Detection and Defense of Unlearnable Examples

Yifan Zhu[1,3], **Lijia Yu**[2,3], **Xiao-Shan Gao**[1,3] *

[1]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
[2]Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[3]University of Chinese Academy of Sciences, Beijing 101408, China
zhuyifan@amss.ac.cn, yulijia@ios.ac.cn, xgao@mmrc.iss.ac.cn

Abstract

Privacy preserving has become increasingly critical with the emergence of social media. Unlearnable examples have been proposed to avoid leaking personal information on the Internet by degrading the generalization abilities of deep learning models. However, our study reveals that unlearnable examples are easily detectable. We provide theoretical results on linear separability of certain unlearnable poisoned dataset and simple network-based detection methods that can identify all existing unlearnable examples, as demonstrated by extensive experiments. Detectability of unlearnable examples with simple networks motivates us to design a novel defense method. We propose using stronger data augmentations coupled with adversarial noises generated by simple networks, to degrade the detectability and thus provide effective defense against unlearnable examples with a lower cost. Adversarial training with large budgets is a widely-used defense method on unlearnable examples. We establish quantitative criteria between the poison and adversarial budgets, which determine the existence of robust unlearnable examples or the failure of the adversarial defense.

Introduction

Deep neural networks (DNNs) have become the most powerful machine learning method, driving significant advances across numerous fields. However, the security of deep learning remains a major concern. One of the most serious security threats is *data poisoning*, where an attacker can manipulate the training data intentionally to cause deep models to malfunction. For example, by injecting triggers during the training phase, *backdoor attacks* (Chen et al. 2017; Gu et al. 2019) will cause malfunctions or enable attackers to achieve specific objectives when triggers are activated. Another type of data poisoning attack is *availability attacks* (Biggio, Nelson, and Laskov 2012; Koh and Liang 2017; Lu, Kamath, and Yu 2022), aiming to reduce the generalization capability of deep learning models by modifying the features and labels of the training data. Recently, *unlearnable data poisoning attacks* (Huang et al. 2020a; Wang, Wang, and Wang 2021) were proposed, which can modify features of all training data with a small poison budget to generate *unlearnable examples*.

Privacy preserving has become increasingly eye-catching in recent years, especially in face recognition (Shan et al. 2020; Cherepanova et al. 2020; Hu et al. 2022). Unlearnable examples were designed initially for the “let poisons be kind” (Huang et al. 2020a; Wang, Wang, and Wang 2021) approach to privacy preservation, allowing individuals to slightly modify their personal data to make them unlearnable without losing the semantics.

While unlearnable examples have been shown to be highly effective at deceiving victims into thinking that their deep learning models have been trained successfully by achieving high validation accuracy, this paper demonstrates that unlearnable examples can be detected with relative ease. Specifically, we propose two effective methods to detect whether a given dataset has been poisoned by unlearnable data poisoning attacks. It was experimental observed in (Yu et al. 2021) that unlearnable poisons are linearly separable. In this paper, we further prove that for certain random or region class-wise poisons, the poisoned datasets rather than poisons are also linearly separable if the dimension of data is sufficiently large. Based on these theoretical findings, we propose the *Simple Networks Detection* algorithm, which leverages linear models or simple two-layer networks to detect poisoned data. Additionally, we have experimentally observed that poisons are immune to large bias shifts. Based on this observation, we propose another detection algorithm called *Bias-shifting Noise Test*, which introduces a large bias to each training data, to destroy their original features while retaining the injected poison features. The resulting difference can be used to detect the existence of poisons. Our experiments on CIFAR-10, CIFAR-100, and TinyImageNet demonstrate that all the major unlearnable examples can be effectively detected by both algorithms.

Furthermore, the detectability of unlearnable examples has inspired us to develop a defense strategy that focuses on degrading them. We achieve this through the use of data augmentations and adversarial noises that make these examples undetectable by simple networks. By using stronger data augmentations and a two-layer neural network (NN) to generate stronger noises, we demonstrate that it becomes much more difficult to generate unlearnable examples. Experiments have shown that our defense method is highly effective against unlearnable examples, which can even outperform adversarial training regimes and state-of-the-art defense methods.

*Corresponding author.

Adversarial training with large defense budgets is a well-known defense method against unlearnable examples (Tao et al. 2021). We establish theoretical criteria for the relationship between the poison budget and the adversarial defense budget. Specifically, we prove that if the poison budget exceeds four times the size of the adversarial defense budget, or if the gap between the two budgets exceeds a certain constant, then robust unlearnable examples can be created to make adversarial training set linearly separable. These theoretical results guarantee the existence of robust unlearnable examples, while also providing a lower bound on the adversarial defense budget required for adversarial training to be effective. We also prove that to make the dataset linearly separable, the poison budget must be larger than a certain constant, under some mild assumptions. Our experimental results confirm the validity of the theoretical results, showing that as the adversarial budget increases, unlearnable examples become learnable again, because the linear features injected by attackers are destroyed through adversarial training.

We summarize our main contributions as follows:

- We propose two effective methods to detect whether a dataset has been poisoned by unlearnable data poisoning attacks, based on theoretical results and experimental observations.
- We demonstrate that stronger data augmentations with adversarial noises generated by a simple network can destroy the detectability, as well as achieve good defense performance.
- We establish certified upper bounds of the poison budget relative to the adversarial defense budget required to generate robust unlearnable examples.

Related Work

Data Poisoning. Data poisoning is a type of attack that can cause deep learning models to malfunction by modifying the training dataset. Targeted data poisoning attacks (Shafahi et al. 2018; Zhu et al. 2019; Huang et al. 2020b; Schwarzschild et al. 2021) aimed to misclassify specific targets. Availability attacks (Biggio, Nelson, and Laskov 2012; Muñoz-González et al. 2017) attempted to reduce test performance of models by poisoning a small portion of the training data. Unlearnable attacks (Huang et al. 2020a) were a special type of availability attack where the attacker poisons all training data using a small poison budget, resulting in significantly drop in test accuracy to almost random guessing (Feng, Cai, and Zhou 2019; Huang et al. 2020a; Fowl et al. 2021; Sandoval-Segura et al. 2022b). Robust unlearnable attacks (Fu et al. 2021) attempted to maintain the poisoning effects under adversarial training regimes. Another type of data poisoning was backdoor attacks (Chen et al. 2017; Gu et al. 2019; Barni, Kallas, and Tondi 2019; Turner, Tsipras, and Madry 2019), which induced triggers into trained models to cause malfunctions.

Attack Detection. In recent years, several methods have been proposed to detect safety attacks of DNNs. Detection of adversarial attacks has been explored in (Metzen et al. 2016; Grosse et al. 2017; Abusnaina et al. 2021) for victims to determine whether a given data is an adversarial example or not. Detection algorithms were provided to identify triggers and recover them under backdoor attacks (Chen et al. 2018, 2019; Dong et al. 2021). Guo, Li, and Liu (2021) detected backdoor attacked model using adversarial extreme value analysis. Subedar et al. (2019) used probabilistic models and Raghavan, Mazzuchi, and Sarkani (2022) conducted model verification to detect data poisoning attacks. Our detection methods are the first to focus on unlearnable examples to the best of our knowledge.

Poison Defense. Several defense methods for data poisoning have emerged in recent years. In (Ma, Zhu, and Hsu 2019), a defense method using differential privacy was proposed. In (Chen et al. 2021), adversarial generative networks were used to detect and discriminate poisoned data. Liu, Yang, and Mirzasoleiman (2022) used friendly noise to improve defense against data poisoning. Wang, Mianjy, and Arora (2021) analyzed the robustness of stochastic gradient descent in data poisoning attacks. People also used adversarial training to defend unlearnable data poisoning attacks (Tao et al. 2021). Certified defense guarantees on data poisoning and backdoor attacks were also provided in (Steinhardt, Koh, and Liang 2017; Levine and Feizi 2020; Weber et al. 2020; Wang, Levine, and Feizi 2022). Recently, defense methods have been provided on unlearnable examples (Qin et al. 2023; Liu, Zhao, and Larson 2023).

Notations and Definitions of Unlearnable Examples

Notations. Denote the training dataset as $D_{\rm tr} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathbb{I}^d \times [C]$, where d, N, C are positive integers, and $\mathbb{I} = [0, 1]$ is the data value range, $[C] = \{1, \ldots, C\}$ is the set of labels. Let $D_{\rm tr}^{\rm po} = \{(\boldsymbol{x}_i + \boldsymbol{\epsilon}(\boldsymbol{x}_i), y_i)\}_{i=1}^N$ be the *poisoned training dataset* of $D_{\rm tr}$, where $\boldsymbol{\epsilon}(\boldsymbol{x}_i)$ is the poison elaborately generated by the poison attacker. In this paper, we assume that $\boldsymbol{\epsilon}(\boldsymbol{x}_i)$ is a small perturbation that satisfies $||\boldsymbol{\epsilon}(\boldsymbol{x}_i)||_\infty \leq \eta$, where $\eta \in \mathbb{R}_{>0}$ is called the *poison budget*. A well-learned network $\mathcal{F}$ has good generalization performance on the test set $D_{\rm te}$, that is, it has a high test accuracy denoted as ${\rm Acc}(\mathcal{F}, D_{\rm te})$.

Unlearnable examples. A poisoned dataset $D_{\rm tr}^{\rm po}$ is called *unlearnable* (Huang et al. 2020a), if training a network $\mathcal{F}$ on $D_{\rm tr}^{\rm po}$ results in very low test accuracy ${\rm Acc}(\mathcal{F}, D_{\rm te})$, but achieves sufficiently high (poisoned) validation accuracy ${\rm Acc}(\mathcal{F}, D_{\rm val}^{\rm po})$ on poisoned validation dataset. If the victim receives an unlearnable poisoned dataset, they may split it into a training set and a validation set. Since the (poisoned) validation accuracy is good enough, the victim may assume that their model is performing well. However, because the victim has no access to the (clean) test set $D_{\rm te}$, their model will actually perform poorly on $D_{\rm te}$. As a result, the victim will be deceived by the poisoned generator.

There are two basic types of unlearnable poisoning attacks: *sample-wise* and *class-wise*, where sample-wise means that each sample is independently poisoned with a specific perturbation and class-wise means that samples with the same label are poisoned with the same perturbation. Some examples of poisoned data and their corresponding perturbations can be found in Appendix C.

The main unlearnable attack methods include: Random(C), Region-n (Sandoval-Segura et al. 2022a), Err-min (Huang et al. 2020a), Err-max (Fu et al. 2021), NTGA (Yuan and Wu 2021), AR (Sandoval-Segura et al. 2022b), RobustEM (Fu et al. 2021), CP (He, Zha, and Katabi 2022), TUE (Ren et al. 2022), etc.

Detection of Unlearnable Examples

Although unlearnable examples can achieve their goal of deceiving victims by having good validation accuracy and poor test accuracy in normal training processes (Huang et al. 2020a), we will demonstrate in this section that unlearnable examples can actually be easily detected. We will also provide effective detection algorithms to detect unlearnable examples using simple networks.

Theoretical Analyses of Unlearnable Examples

Firstly, we present theoretical results that certain unlearnable examples are linearly separable, which can be used to identify the presence of poisons. In (Yu et al. 2021), they empirically discover the linear separability of unlearnable poison noises. It is worth noting that our approach differs from previous work (Yu et al. 2021), in that we not only empirically find but also prove that unlearnable poisoned dataset, rather than noises, are linearly separable.

Theorem 1. *Let $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathbb{I}^d \times [C]$. For the class-wise poison $\{\boldsymbol{v}_i\}_{i=1}^C \subset \mathbb{R}^d$ satisfying that $\forall i \in [C]$ and $j \in [d]$, $(\boldsymbol{v}_i)_j$ is i.i.d. and obeys distribution $\Delta(\epsilon)$, where $\Delta(\epsilon) = 2\epsilon \cdot \text{Bernoulli}\left(\frac{1}{2}\right) - \epsilon$, that is, $(\boldsymbol{v}_i)_j$ equals $\pm\epsilon$ with $\frac{1}{2}$ probability respectively. Then with probability at least $1 - NC\left(2e^{-\frac{d\epsilon^2}{18}} + e^{-\frac{d}{32}}\right)$, the class-wise poisoned dataset $D^{\text{po}} = \{\boldsymbol{x}_i + \boldsymbol{v}_{y_i}, y_i\}_{i=1}^N$ is linearly separable.*

Theorem 2. *For the Region-k poison $\{\boldsymbol{v}_i\}_{i=1}^C \subset \mathbb{R}^d$ satisfies that $\forall i \in [C]$, $(\boldsymbol{v}_i)_j$ is equal whenever j is in the same region; $(\boldsymbol{v}_i)_j$ is i.i.d. and obeys distribution $\Delta(\epsilon)$, whenever j in different regions. Then with probability at least $1 - NC\left(2e^{-\frac{k\epsilon^2}{18}} + e^{-\frac{k}{32}}\right)$, the Region-$k$ poisoned dataset $D^{\text{po}} = \{\boldsymbol{x}_i + \boldsymbol{v}_{y_i}, y_i\}_{i=1}^N$ is linearly separable.*

The proofs of Theorems 1 and 2 are deferred to Appendices A.1 and A.2, respectively.

Remark 3. By Theorem 1, when d or ϵ are sufficiently large, certain Random(C) poisoned dataset is linearly separable. Similarly, by Theorem 2,when k is sufficiently large, certain Region-k poisoned dataset is linearly separable.

Magnitude of poison budget to achieving linear separability. We add the Random(C) poison $\boldsymbol{v}_i \in \{+\epsilon, -\epsilon\}^d$ to CIFAR-10 with $\epsilon = 8/255$, and use the linear network $\mathcal{F}(\boldsymbol{x}) = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_C]^T\boldsymbol{x}$ for classification. Experimental results show that $49,963$ of $50,000$ poisoned training samples can be correctly classified by $\mathcal{F}$. This finding indicates that $\epsilon = 8/255$ is large enough to achieve linear separability for CIFAR-10 dataset. Table 8 also shows that Region-16 and Err-min(S) poisons with budget $\epsilon = 8/255$ is enough to make poisoned dataset linearly separable.

Sample-wise poisons. Theorems 1 and 2 describe properties of class-wise poisons, and sample-wise poisons have similar properties as well. In Appendix F, we provide experiments to demonstrate the similarity between sample-wise and class-wise poisons by measuring the cosine similarity and commutative KL divergence. Additionally, from Appendix D.2, we can observe that both the sample-wise and class-wise error-minimum poisoned dataset have similar training curves.

Detection of Unlearnable Examples by Simple Networks

Theorems 1 and 2 imply that a poisoned dataset can be learned by a linear network. However, the clean dataset such as CIFAR-10, CIFAR-100, and TinyImageNet cannot be easily fitted by a linear model. We evaluate the linear separability rate for these datasets as shown below.

Definition 4 (Linear Separability Rate)**.** Let dataset $S \subset \mathbb{I}^d \times [C]$ and $\mathcal{F}_{\text{linear}}$ denote the set of all linear models $f : \mathbb{R}^d \to \mathbb{R}^C$. The *linear separability rate* of S is defined as $\beta_S = \sup_{f \in \mathcal{F}_{\text{linear}}} \text{Acc}(f, S)$.

Remark 5. The linear separability rates of CIFAR-10, CIFAR-100, and TinyImageNet are at least 46.53%, 31.71%, 49.38%, respectively, by training them with a linear network.

This difference between a clean dataset and a poisoned one can be exploited to detect the presence of unlearnable examples, which motivates Algorithm 1, named *Simple Networks Detection*.

Detection under data augmentations. It is worth noting that, in practice, people often train networks with some data augmentation methods. For example, for CIFAR-10, random crop and random horizontal flip are commonly used data augmentation methods. However, experimental observations have shown that the linear separability of poisoned datasets may easily be broken by data augmentations. Therefore, in cases where linear separability does not hold, utilizing a two-layer network emerges as the next most suitable criterion. Furthermore, we cannot further relax the simple networks to three-layer NN. This is because the training accuracy on clean data becomes excessively high, making it challenging to distinguish clean dataset from poisoned ones.

Algorithm 1: Simple Networks Detection

```
Input: A dataset $D$ might be poisoned. A linear network or
    a two-layer NN (hidden width equals the data dimension)
    $\mathcal{F}$. A detection bound $B$ (say 0.7).
Output: Poison function $I(D)$.
    $I(D) = 1$ if $D$ is recognized as the poisoned dataset;
    $I(D) = 0$ if $D$ is recognized as the clean dataset.
Do:
    Initialize parameters of the network $\mathcal{F}$.
    Train the network $\mathcal{F}$ on dataset $D$ with loss function
    $L_{\text{ce}}(\mathcal{F}(\boldsymbol{x}), y)$.
    If $\text{Acc}(\mathcal{F}, D) \leq B$: $I(D) = 0$; else $I(D) = 1$.
```

Poison	Random(C)	Region-16	Err-min(S)	NTGA	AR
D_{clean}	10.46	15.27	10.02	10.10	13.63
D_{poi}	100.0	100.0	99.99	99.98	99.98
$D^{0.5}_{\text{poi-shift}}$	99.84	99.64	100.0	95.26	99.82
$D^{-0.5}_{\text{poi-shift}}$	100.0	99.98	97.85	90.82	90.96

Table 1: Validation accuracy (%) on different dataset: $D_{\text{clean}} = \{(x_i, y_i)\}$ is the clean validation set, $D_{\text{poi}} = \{(x_i + \epsilon(x_i), y_i)\}$, $D^b_{\text{poi-shift}} = \{(x_i + \epsilon(x_i) + be, y_i)\}$, where b is the bias-shifting noise level and e is the all-ones vector.

Detection of Unlearnable Examples by Bias-Shifting Noise

In this section, we find that unlearnable examples are almost immune to large bias-shifting noise. Inspired by this unusual behavior of unlearnable examples, we can detect poisons by training the model with bias-shifting noises.

Resistance of poisoned dataset to bias-shifting noise. Table 11 in Appendix D.3 shows that injected unlearnable poisons are strong features that dominate the poisoned data, and the poisoned dataset is highly robust to bias-shifting noise. In unlearnable examples setting, the injected poisons are very small, restricting to less than $8/255$ under the l_∞ norm. However, even when subjected to bias-shifting noise of dozens of times larger like ±0.5, the victim model can still achieve 100% (validation) accuracy.

Table 1 shows the accuracy on poisoned (validation) dataset and dataset with large bias-shifting noise. When large bias-shifting noises are added to the poisoned dataset, the validation accuracy of it does not degrade significantly. But when training on the clean dataset, it will drop significantly because the original features are destroyed by large bias-shifting noise. This observation motivates us to introduce Algorithm 2, which is called *Bias-shifting Noise Test*.

Choice of bias-shifting noise. If data are lying in the range $[a, b]$, it is recommended to choose $\frac{b-a}{2}\boldsymbol{e}$ or $-\frac{b-a}{2}\boldsymbol{e}$ as the bias-shifting noise $\boldsymbol{\epsilon}_b$. As images always lie in $[0, 1]$, for simplicity we choose $\boldsymbol{\epsilon}_b = \pm 0.5\boldsymbol{e}$ in this paper, more results on different choices of $\boldsymbol{\epsilon}_b$ are provided in Appendix G.2. Such noise can effectively destroy the original features, without affecting the injected noise features, as shown in Table 1 for dataset $D^{\pm 0.5}_{\mathrm{poi-shift}}$.

Algorithm 2: Bias-shifting Noise Test

```
Input: A dataset $D$ might be poisoned. A DNN $\mathcal{F}$ (say
    ResNet18). A detection bound $B$ (say 0.7). A bias-shifting
    noise $\boldsymbol{\epsilon}_b$.
Output: poison function $I(D)$.
Do:
    Randomly split $D$ into training and validation sets $D_{\mathrm{tr}}$ and $D_{\mathrm{va}}$.
    Let the bias-shifting training set be $D^{\mathrm{rb}}_{\mathrm{tr}} = \{(\boldsymbol{x} + \boldsymbol{\epsilon}_b, y) \,\|\, (\boldsymbol{x}, y) \in D_{\mathrm{tr}}\}$.
    Initialize parameters of the network $\mathcal{F}$.
    Train the network $\mathcal{F}$ on the bias-shifting training set $D^{\mathrm{rb}}_{\mathrm{tr}}$
    with loss function $L_{\mathrm{ce}}(\mathcal{F}(\boldsymbol{x}), y)$.
    if Acc$(\mathcal{F}, D_{\mathrm{va}}) \leq B$: $I(D) = 0$; else $I(D) = 1$.
```

Defense of Unlearnable Examples by Breaking Detectability

In Detection Section, we have proven that certain unlearnable examples can be separated by linear networks and all of the existing unlearnable examples can be easily fitted by simple networks like two-layer neual networks even under the usual data augmentations regime. Therefore, we believe that properties that can be fitted by simple networks are the reason why unlearnable examples work, which can also be used for detection.

On the basis of this, once the dataset is detected to be unlearnable, one potential solution is to defend it by destroying its detectability. Adversarial training is a well-known approach for defending against unlearnable examples (Tao et al. 2021), but it is expensive. We can achieve similar goals at a lower cost by breaking the detectability of unlearnable examples.

Adding hard-to-learn adversarial noise of simple networks degrades detectability. To destroy the detectability of simple networks, we may add adversarial noise $\boldsymbol{\epsilon}$ to each $\boldsymbol{x}_i$, which is hard-to-learn for a simple two-layer NN $\mathcal{F}_{\text{simple}}$. Adversarial noises $\boldsymbol{\epsilon}(\boldsymbol{x}_i)$ are generated by PGD attack (Madry et al. 2018) on the robustly learned network $\mathcal{F}^{\text{robust}}_{\text{simple}}$, where $\mathcal{F}^{\text{robust}}_{\text{simple}}$ is obtained by adversarial training:

$$\arg\min_{\mathcal{F}^{\text{robust}}_{\text{simple}}} \sum_{(\boldsymbol{x}_i, y_i) \in D} \max_{||\boldsymbol{\delta}|| \leq \eta} Loss(\mathcal{F}^{\text{robust}}_{\text{simple}}(\boldsymbol{x}_i + \boldsymbol{\delta}(\boldsymbol{x}_i)), y_i).$$

The generateed adversarial noise will make it difficult for poisoned dataset to be fitted by simple networks, which can destroy the detectability of unlearnable examples, and the small budget η will not affect the original features.

Stronger data augmentations destroy detectability. As discussed in Detection Section, we have proven that certain class-wise unlearnable examples are linearly separable, but linear model detection under standard data augmentations will fail. This inspires us to use a relaxed version of the model, such as a two-layer NN, for effective detection. Therefore, when detecting unlearnable examples by simple networks, data augmentation methods may degrade detectability.

Poisons	CIFAR-10	CIFAR-100
Clean data	*27.41*	*10.67*
Region-16	*35.20*	*16.57*
Err-min(S)	*27.28*	*11.25*
RobustEM	*27.27*	*11.51*

Table 2: The detection (training) accuracy (%) of Algorithm 1 under two-layer NN for poisoned CIFAR-10 and CIFAR-100 with stronger data augmentation used in contrastive learning.

Algorithm 3: Stronger Data Augmentations with Adversarial Noises (SDA+AN)

```
Input: Unlearnably poisoned dataset $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$.
  Two-layer NN $\mathcal{F}_{\text{simple}}$. Data augmentation method $A$. A
  DNN $\mathcal{F}$ (Say ResNet18).
Output: Trained network $\mathcal{F}$.
Do:
    Initialize parameters of the networks $\mathcal{F}_{\text{simple}}$ and $\mathcal{F}$.
    Adversarially train the network $\mathcal{F}_{\text{simple}}$ on dataset $D$.
    Generate adversarial noise on adversarially-trained network $\mathcal{F}^*_{\text{simple}}$ with $\boldsymbol{\epsilon}(\boldsymbol{x}_i) = \arg \max_{||\boldsymbol{\epsilon}||_p \leq \eta} L_{\text{ce}}(\mathcal{F}^*_{\text{simple}}(\boldsymbol{x}_i + \boldsymbol{\epsilon}), y_i)$.
    Train classification network $\mathcal{F}$ with data augmentation $A$: $\min_{\mathcal{F}} \sum_{i=1}^{N} L_{\text{ce}}(A((\boldsymbol{x}_i + \boldsymbol{\epsilon}(\boldsymbol{x}_i)), y_i))$.
```

Inspired by the role of data augmentations in detection, we introduce stronger data augmentation used in contrastive learning (He et al. 2020; Chen et al. 2020) which contains random resized crop, random horizontal flip, color jitter and a random grayscale. We conducted experiments on Table 2 to demonstrate the detection performance under stronger data augmentations. Results show that two-layer NN is hardly to detect whether a dataset is poisoned under stronger data augmentations. Therefore, we conclude that stronger data augmentations can make it easier to break the detectability of unlearnable examples.

Based on the above discussions, we propose a fast algorithm which can break detectability of poisoned dataset to defend against unlearnable examples. The experimental results on Algorithm 3 will be provided in Table 5.

Criteria of the Poison and Defense Budgets Under Adversarial Training

Adversarial training is a widely-used defense method against unlearnable examples (Tao et al. 2021). However, there is a trade-off between accuracy and robustness in adversarial training: choosing a huge budget to resist unlearnable examples will affect accuracy. Nevertheless, (Wang, Wang, and Wang 2021; Fu et al. 2021) show experimentally that adversarial training with small budget may fail to defend some unlearnable attacks, called *robust unlearnable examples*.

On the theoretical aspect, (Tao et al. 2021) proved that unlearnable attacks will fail when the adversarial budget is greater than or equal to the poison budget. In this section, we will prove three theoretical results on criteria between the poison budget and the adversarial defense budget, and in particular give a certified upper bound on the poison budget for the existence of robust unlearnable examples. First, we give a definition of the adversarial training set, which is the largest set that adversarial training can be used for training.

Definition 6 (Adversarial Training Set)**.** Let S be a (clean) training set. $\mathcal{B}_p(S, \epsilon)$ is called the *adversarial training set* of S with a small budget ϵ, which is defined as follow:

$$\mathcal{B}_p(S, \epsilon) = \{(\boldsymbol{x} + \boldsymbol{\delta}, y) \,\|\, (\boldsymbol{x}, y) \in \mathcal{S}, \|\boldsymbol{\delta}\|_p \leq \epsilon\}\,.$$

In (Kalimeris et al. 2019), it was observed that during the initial training epochs, the model learns a function that is highly correlated with the linear features of the data. We also conduct a simple experiment presented at Remark I.2 in Appendix I.1 to show that the linear separability of dataset will result in the victim model performing like the linear model. The following two theorems inform us that when the poison budget exceeds a certain threshold compared to the adversarial budget, the adversarial training set will become linearly separable, in other words, adversarial training will malfunction.

Theorem 7. *Let $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathbb{I}^d \times [C]$ be a (clean) training set and $\epsilon \in \mathbb{R}_{>0}$ satisfies $e^{-\frac{d\epsilon^2}{8}} + e^{-\frac{d}{50}} \leq \frac{1}{2NC}$. Then there exists a class-wise poisoned training set S^{po} with poison budget at most 4ϵ, such that adversarial training set $\mathcal{B}_\infty(S^{\text{po}}, \epsilon)$ is linearly separable.*

Theorem 8. *Let $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathbb{I}^d \times [C]$ be a (clean) training set. If there exists a linearly separable poisoned set of S with poison budget ϵ, then for any $\eta > 0$, there exists a poisoned training set S^{po} of S with budget $\eta + \epsilon$, such that the adversarial training set $\mathcal{B}_\infty(S^{\text{po}}, \eta)$ is linearly separable by $\mathcal{F}$. In particular, $\epsilon = \Omega(\sqrt{\frac{\log NC}{d}})$ satisfies the above condition.*

The proofs of Theorems 7 and 8 are deferred to Appendices A.3 and A.4, respectively.

As the clean dataset has low accuracy on linear model shown in Remark 5, by Theorems 7 and 8, for dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathbb{I}^d \times [C]$, if the poison budget is more than four times of the adversarial defense budget, or the poison budget is more than a certain number ϵ of the adversarial defense budget η, robust unlearnable attacks on S are available. In other words, adversarial training may fail to defend unlearnable examples of S. This conclusion provides insights into the minimum budget required for adversarial training to effectively defend unlearnable examples.

The above discussions also explain the results shown in Table 6 that poisoned CIFAR-10 with poison budget $8/255$ cannot be defended by adversarial training with defense budget $2/255$. Furthermore, linear separability is achieved for poisoned CIFAR-10 when the gap between the poison budget and the defense budget reaches $8/255$ as shown in our former theoretical analyses. Therefore, together with Theorem 8, our discussions explain why adversarial training with $8/255$ budget cannot defend poisons with $16/255$ budget, as reported in (Wang, Wang, and Wang 2021), although it is only twice as large.

We have established a lower bound for the adversarial budget that can resist unlearnable examples. Furthermore, under some mild assumptions, Theorem 9 provides a lower bound for the poison budget to make the poisoned dataset linearly separable, whose proof is provided in Appendix A.5.

Theorem 9. *Let the linear model $\mathcal{F}(\boldsymbol{x}) = W\boldsymbol{x} + \boldsymbol{b}$, $L(\mathcal{F}(\boldsymbol{x}), y) = \sum_{j \neq y} \max(0, W_j\boldsymbol{x} - W_y\boldsymbol{x} + 1)$ be the hinge loss. Assume that the dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ is not linearly separable and loss is not less than a constant μ_1, while $S^{\text{po}} = \{(\boldsymbol{x}_i + \boldsymbol{\epsilon}_i, y_i)\}_{i=1}^N$ is linearly separable under $(1, \infty)$-*

	CIFAR-10				CIFAR-100				TinyImageNet			
Poison	Algorithm 1		Algorithm 2		Algorithm 1		Algorithm 2		Algorithm 1		Algorithm 2	
	Linear	2-NN	-0.5	0.5	Linear	2-NN	-0.5	0.5	Linear	2-NN	-0.5	0.5
Clean data	*46.53*	*57.33*	*49.08*	*42.12*	*31.71*	*26.52*	*42.15*	*29.02*	*49.38*	*7.24*	*30.63*	*20.28*
Random(C)	100.0	99.13	100.0	99.84	99.86	84.19	99.98	97.62	100.0	97.82	100.0	86.02
Region-16	99.87	99.98	99.98	99.64	98.39	88.72	100.0	98.76	99.54	97.82	99.72	89.02
Err-min(S)	99.99	99.77	97.85	100.0	99.37	83.94	100.0	100.0	99.93	97.21	99.99	99.27
Err-min(C)	100.0	99.66	100.0	100.0	99.96	84.78	100.0	100.0	100.0	98.29	100.0	100.0
Err-max	82.48	97.52	99.96	99.20	*37.41*	83.34	97.56	76.02	*63.89*	97.32	98.85	85.22
RobustEM	78.49	99.41	99.10	99.40	*40.25*	89.94	97.84	76.00	73.44	98.75	94.36	*68.59*

Table 3: The detection accuracy (%) for unlearnable examples under Algorithms 1 and 2, the accuracy is for the training and validation set respectively. "Linear" represents linear model and "2-NN" represents two-layer NN. For Algorithm 2, two columns are for bias-shifting noise $\epsilon_b = \pm 0.5e$. If dataset is recognized as a clean set, their accuracy are marked in italic.

norm regularization and loss is not greater than a constant μ_2*. If* $\mu_1 > \mu_2$*, then it holds* $\max_i \|\boldsymbol{\epsilon}_i\| \geq \frac{\mu_1}{2\mu_2(C-1)}$*, where* C *is the number of classes.*

Theorem 9 indicates that if the poison budget is not larger than a certain constant, the poisoned training set will not be linearly separable, which results in a failure of robust unlearnable attacks. Table 8 demonstrates that the poison budget $2/255$ is not enough to make Region-16 and Err-min(S) poisoned CIFAR-10 linearly separable. Together with Theorem 8, our discussions explain why adversarial training with budget $6/255$ is effective in defending against poisons with budget $8/255$ as shown in Table 6.

Experimental Results

In this section, experimental results are provided to verify the algorithms and the theoretical results of this paper. Experimental setups are given in Appendix B. Our codes are available at https://github.com/hala64/udp.

Experimental Results on Poison Detection

Experimental results for Algorithms 1 and 2 are given in Table 3. The detection bound B is 0.7. For more results on poison detection, please refer to Appendix G.

Detection performance. Results in Table 3 show that most of the poisons can be detected by linear models and bias-shifting noise tests with $\boldsymbol{\epsilon}_b = 0.5\boldsymbol{e}$, and all of them can be detected by two-layer NN and bias-shifting noise with $\boldsymbol{\epsilon}_b = -0.5\boldsymbol{e}$. Combined with results in Appendix G, even for robust unlearnable examples, such as RobustEM and Adv Inducing (Wang, Wang, and Wang 2021), or for poisons designed to reduce robust generalization power, such as Hypocritical (Tao et al. 2022), our detection methods still work.

Different poison ratios. Although unlearnable examples only work when all training data is poisoned, our detection methods still perform well even only a part of data is poisoned. Table 4 presents detection results for different poison ratios. It is shown that as long as 60% of data are unlearnably poisoned, it will be detected by the victims.

Ratio	100%	80%	60%	40%	20%
Clean data	*49.08*	—	—	—	—
Random(C)	100.0	86.24	77.53	*65.92*	*53.07*
Region-16	99.98	87.15	78.77	70.53	*57.90*
Err-min(S)	97.85	85.06	73.66	71.59	70.73
Err-min(C)	100.0	85.38	75.18	*62.07*	*53.44*
RobustEM	99.10	86.71	73.06	*60.92*	*51.52*

Table 4: Detection accuracy under different poison ratios by Algorithm 2 with $\epsilon_b = -0.5e$.

False poisitives and false negatives Our detection methods are designed to assess entire datasets, rather than individual data. Therefore, we analyze false positives and false negatives on different unlearnable datasets under varying detection bound B. With reference to Tables 3 and 16, we can evaluate the occurrences of FP and FN across a range of clean and poisoned datasets. The results are presented in Table 7.

Experimental Results on Poison Defense

We evaluate the defense power of Algorithm 3 with AT-based methods, UEraser (Qin et al. 2023), and ISS (Liu, Zhao, and Larson 2023). The results demonstrate that even without adversarial noise, using stronger data augmentation alone can achieve defense power comparable to adversarial training, which indicates that breaking detectability is a key to defending against unlearnable examples.

Our defense method achieves state-of-the-art performance on most of the existing unlearnable examples, such as Region-16, Err-min(S), NTGA and TUE poisons, and achieves comparable defense power for RobustEM and AR, only performs a little suboptimally for Err-max poisons, as shown in Table 5 compared to Tables 6 and 23. We also evaluate the poison method for deceiving adversarial training, called EntF (Wen et al. 2023), and the results also show the advantage of our method. Additional experimental results are in Appendix H.

Moreover, since we only conduct adversarial training on the simple two-layer NN, our method is much more time efficient than AT-based methods, as shown in Table 13, adversarial noise generated on two-layer NN is more than 3 times

Method/Poison	Region-16	Err-min(S)	Err-max	RobustEM	NTGA	AR	EntF	TUE
No defense	19.86	10.09	7.19	25.30	11.23	17.18	83.10	10.00
AT-based methods	85.78	84.00	**84.75**	81.06	84.19	85.25	73.61	83.94
Adversarial Noises	56.35	10.72	28.74	25.69	17.96	14.57	48.15	90.07
UEraser	82.66	86.37	47.46	78.39	82.15	**87.21**	87.40	75.96
ISS	67.11	85.24	84.36	**83.84**	85.64	85.11	77.02	84.32
SDA(ours)	77.20	75.62	56.96	79.21	78.48	75.81	**90.84**	73.80
AN+SDA(ours)	**93.51**	**88.01**	61.19	79.28	**89.00**	80.20	88.17	**92.76**

Table 5: Test accuracy (%) of different defense methods for poisoned CIFAR-10.

Poison	0	1/255	2/255	3/255	4/255	6/255	8/255	12/255	16/255
Region-16	19.86	24.32	29.57	50.09	72.13	**77.03**	72.65	67.65	62.78
Err-min(S)	10.09	10.01	10.13	18.64	69.92	**76.53**	72.13	67.23	61.70
RobustEM	25.30	24.92	28.50	33.74	46.01	**76.69**	72.19	63.16	53.34

Table 6: Test accuracy (%) when conducting adversarial training with budgets $i/255, i = 0, \ldots, 16$ to defend poisoned CIFAR-10 with the poison budget $\epsilon = 8/255$. We do not use any data augmentation here for better verification of our theorems.

Bound/(FP/FN)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Bias+0.5	3/0	3/0	1/1	1/1	0/1	0/1	0/2	0/5	0/8
Bias-0.5	3/0	3/1	3/1	2/1	0/1	0/1	0/1	0/1	0/3
Linear	3/0	3/0	3/0	2/1	0/4	0/5	0/6	0/10	0/13
2-NN	2/0	2/0	1/0	1/0	1/0	0/1	0/1	0/2	0/9

Table 7: False positives and false negatives of our detection methods across different datasets and unlearnable examples.

Poison/budget	0	1/255	2/255	4/255	6/255	8/255
Region-16	46.53	67.85	89.06	98.32	99.65	99.87
Err-min(S)	46.53	76.63	93.15	99.87	99.97	99.99
RobustEM	46.53	55.44	61.29	69.03	74.62	78.49

Table 8: Linear separability rate of poisoned CIFAR-10 with different poison budget η.

faster than that generated on ResNet18.

Evaluation of Criteria Between the Poison Budget and the Defense Budget

Table 6 shows the test accuracy under different adversarial defense budgets. For adversarial training with budget $\epsilon \leq 2/255$ and unlearnable poisoning attacks with budget $\eta = 8/255 \geq 4\epsilon$, the test accuracy is less than 30%, which is much lower than the linear separability rate 46.53% of CIFAR-10. This verifies Theorem 7 that adversarial training fails when the defense budget is too small.

Also, with the increase of defense budgets, the defense power of adversarial training initially increases rapidly, but then begins to gradually decrease. This is because when the defense budget increases, the gap between poison and defense budgets decreases, eventually becoming too small to achieve linear separability. As shown in Table 6, adversarial training is effective when the defense budget reaches $6/255$, since the gap $2/255$ is not large enough to make the poisoned dataset linearly separable, which verifies Theorems 8 and 9.

It is worth noting that different from (Tao et al. 2021), adversarial training here is the tool to maintain accuracy rather than robustness. Therefore, the effective defense budget here is less than in (Tao et al. 2021). From Table 8, RobustEM never becomes linearly separable, indicating that achieving linear separability is a sufficient but not necessary condition for generating unlearnable examples. With a further increase in the defense budget, the trade-off between accuracy and robustness emerges (Tsipras et al. 2018), leading to a gradual drop in test accuracy. For more discussions and experiments on this topic, please refer to Appendix I.2.

Conclusion

In this paper, we demonstrate that unlearnable examples can be easily detected. We prove that linear separability always exists for certain unlearnable poisoned dataset, and propose effective detection methods. We use stronger data augmentations with adversarial noises of simple networks to achieve effective defense for unlearnable examples. Furthermore, we derive a certified upper bound for the poison budget relative to the adversarial budget on adversarial training.

Limitations and future work. From Table 6, the results in Theorem 7 have rooms for improvements. It is desirable to craft more potent defense methods against Err-max poisons. Also, it is imperative to create more sophisticated unlearnable examples that resist existing detection and defense methods.

Supplementary Materials The appendix and full version of this paper are provided in (Zhu, Yu, and Gao 2023).

Acknowledgements

This work is supported in part by NKRDP grant No. 2018YFA0306702 and NSFC grant No. 11688101.

References

Abusnaina, A.; Wu, Y.; Arora, S.; Wang, Y.; Wang, F.; Yang, H.; and Mohaisen, D. 2021. Adversarial example detection using latent neighborhood graph. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7687–7696.

Barni, M.; Kallas, K.; and Tondi, B. 2019. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, 101–105. IEEE.

Biggio, B.; Nelson, B.; and Laskov, P. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*.

Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I.; and Srivastava, B. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*.

Chen, H.; Fu, C.; Zhao, J.; and Koushanfar, F. 2019. DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks. In *IJCAI*, 4658–4664.

Chen, J.; Zhang, X.; Zhang, R.; Wang, C.; and Liu, L. 2021. De-pois: An attack-agnostic defense against data poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 16: 3412–3425.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Cherepanova, V.; Goldblum, M.; Foley, H.; Duan, S.; Dickerson, J. P.; Taylor, G.; and Goldstein, T. 2020. LowKey: Leveraging Adversarial Attacks to Protect Social Media Users from Facial Recognition. In *International Conference on Learning Representations*.

Dong, Y.; Yang, X.; Deng, Z.; Pang, T.; Xiao, Z.; Su, H.; and Zhu, J. 2021. Black-box detection of backdoor attacks with limited information and data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16482–16491.

Feng, J.; Cai, Q.-Z.; and Zhou, Z.-H. 2019. Learning to confuse: generating training time adversarial data with autoencoder. *Advances in Neural Information Processing Systems*, 32.

Fowl, L.; Goldblum, M.; Chiang, P.-y.; Geiping, J.; Czaja, W.; and Goldstein, T. 2021. Adversarial examples make strong poisons. *Advances in Neural Information Processing Systems*, 34: 30339–30351.

Fu, S.; He, F.; Liu, Y.; Shen, L.; and Tao, D. 2021. Robust unlearnable examples: Protecting data privacy against adversarial learning. In *International Conference on Learning Representations*.

Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; and McDaniel, P. 2017. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*.

Gu, T.; Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7: 47230–47244.

Guo, J.; Li, A.; and Liu, C. 2021. AEVA: Black-box Backdoor Detection Using Adversarial Extreme Value Analysis. In *International Conference on Learning Representations*.

He, H.; Zha, K.; and Katabi, D. 2022. Indiscriminate poisoning attacks on unsupervised contrastive learning. *arXiv preprint arXiv:2202.11202*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Hu, S.; Liu, X.; Zhang, Y.; Li, M.; Zhang, L. Y.; Jin, H.; and Wu, L. 2022. Protecting facial privacy: Generating adversarial identity masks via style-robust makeup transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15014–15023.

Huang, H.; Ma, X.; Erfani, S. M.; Bailey, J.; and Wang, Y. 2020a. Unlearnable Examples: Making Personal Data Unexploitable. In *International Conference on Learning Representations*.

Huang, W. R.; Geiping, J.; Fowl, L.; Taylor, G.; and Goldstein, T. 2020b. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33: 12080–12091.

Kalimeris, D.; Kaplun, G.; Nakkiran, P.; Edelman, B.; Yang, T.; Barak, B.; and Zhang, H. 2019. Sgd on neural networks learns functions of increasing complexity. *Advances in neural information processing systems*, 32.

Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, 1885–1894. PMLR.

Levine, A.; and Feizi, S. 2020. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768*.

Liu, T. Y.; Yang, Y.; and Mirzasoleiman, B. 2022. Friendly noise against adversarial noise: a powerful defense against data poisoning attack. *Advances in Neural Information Processing Systems*, 35: 11947–11959.

Liu, Z.; Zhao, Z.; and Larson, M. 2023. Image shortcut squeezing: Countering perturbative availability poisons with compression. *arXiv preprint arXiv:2301.13838*.

Lu, Y.; Kamath, G.; and Yu, Y. 2022. Indiscriminate Data Poisoning Attacks on Neural Networks. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*.

Ma, Y.; Zhu, X.; and Hsu, J. 2019. Data poisoning against differentially-private learners: attacks and defenses. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 4732–4738.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Metzen, J. H.; Genewein, T.; Fischer, V.; and Bischoff, B. 2016. On Detecting Adversarial Perturbations. In *International Conference on Learning Representations*.

Muñoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E. C.; and Roli, F. 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 27–38.

Qin, T.; Gao, X.; Zhao, J.; Ye, K.; and Xu, C.-Z. 2023. Learning the unlearnable: Adversarial augmentations suppress unlearnable example attacks. *arXiv preprint arXiv:2303.15127*.

Raghavan, V.; Mazzuchi, T.; and Sarkani, S. 2022. An improved real time detection of data poisoning attacks in Deep Learning Vision systems. *Discover Artificial Intelligence*, 2(1): 18.

Ren, J.; Xu, H.; Wan, Y.; Ma, X.; Sun, L.; and Tang, J. 2022. Transferable Unlearnable Examples. In *The Eleventh International Conference on Learning Representations*.

Sandoval-Segura, P.; Singla, V.; Fowl, L.; Geiping, J.; Goldblum, M.; Jacobs, D.; and Goldstein, T. 2022a. Poisons that are learned faster are more effective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 198–205.

Sandoval-Segura, P.; Singla, V.; Geiping, J.; Goldblum, M.; Goldstein, T.; and Jacobs, D. 2022b. Autoregressive perturbations for data poisoning. *Advances in Neural Information Processing Systems*, 35: 27374–27386.

Schwarzschild, A.; Goldblum, M.; Gupta, A.; Dickerson, J. P.; and Goldstein, T. 2021. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, 9389–9398. PMLR.

Shafahi, A.; Huang, W. R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; and Goldstein, T. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31.

Shan, S.; Wenger, E.; Zhang, J.; Li, H.; Zheng, H.; and Zhao, B. Y. 2020. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th USENIX security symposium (USENIX Security 20)*, 1589–1604.

Steinhardt, J.; Koh, P. W. W.; and Liang, P. S. 2017. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30.

Subedar, M.; Ahuja, N.; Krishnan, R.; Ndiour, I. J.; and Tickoo, O. 2019. Deep probabilistic models to detect data poisoning attacks. *arXiv preprint arXiv:1912.01206*.

Tao, L.; Feng, L.; Wei, H.; Yi, J.; Huang, S.-J.; and Chen, S. 2022. Can Adversarial Training Be Manipulated By Non-Robust Features? *Advances in Neural Information Processing Systems*, 35: 26504–26518.

Tao, L.; Feng, L.; Yi, J.; Huang, S.-J.; and Chen, S. 2021. Better safe than sorry: Preventing delusive adversaries with adversarial training. *Advances in Neural Information Processing Systems*, 34: 16209–16225.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2018. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*.

Turner, A.; Tsipras, D.; and Madry, A. 2019. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*.

Wang, W.; Levine, A. J.; and Feizi, S. 2022. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In *International Conference on Machine Learning*, 22769–22783. PMLR.

Wang, Y.; Mianjy, P.; and Arora, R. 2021. Robust learning for data poisoning attacks. In *International Conference on Machine Learning*, 10859–10869. PMLR.

Wang, Z.; Wang, Y.; and Wang, Y. 2021. Fooling Adversarial Training with Inducing Noise. *arXiv preprint arXiv:2111.10130*.

Weber, M.; Xu, X.; Karlaš, B.; Zhang, C.; and Li, B. 2020. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904*.

Wen, R.; Zhao, Z.; Liu, Z.; Backes, M.; Wang, T.; and Zhang, Y. 2023. Is Adversarial Training Really a Silver Bullet for Mitigating Data Poisoning? In *The Eleventh International Conference on Learning Representations*.

Yu, D.; Zhang, H.; Chen, W.; Yin, J.; and Liu, T.-Y. 2021. Indiscriminate poisoning attacks are shortcuts. *arXiv preprint arXiv:2111.00898*.

Yuan, C.-H.; and Wu, S.-H. 2021. Neural tangent generalization attacks. In *International Conference on Machine Learning*, 12230–12240. PMLR.

Zhu, C.; Huang, W. R.; Li, H.; Taylor, G.; Studer, C.; and Goldstein, T. 2019. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, 7614–7623. PMLR.

Zhu, Y.; Yu, L.; and Gao, X.-S. 2023. Detection and Defense of Unlearnable Examples. *arXiv preprint arXiv:2312.08898*.