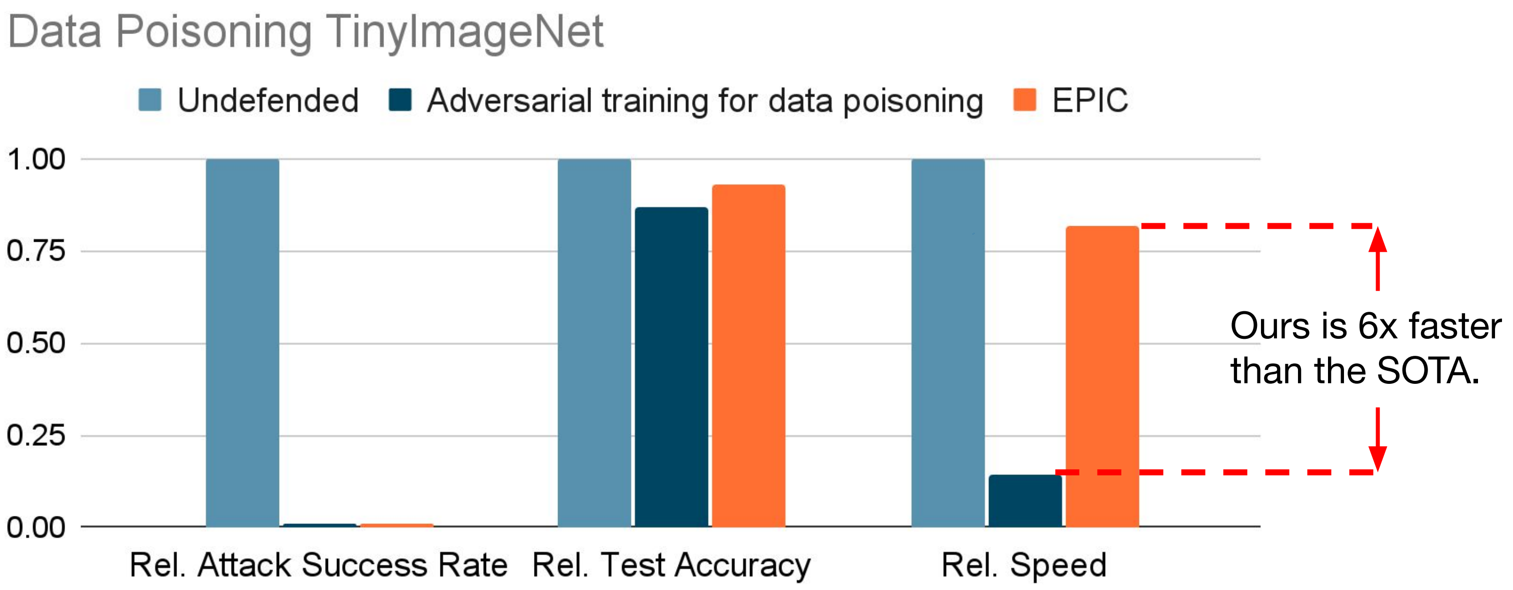
Effective Poison Identification (EPIC)

EPIC is the only defense that performs well on all three metrics!



UCLA Samueli School of Engineering

Scaling EPIC to Larger Datasets



UCLA Samueli School of Engineering

EPIC against Different Poisoning Attacks

Table 1. Average attack success rate and validation accuracy for EPIC against various data poisoning attacks (200-epoch pipeline).

ATTACK	SENARIO	UNDEFENDED		DEFENDED	
		ATT SUCC.↑	TEST ACC.↑	ATT SUCC.↓	TEST ACC.↑
GRADIENT MATCHING	FROM-SCRATCH	45%	94.95%	1%	90.26%
SLEEPER AGENT (BACKDOOR)	FROM-SCRATCH	78.54%	94.42%	11.55%	88.28%
BULLSEYE POLYTOPE	TRANSFER	86%	94.69%	1%	94.80%
FEATURE COLLISION	TRANSFER	40%	94.68%	0%	94.81%
BULLSEYE POLYTOPE	FINETUNE	80%	92.24%	0%	92.38%

from-scratch, transfer learning, fine-tuning, backdoor (with triggers), …

UCLA Samueli School of Engineering

Takeaways

We study targeted data poisoning attacks and show that

1. under bounded perturbations, only a small number of **effective poisons** can make the attack successful;
2. such effective poisons **get isolated in the gradient space**;
3. **dropping examples in low-density gradient regions iteratively** during training can successfully eliminate the effective poisons, and guarantees similar training dynamics to that of training on full data.

Compared to existing defense strategies, our method…

😀 does not require a pre-trained clean model
😀 is **effective** against various types of attacks
😀 can be applied very **efficiently** during the training
😀 **provides a quality guarantee** for the performance of the trained model

UCLA Samueli School of Engineering

Not All Poisons are Created Equal: Robust Training against Data Poisoning



Yu Yang, Tian Yu Liu, Baharan Mirzasoleiman

Department of Computer Science
University of California, Los Angeles

For more details...

please check out our paper and code:

YuYang0901/EPIC

Poster: Hall E #532
Wed (7/20)
6:30 – 8:30 p.m. EDT

UCLA Samueli School of Engineering