INDISCRIMINATE POISONING ATTACKS ARE SHORTCUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Indiscriminate data poisoning attacks, which add imperceptible perturbations to training data to maximize the test error of trained models, have become a trendy topic because they are thought to be capable of preventing unauthorized use of data. In this work, we investigate why these perturbations work in principle. We find that the perturbations of advanced poisoning attacks are almost **linear separable** when assigned with the target labels of the corresponding samples. This is an important population property for various perturbations that were not unveiled before. Moreover, we further confirm that linear separability is indeed the workhorse for poisoning attacks. We synthesize linear separable data as perturbations and show that such synthetic perturbations are as powerful as the deliberately crafted attacks. Our finding also suggests that the *shortcut learning* problem is more serious than previously believed as deep learning heavily relies on shortcuts even if they are of an imperceptible scale and mixed together with the normal features. It also suggests that pre-trained feature extractors can be a powerful defense.

1 INTRODUCTION

Big datasets crawled from the Internet keep advancing the state-of-the-art results (Devlin et al., 2018; He et al., 2020; Chen et al., 2020). However, there are increasing concerns about the unauthorized use of personal data (Hill & Krolik, 2019; Prabhu & Birhane, 2020; Carlini et al., 2020). For instance, a private company has collected more than three billion face images to build commercial face recognition models without acquiring any user consents (Hill, 2020). To address those concerns, many data poisoning attacks have been proposed to prevent data from being learned by unauthorized deep models (Feng et al., 2019; Shen et al., 2019; Shan et al., 2020; Huang et al., 2021; Cherepanova et al., 2021; Yuan & Wu, 2021; Fowl et al., 2021a;b). They add imperceptible perturbations to the training data so that the model accuracy on unseen data is arbitrarily bad. These attacks are also known as *indiscriminate poisoning attacks* as they do not have specific target examples (Barreno et al., 2010). We give an illustration of indiscriminate poisoning attacks in Figure 1.

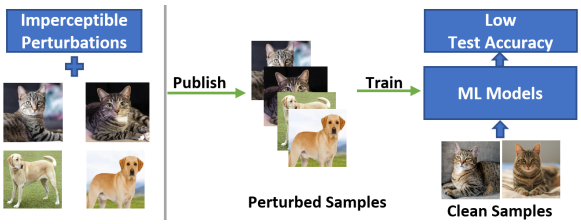

Figure 1: An illustration of indiscriminate poisoning attacks.

Roughly speaking, there are three methods available to construct the indiscriminate poisoning attack. The first method formulates the perturbations as the solution of a bi-level optimization problem (Biggio et al., 2012; Feng et al., 2019; Fowl et al., 2021a; Yuan & Wu, 2021). The bi-level optimization problem requires models trained on perturbed data to have the maximum loss on unseen data. Then Huang et al. (2021) conceive a simpler poisoning attack called *error-minimizing noises*, where the perturbation on training data is crafted by minimizing the training loss so that nothing is left to backpropagate in the regular training procedure. Recently, Fowl et al. (2021b) point out that the *error-maximizing noises*, which are commonly used as adversarial examples, can serve as an

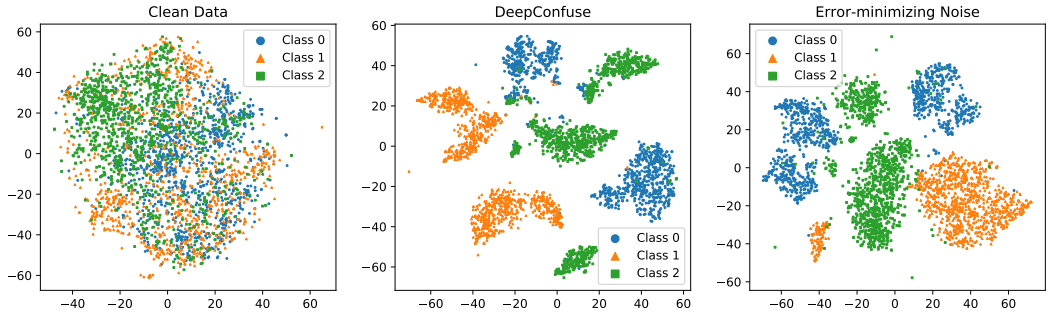

Figure 2: T-SNEs of the first three classes of clean CIFAR-10 data and the perturbations generated via DeepConfuse (Feng et al., 2019) and error-minimizing noises (Huang et al., 2021).

indiscriminate poisoning attack as well. Despite of these quite different approaches, they all give strong poisoning attacks. Intrigued by this observation, we ask the following question:

What is the underlying working principle of indiscriminate poisoning attacks?

To answer the above question, we first take a closer look at the perturbations of existing attacks. We visualize the perturbations of two advanced attacks via two-dimensional T-SNEs (Van der Maaten & Hinton, 2008) in Figure 2. The experimental setup is in Section 2.2. Surprisingly, the perturbations with the same class label are well clustered, suggesting that the perturbations would be **linear separable** in the original high-dimensional space. We confirm this by fitting the perturbations with linear models. The perturbations are assigned with the labels of their target examples. It turns out that simple softmax regression models can fit the perturbations of four representative attacks with $> 90\%$ accuracy. This finding suggests that using linear separable perturbations may be the key for an indiscriminate poisoning attack to succeed.

To further confirm that the linear separability is a sufficient (not only necessary) condition, we reverse the above procedure: synthesizing some linear separable data to see if they can serve as data poisoning attacks. Extensive experiments on benchmark datasets and models demonstrate that the synthetic perturbations can be as powerful as advanced indiscriminate poisoning attacks. Notably, generating synthetic data as perturbations is significantly easier and cheaper than existing attacks as it does not require solving any optimization problems. This finding reveals that linear separability is indeed the workhorse to the success of indiscriminate poisoning attacks.

The above finding coincidentally matches a recent concept called *shortcut learning* (Geirhos et al., 2020). Shortcut learning stands for the behavior that deep models tend to rely on features that have a correlation with training labels but do not generalize on test data. Such features are referred to as shortcuts. With this concept, the perturbations of indiscriminate poisoning attacks are also shortcuts. We note that the shortcuts in previous works are usually part of natural data, which are somehow heuristic, e.g., "grass" is a shortcut for recognizing "cow" in natural images (Beery et al., 2018; Geirhos et al., 2020). In this work, we expose a more explicit form of shortcuts and discuss extensively how to construct such shortcuts. We unveil that deep learning models would overwhelmingly rely on spurious shortcuts even though the shortcuts are scaled down to an imperceptible magnitude. This finding exposes a fundamental vulnerability of deep models and hence may be of independent interest to the community.

Finally, our understanding of the working principle of indiscriminate poisoning attacks also motivates a powerful defense against them. As the attacks succeed by providing shortcuts, they will be less effective if those shortcuts are filtered out. To achieve this, we use pre-trained models as feature extractors. If a pre-trained model is trained on datasets that have similar distributions as the target dataset, it can extract useful features and avoid shortcuts. Our experiments suggest that training only the linear classification layers with the features extracted from the pre-trained model can successfully defend existing attacks.

Our findings and contributions are summarized as follows:

- We find that the perturbations used in several advanced indiscriminate poisoning attacks are (almost) linear separable. We further validate that this property is the workhorse for the perturbations to be effective in a poisoning attack. To the best of our knowledge, we are the first to unveil this important property, which is fundamental to understand various poisoning attacks and the vulnerability of machine learning.
- We connect this property to the shortcut learning problem and demonstrate the omnipresence of shortcuts in learning. We show that one can construct invisible shortcuts that machine learning models would heavily rely on. We believe this finding greatly widens the understanding of shortcuts in machine learning.
- Motivated by breaking the shortcuts, we use pre-trained feature extractors to defend against indiscriminate poisoning attacks. Experiments show that the proposed defense is very powerful.

1.1 Related Work

Data poisoning. In general, data poisoning attacks perturb training data to intentionally cause some malfunctions of the target model (Biggio & Roli, 2018; Goldblum et al., 2020; Schwarzschild et al., 2021). A common class of poisoning attacks aims to cause test-time error on some given samples (Koh & Liang, 2017; Muñoz-González et al., 2017; Chen et al., 2017; Koh et al., 2018; Shafahi et al., 2018; Zhu et al., 2019; Geiping et al., 2020; Huang et al., 2020) or on all unseen samples (Biggio et al., 2012; Feng et al., 2019; Liu & Shroff, 2019; Shen et al., 2019; Huang et al., 2021; Yuan & Wu, 2021; Fowl et al., 2021a;b). The latter attacks are also known as indiscriminate poisoning attacks as they do not have specific target examples (Barreno et al., 2010). Backdoor attack is another type of poisoning attack that perturbs training data so that the attacker can manipulate the target model's output with a designed trigger (Chen et al., 2017; Shafahi et al., 2018; Turner et al., 2018; Xie et al., 2019; Bagdasaryan et al., 2020; Nguyen & Tran, 2020; Saha et al., 2020; Nguyen & Tran, 2021). In this work, we investigate indiscriminate poisoning attacks and reveal the workhorse of them. We show that the perturbations of advanced attacks are (almost) linear separable. We further confirm that using linear separable perturbations is a sufficient condition to perform strong attacks.

There are effective defenses against indiscriminate poisoning attacks. Huang et al. (2021) show training the target model with advanced data augmentation methods improves its test performance. For example, when the training data are perturbed with error-minimizing noises, *Mixup* (Zhang et al., 2017) improves the test accuracy from 19.9% to 58.5%. Another defense is to train the target model with *adversarial training* (Madry et al., 2018), which currently achieves the best performance (Huang et al., 2021; Fowl et al., 2021b; Tao et al., 2021). Recently, Radiya-Dixit & Tramèr (2021) challenge the security of poisoning attacks from another view. They argue that unless existing attacks can fool all future defenses, they cannot protect the data well because the adversary can simply save the perturbed data and wait for better defenses. In this work, we propose to use pre-trained models to extract useful features from perturbed data. Experiments show the proposed defense is very powerful as the pre-trained feature extractors can avoid shortcuts.

Shortcut learning. Recently, the community has realized that deep models may rely on shortcuts to make decisions (Beery et al., 2018; Niven & Kao, 2019; Ilyas et al., 2019; Geirhos et al., 2020; Huh et al., 2021). Shortcuts are spurious features that are correlated with training labels but do not generalize on test data. Beery et al. (2018) show that a deep model would fail to recognize cows when the grass background is removed, suggesting that the model takes "grass" as a shortcut for "cow". Niven & Kao (2019) show that large language models use the strong correlation between some simple words and labels to make decisions, instead of trying to understand the sentence. For instance, the word "not" is directly used to predict negative labels. In this work, we show the shortcut learning problem is more serious than previously believed. Our experiments in Section 3 demonstrate that deep models only pick shortcuts even if the shortcuts are scaled down to an imperceptible magnitude and mixed together with normal features. These experiments reveal another form of shortcut learning, which has already been unconsciously exploited by indiscriminate data poisoning attacks.

1.2 Notations

We use bold lowercase letters, e.g., $\boldsymbol{v}$, and bold capital letters, e.g., $\boldsymbol{M}$, to denote vectors and matrices, respectively. The L_p norm of a vector $\boldsymbol{v}$ is denoted by $\|\boldsymbol{v}\|_p$. A sample consists of feature $\boldsymbol{x}$ and label

y. We use $\mathbb{D}$ to denote a dataset which is sampled from distribution $\mathcal{D}$. The loss of a model f on a given sample is denoted by $\ell(f(\boldsymbol{x}), y)$.

2 Indiscriminate Poisoning Attacks Use Linear Separable Perturbations

In this section, we investigate the perturbations of several advanced indiscriminate poisoning attacks. First, we introduce three approaches of poisoning attacks. Then, we visualize them with two-dimensional T-SNEs. Finally, we verify that the perturbations of four advanced attacks are almost linear separable by fitting them with linear models and two-layer neural networks.

2.1 Three Types Of Indiscriminate Poisoning Attacks

2.1.1 The Alternative Optimization Approach

We first introduce the alternative optimization approach to generate perturbations for indiscriminate poisoning attacks. It solves the following bi-level objective,

$$\begin{aligned} &\arg\max_{\{\boldsymbol{\delta}\}\in\Delta} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(f^*(\boldsymbol{x}), y)], \\ &\text{s.t. } f^* \in \arg\min_{f} \sum_{(\boldsymbol{x},y)\in\mathbb{D}} \ell(f(\boldsymbol{x}+\boldsymbol{\delta}), y), \end{aligned} \tag{1}$$

where $\boldsymbol{\delta}$ is a sample-wise perturbation and Δ is a constraint set of all perturbations. To put in other words, the optimal solution on perturbed data (specified by the second objective) should have a maximum loss on clean data (specified by the first objective). The perturbations are restricted to not affect the normal data utility.

Directly solving Equation (1) is intractable for deep neural networks and recent works have designed multiple approximate solutions (Feng et al., 2019; Fowl et al., 2021a; Yuan & Wu, 2021). Feng et al. (2019) use multiple rounds of optimization to generate perturbations. At each round, they first approximately optimize the second objective by updating a target model on perturbed data for a few steps. Then they approximate the first objective by updating a generator for a few steps. The outputs of the generator are used as perturbations. Another example is the Neural Tangent Generalization Attacks (NTGAs) in Yuan & Wu (2021). They rewrite the entire Equation (1) into a single objective based on the recent development of Neural Tangent Kernels (Jacot et al., 2018). Then they solve the new objective with a lightweight surrogate model.

2.1.2 The Error-minimizing Noise

Huang et al. (2021) propose a simple way to generate data poisoning perturbations. Instead of solving Equation (1), they use the following objective,

$$\min_{f} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathbb{D}}[\arg\min_{\{\boldsymbol{\delta}\}\in\Delta} \ell(f(\boldsymbol{x}+\boldsymbol{\delta}), y)]. \tag{2}$$

That is to say, the perturbations are intentionally optimized to reduce the training loss. A randomly initialized model is used as a surrogate of the target model. They use multiple rounds of bi-level optimization to generate perturbations. At each round, they first train the surrogate model for a few steps to minimize the loss on perturbed data. Then they optimize the perturbations to also minimize the loss of the surrogate model. They repeat the above process until the loss on perturbed data is smaller than a pre-defined threshold. The main motivation is that if the training loss is zero, then the target model will have nothing to learn as there is nothing to backpropagate.

2.1.3 THE ERROR-MAXIMIZING NOISE

Instead of using bi-level objectives, Fowl et al. (2021b) show that the objective of adversarial examples, which is a single objective of maximizing the loss, is sufficient to generate powerful data poisoning perturbations. They use the classic objective of adversarial examples,

$$\underset{\{\boldsymbol{\delta}\}\in\Delta}{\arg\max}\, \mathbb{E}_{(\boldsymbol{x},y)\sim\mathbb{D}}\left[\ell\left(f\left(\boldsymbol{x}+\boldsymbol{\delta}\right),y\right)\right], \tag{3}$$

where f is a trained model and is referred to as the crafting model. Surprisingly, Fowl et al. (2021b) demonstrate that this simple objective can generate perturbations that achieve state-of-the-art attack performance.

2.2 VISUALIZING THE PERTURBATIONS

Although the three approaches in Section 2.1 have different objectives, they all manage to generate perturbations with the same effect. Intrigued by this observation, we seek to find the underlying working principle of those approaches if there is a common pattern of different types of perturbations.

To find such a common pattern, we first visualize different types of perturbations by computing their two-dimensional t-SNEs (Van der Maaten & Hinton, 2008). We generate perturbations for the CIFAR-10 dataset using the official implementations of DeepConfuse[1] and error-minimizing noise[2]. The configuration is set to be the one that achieves the best attack performance on CIFAR-10.

The results are shown in Figure 2. Surprisingly, for both types of perturbations, the perturbations for the same class are well clustered, suggesting that even linear models can classify them well. For comparison, we also compute the t-SNEs of the clean data. As shown in Figure 2, in contrast with the t-SNEs of perturbations, the projections of different classes of the clean data are mixed together, which indicates that they require a complex neural network to be correctly classified.

2.3 PERTURBATIONS OF POISONING ATTACKS ARE ALMOST LINEAR SEPARABLE

In order to quantify the 'linear separability' of the perturbations of different indiscriminate poisoning attacks, we fit the perturbations with simple models and examine the training accuracy. The perturbations are labeled with the labels of the corresponding target examples. We use linear models and two-layer neural networks as simple models.

In addition to the two attacks in Section 2.2, we also test two recent attacks NTGA (Yuan & Wu, 2021) and error-maximizing noise (Fowl et al., 2021b). For NTGA, we use the official implementation to generate perturbations[3]. For error-maximizing noise, we implement their algorithm with the setup in Appendix A.1 of Fowl et al. (2021b). These four chosen attacks achieve advanced attack performance and cover all three approaches introduced in Section 2.1.

We train the simple models with standard cross-entropy loss. Before training, all perturbations are flattened into 1-dimensional vectors and normalized to unit norm. The two-layer neural networks have a width of 30. All models are trained with the L-BFGS optimizer (Liu & Nocedal, 1989) for 50 steps.

The results are presented in Table 1. Compared to the results on clean data, simple models can easily fit the perturbations. On all attacks considered, linear models achieve more than 90% training accuracy and two-layer neural networks achieve nearly 100% training accuracy. These results confirm that the perturbations of advanced indiscriminate poisoning attacks are all (almost) linear separable.

2.4 CONNECTING OUR FINDINGS TO SHORTCUT LEARNING

The fact that the perturbations can be easily fitted by linear models naturally connects to a recent concept named shortcut learning (Geirhos et al., 2020). Shortcut learning summarizes a general phenomenon when any learning system makes decisions based on spurious features that are useful

[1] https://github.com/kingfengji/DeepConfuse
[2] https://github.com/HanxunH/Unlearnable-Examples
[3] https://github.com/lionelmessi6410/ntga

Table 1: Training accuracy (in %) of simple models on the perturbations of different attacks.

Algorithm	Linear Model	Two-layer NN
Clean Data	49.9	70.1
DeepConfuse (Feng et al., 2019)	100.0	100.0
NTGA (Yuan & Wu, 2021)	100.0	100.0
Error-minimizing (Huang et al., 2021)	100.0	100.0
Error-maximizing (Fowl et al., 2021b)	91.5	99.9

for training but do not generalize on test data, referred to as *shortcut features*. Shortcut features have been found in many different fields. For vision tasks, Beery et al. (2018) show deep models fail to recognize cows when the grass background is removed from natural images, suggesting the grass background is a shortcut for predicting cows. In the field of natural language processing, Niven & Kao (2019) show language models use the strong correlation between some simple words and labels to make decisions, instead of really understanding the data.

With the presence of shortcut learning, it seems reasonable to postulate that indiscriminate poisoning attacks succeed by creating shortcuts to the target model. A major difference between the perturbations of poisoning attacks and existing shortcut features is that the perturbations are of an imperceptible scale and mixed together with useful features. Since there is no direct evidence to show deep models will take this kind of shortcuts, the observation that the perturbations are linear separable may only be some superficial results of the underlying root cause. In the next section, we design experiments to confirm the postulated explanation. We synthesize imperceptible and linear separable data and show deep models are very vulnerable to such synthetic shortcuts.

3 Is Linear Separability A Sufficient Condition For Poisoning Attacks to Succeed?

Although we have demonstrated that the perturbations of four advanced attacks are all almost linear separable, it is a bit early to claim that 'linear separability' is the underlying working principle of indiscriminate poisoning attacks. For example, perturbations are linear separable may only be a necessary but not sufficient condition for poisoning attacks to succeed. In order to verify this postulated explanation, we use simple synthetic data to serve as perturbations and compare their effectiveness with existing poisoning attacks. It turns out that the synthetic perturbations are as powerful as the perturbations of advanced attacks. We note that there exist synthesize datasets that offer a stratification of features (Hermann & Lampinen, 2020; Shah et al., 2020). Those synthetic data can not be used as perturbations as they are visible and concatenations of features, e.g., vertically concatenations of images from MNIST and CIFAR-10.

The rest of this section is organized as follows. In Section 3.1, we first give an algorithm for generating synthetic data as perturbations. In Section 3.2, we verify the effectiveness of synthetic perturbations on different models and datasets.

3.1 Generating Simple Synthetic Data

The synthetic data in this section are generated via two building blocks. In the first block, we use an algorithm in Guyon (2003) to generate samples from some normal distributions. In the second block, we transfer the samples into two-dimensional images in order to apply them to benchmark datasets.

The first building block proceeds as follows. We first generate some points that are normally distributed around the vertices of a hypercube. The points around the same vertex are assigned with the same label. Then for each class, we introduce different covariance. Any two classes of the generated points can be easily classified by a hyperplane as long as the side length of the hypercube is reasonably large. We give the pseudocode of this block in Appendix A.

In the second building block, we pad each dimensional of the sampled points and reshape them into two-dimensional images. The padding operation introduces local correlation into the synthetic images. Local correlation is an inherent property of natural images. In Appendix C, we show the padding

Algorithm 1: Transform A 1-D Sample Into A 2-D Image

1: **Input:** input vector $\boldsymbol{x} \in \mathbb{R}^d$, target image size s.

//For simplicity, we assume the image is square and s is divisible by $\sqrt{d}$.

2: Compute patch size $p = s/\sqrt{d}$.

3: Get $\boldsymbol{x}' \in \mathbb{R}^{s^2}$ by duplicating each dimension of $\boldsymbol{x}$ for p^2 times.

4: Rearrange $\boldsymbol{x}'$ to get d patches so that the pixel values in each patch are the same.

operation is necessary to make the synthetic perturbations remain effective when data augmentation methods are applied. The pseudocode of this block is given in Algorithm 1.

The synthetic images are scaled down before being used as perturbations. We visualize the synthetic perturbations and corresponding perturbed images in Figure 3. We also visualize the perturbations in Huang et al. (2021) for a comparison. The details of perturbations can be found in Section 3.2. As shown in Figure 3, the synthetic perturbations do not affect data utility.

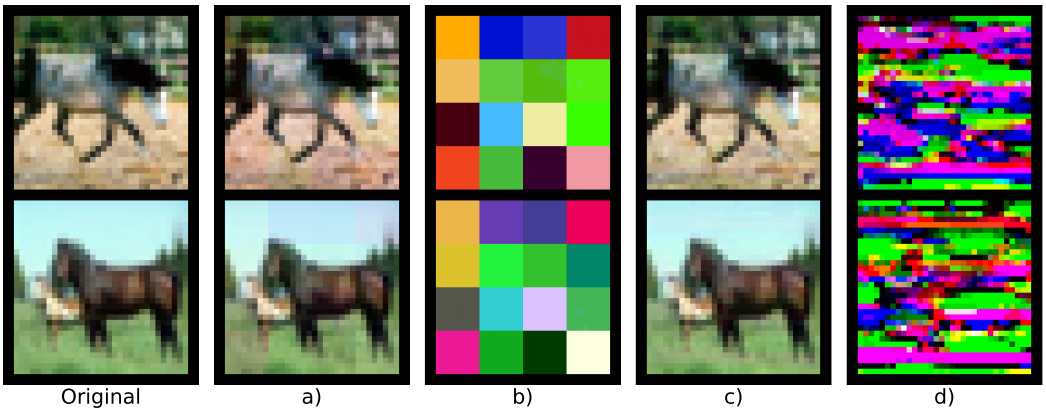

Figure 3: Visualization of perturbed images and normalized perturbations. Columns a) and b) use synthetic perturbations. Columns c) and d) use the attack in Huang et al. (2021).

3.2 Synthetic Perturbations Are Highly Effective as Data Poisoning Attack

Now we verify the effectiveness of synthetic perturbations and make comparisons with existing poisoning attacks. We perturb the entire training set following the setup in previous works (Feng et al., 2019; Huang et al., 2021; Yuan & Wu, 2021; Fowl et al., 2021b). That is, we synthesize a perturbation for every training example. We use L_2-norm for synthetic perturbations to keep the sample-wise variation in the same class. We normalize the synthetic noises into a L_2-norm ball with radius $\sqrt{d}\epsilon'$, where d is the dimension of the input.

We evaluate synthetic perturbations on three benchmark datasets: SVHN (Netzer et al., 2011), CIFAR-10, and CIFAR-100 (Krizhevsky & Hinton, 2009). The target model architectures include VGG (Simonyan & Zisserman, 2014), ResNet (He et al., 2016), and DenseNet (Huang et al., 2017). We adopt standard random cropping and flipping as data augmentation. The hyperparameters for training are standard and can be found in Appendix B. We use $\epsilon' = 6/255$ for synthetic perturbations. The patch size in Algorithm 1 is set as 8.

We first compare synthetic perturbations with existing poisoning attacks. The comparisons are made on the CIFAR-10 dataset with ResNet-18 as the target model. The perturbations in previous works are normalized with L_∞ bound with $\epsilon = 8/255$. We present the comparisons in Table 2. Then we evaluate synthetic perturbations on different models and datasets. The final test accuracy of target models is in Table 3. We also plot the training curves of target models on both clean and perturbed data in Figure 4.

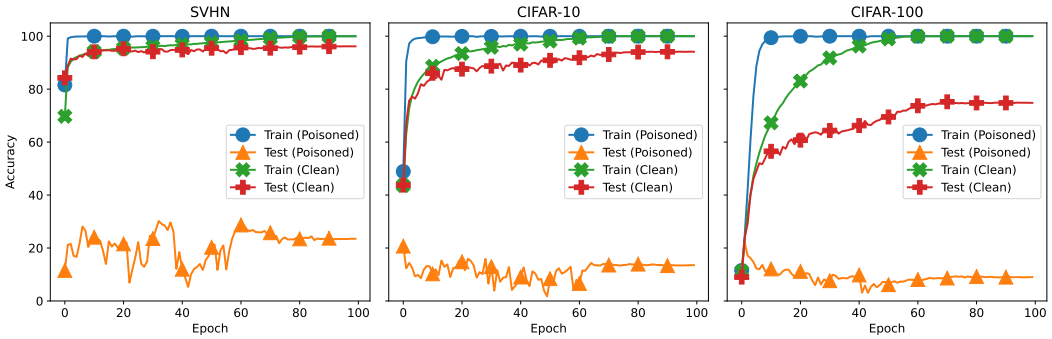

Figure 4: Training curves of ResNet-18 models on perturbed and clean data. The test accuracy is low throughout training when synthetic perturbations are added.

Table 2: Accuracy on clean test data of CIFAR-10. The target model is ResNet-18. The training data are poisoned with different attacks. The smaller the accuracy, the better the attack efficiency.

Algorithm	Test Accuracy (in %)
No Perturbation	94.69
TensorClog (Shen et al., 2019)	48.07
Alignment (Fowl et al., 2021a)	56.65
DeepConfuse (Feng et al., 2019)	28.77
NTGA (Yuan & Wu, 2021)	33.29
Error-minimizing (Huang et al., 2021)	19.93
Error-maximizing (Fowl et al., 2021b)	6.25
Synthetic Perturbations	13.54

As shown in Table 2, synthetic perturbations are as powerful as advanced poisoning attacks and reduce the test accuracy close to that of random guessing. The results in Table 3 and Figure 4 further confirm the effectiveness of synthetic perturbations. Notably, generating synthetic perturbations is data irrelevant and only takes several seconds using a single CPU core.

In summary, our experiments demonstrate that using linear separable perturbations is indeed a sufficient condition for indiscriminate poisoning attacks to succeed. Moreover, these results also expose that deep models are very vulnerable to obscured shortcuts. This finding has two meanings to the community. First, it confirms that advanced indiscriminate poisoning attacks do succeed by providing shortcuts. Second, it further exposes the shortcut learning problem, which is a fundamental vulnerability of deep models.

Table 3: Accuracy (in %) on clean test data. The target models are trained on clean data ($\mathbb{D}_c$) and data perturbed by synthetic perturbations ($\mathbb{D}_{syn}$).

Target Model	**SVHN**		**CIFAR-10**		**CIFAR-100**	
	$\mathbb{D}_c$	$\mathbb{D}_{syn}$	$\mathbb{D}_c$	$\mathbb{D}_{syn}$	$\mathbb{D}_c$	$\mathbb{D}_{syn}$
VGG-11	95.4	18.1	91.3	28.3	67.5	10.9
ResNet-18	96.2	8.0	94.7	13.5	74.8	9.0
ResNet-50	96.4	7.8	94.8	14.9	75.2	8.4
DenseNet-121	96.7	9.7	95.0	10.6	76.5	7.6

4 DEFEND INDISCRIMINATE POISONING ATTACKS WITH PRE-TRAINED FEATURE EXTRACTORS

We now explore possible defenses against indiscriminate poisoning attacks with our understanding. The experiments in Section 2 and 3, suggest that deep models only use shortcuts to make predictions and ignore the useful features. A reasonable defense should be able to filter out such shortcuts and make the decision based on normal features. To achieve this goal, we explore using pre-trained models to extract useful features. Pre-training is a popular technique that trains the model on some auxiliary data before learning on target data. If the auxiliary data is similar to the target data, the pre-trained models can recognize some useful features and hence avoid relying on shortcuts.

Our implementation is as follows. We use a ResNet-152 model pre-trained on unlabeled ImageNet with SimCLR (Chen et al., 2020). To apply the pre-trained model on CIFAR-10, we remove the final layer and add a ten-class readout layer. The output dimension of the penultimate layer is 4096. We try two different approaches to take advantage of the pre-trained model. The first approach uses the pre-trained weights as initialization and fine-tunes the whole model. The second approach only uses the outputs of the penultimate layer and trains a linear classifier on them.

We run experiments on five types of perturbed CIFAR-10 datasets that are crafted via DeepConfuse (Feng et al., 2019), NTGA (Yuan & Wu, 2021), error-minimizing noise (Huang et al., 2021), error-maximizing noise (Fowl et al., 2021b), and synthetic perturbations. The hyperparameter choices of fine-tuning can be found in Appendix B.

We use adversarial training as a baseline method. Adversarial training has been shown to be a powerful defense against indiscriminate poisoning attacks (Huang et al., 2021; Tao et al., 2021; Fowl et al., 2021b). We use the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) to generate adversarial examples. The metric is L_∞ norm with $\epsilon = 8/255$.

Table 4: Test accuracy (in %) when different countermeasures are applied. 'Linear eval.' means we fix the pre-trained weights and only train the classification layer.

Algorithm	Adv. Training	Pre-train	Pre-train (Linear eval.)
No Perturbation	86.2	95.9	95.4
DeepConfuse (Feng et al., 2019)	81.6	17.6	**94.1**
NTGA (Yuan & Wu, 2021)	85.0	39.5	**95.2**
Error-minimizing (Huang et al., 2021)	84.3	21.6	**94.0**
Error-maximizing (Fowl et al., 2021b)	81.3	37.3	**87.6**
Synthetic Perturbations	83.7	15.7	**93.8**

The results are shown in Table 4. When fine-tuning the full models, the models still pick the shortcuts and achieve poor test accuracy. In contrast, when we freeze the pre-trained weights and only train linear classifiers on the extracted features, pre-training substantially increases the test accuracy. Moreover, the pre-trained feature extractor achieves better defense than adversarial training.

Although the proposed defense achieves state-of-the-art results against advanced attacks, it could have potential risks when facing white-box attacks. The four advanced poisoning attacks evaluated in this section are all black-box, i.e., they do not have access to the target model or try to recover the target model first. The power of our defense may be compromised if the perturbations are generated to invalid the chosen pre-trained model.

5 CONCLUSION

This work gives an explanation of the working principle of indiscriminate poisoning attacks. We show advanced attacks coincidentally generate linear separable perturbations. We further synthesize linear separable perturbations to demonstrate that using linear separable perturbations is a sufficient condition for attacks to succeed. Our findings also suggest deep models are more prone to shortcuts than previously believed as they will find and heavily rely on shortcuts even when the shortcuts are scaled down to an imperceptible magnitude. Finally, our explanation also motivates us to use pre-trained feature extractors as a powerful defense.

REPRODUCIBILITY STATEMENT

We upload our source code in the supplementary materials. The README files in the uploaded code provide example commands to verify our results. Our implementation is based on Pytorch (`https://pytorch.org/`), which is a popular open-source machine learning framework.

REFERENCES

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020.

Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. *Machine Learning*, 2010.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 2018.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*, 2020.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John Dickerson, Gavin Taylor, and Tom Goldstein. Lowkey: leveraging adversarial attacks to protect social media users from facial recognition. *arXiv preprint arXiv:2101.07922*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: generating training time adversarial data with auto-encoder. *arXiv preprint arXiv:1905.09027*, 2019.

Liam Fowl, Ping-yeh Chiang, Micah Goldblum, Jonas Geiping, Arpit Bansal, Wojtek Czaja, and Tom Goldstein. Preventing unauthorized use of proprietary data: Poisoning for secure dataset release. *arXiv preprint arXiv:2103.02683*, 2021a.

Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojtek Czaja, and Tom Goldstein. Adversarial examples make strong poisons. *arXiv preprint arXiv:2106.10807*, 2021b.

Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Isabelle Guyon. Design of experiments of the nips 2003 variable selection benchmark. In *NIPS 2003 workshop on feature extraction and feature selection*, 2003.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition*, 2020.

Katherine L Hermann and Andrew K Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *arXiv preprint arXiv:2006.12433*, 2020.

Kashmir Hill. The secretive company that might end privacy as we know it. *The New York Times*, 2020.

Kashmir Hill and Aaron Krolik. How photos of your kids are powering surveillance technology. *The New York Times*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. *International Conference on Learning Representations*, 2021.

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *arXiv preprint arXiv:2004.00225*, 2020.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017.

Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 1989.

Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*. PMLR, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *ACM Workshop on Artificial Intelligence and Security*, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *arXiv preprint arXiv:2010.08138*, 2020.

Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*, 2019.

Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? *arXiv preprint arXiv:2006.16923*, 2020.

Evani Radiya-Dixit and Florian Tramèr. Data poisoning won't save you from facial recognition. *arXiv preprint arXiv:2106.14851*, 2021.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, 2021.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. Fawkes: Protecting privacy against unauthorized deep learning models. In *USENIX Security Symposium*, 2020.

Juncheng Shen, Xiaolei Zhu, and De Ma. Tensorclog: An imperceptible poisoning attack on deep neural network applications. *IEEE Access*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Lue Tao, Lei Feng, Jinfeng Yi, Sheng-Jun Huang, and Songcan Chen. Provable defense against delusive poisoning. *arXiv preprint arXiv:2102.04716*, 2021.

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

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019.

Chia-Hung Yuan and Shan-Hung Wu. Neural tangent generalization attacks. In *International Conference on Machine Learning*. PMLR, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, 2019.

A GENERATE SYNTHETIC DATA FROM NORMAL DISTRIBUTIONS

We adopt an algorithm in Guyon (2003) to generate synthetic data from normal distributions. Algorithm 2 shows the pseudocode. The algorithm places data of different classes at different vertices of a hypercube. It also introduce different random covariance to each class.

Algorithm 2: Generating Synthetic Data From Normal Distributions

Input: number of classes k, number of examples in each class $\{n_i\}_{i=1}^k$, data dimension d.
Create a d-dimensional hypercube.
for $i = 1$ **to** k **do**
  Generate $\boldsymbol{D}^{(i)} \in \mathbb{R}^{n_i \times d}$, where each row of $\boldsymbol{D}^{(i)}$ is sampled from $\mathcal{N}(0, \boldsymbol{I}_{d\times d})$.
  // Introduce random covariance among columns.
  Uniformly sample the elements of $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ from $[-1, 1]$.
  Compute $\boldsymbol{D}^{(i)} = \boldsymbol{D}^{(i)}\boldsymbol{A}$.
  Randomly choose an unused vertex and let $\boldsymbol{c}^{(i)} \in \mathbb{R}^d$ be its coordinates.
  // Move the sampled points to the chosen vertex.
  Compute $\boldsymbol{D}^{(i)} = \boldsymbol{D}^{(i)} + \boldsymbol{c}^{(i)}$, i.e., $\boldsymbol{c}^{(i)}$ is added to each row of $\boldsymbol{D}^{(i)}$.
  Assign the rows of $\boldsymbol{D}^{(i)}$ with label i.
end for

B IMPLEMENTATION DETAILS OF EXPERIMENTS

Implementation details of the experiments in Section 3. We use the Stochastic Gradient Descent (SGD) optimizer with a momentum coefficient 0.9 for all experiments. For all datasets, we use a batchsize of 128. The learning rates of all models are set to follow the choices in the original papers (Simonyan & Zisserman, 2014; He et al., 2016; Huang et al., 2017). The learning rate for ResNet and DenseNet models is 0.1. The learning rate for VGG models is 0.01. All models are trained for 100 epochs. The learning rate is divided by 10 at epoch 50 and 75.

Implementation details of the experiments in Section 4. To fine-tune the pre-trained model on the CIFAR-10 dataset, we first use bicubic upsampling to resize the image into 224×224. For the first approach, we use the same hyperparameters as Section 3 to fine-tune the full model. For the second approach, we first extract the features of all samples and train linear classifiers on the extracted features. The linear classifiers are trained with the L-BFGS optimizer for 50steps.

C THE EFFECT OF THE PADDING OPERATION IN ALGORITHM 1

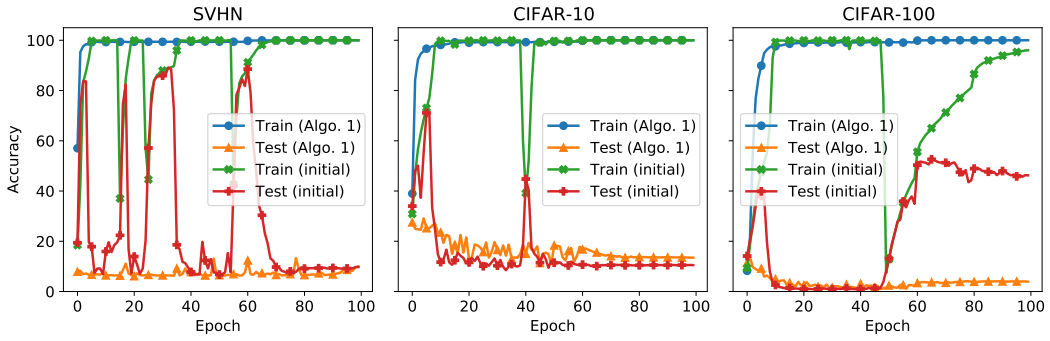

Figure 5: Training curves of ResNet18 models trained on SVHN, CIFAR-10, and CIFAR-100 datasets. The word "initial" denotes the synthetic perturbations are generated directly using Algorithm 2.

Here we explain why we duplicate each dimension of the initial data points into two-dimensional patches in Algorithm 1. Intuitively, it is more convenient to directly generate synthetic perturbations

that have the same dimension as the original images. We will show this straightforward approach has unstable performance when common data augmentation methods are applied.

We implement the above straightforward by directly using the output of Algorithm 2, i.e., the dimension of synthetic data is the same as the dimension of flattened images and we simply reshape the synthetic data into two-dimension. Other configurations are the same as those in Section 3. The models are trained with standard augmentation methods including random crop and flipping. We compare this straightforward approach with the one that further processes the outputs with Algorithm 1. The training curves of the target models are plotted in Figure 5. When using the straightforward approach, the test accuracy sometimes increases to a high point which violates the requirement of indiscriminate poisoning attacks.