Class-Oriented Poisoning Attack

Bingyin Zhao
Clemson University
bingyiz@clemson.edu

Yingjie Lao
Clemson University
ylao@clemson.edu

Abstract

Poisoning attacks on machine learning systems compromise the model performance by deliberately injecting malicious samples in the training dataset to influence the training process. Prior works focus on either availability attacks (i.e., lowering the overall model accuracy) or integrity attacks (i.e., enabling specific instance based backdoor). In this paper, we advance the adversarial objectives of the availability attacks to a per-class basis, which we refer to as *class-oriented* poisoning attacks. We demonstrate that the proposed attack is capable of forcing the corrupted model to predict in two specific ways: (i) classify unseen new images to a targeted "supplanter" class, and (ii) misclassify images from a "victim" class while maintaining the classification accuracy on other non-victim classes. To maximize the adversarial effect, we propose a gradient-based framework that manipulates the logits to retain/eliminate the desired/undesired feature information in the generated poisoning images. Using newly defined metrics at the class level, we illustrate the effectiveness of the proposed class-oriented poisoning attacks on various models (e.g., LeNet-5, Vgg-9, and ResNet-50) over a wide range of datasets (e.g., MNIST, CIFAR-10, and ImageNet-ILSVRC2012).

1 Introduction

In recent years, machine learning has demonstrated superior performance in various fields including computer vision [15, 23], natural language processing [2, 11, 43], autonomous vehicle [5, 9], robotics [20, 35], and healthcare [13]. However, it has also been shown that machine learning models are vulnerable to various types of attacks, such as evasion attacks [1, 7, 12, 14, 27, 36] and poisoning attacks [3, 26, 29, 33, 34, 30, 41]. Evasion attacks occur at the inference phase, which cause misclassification without altering the model. The adversarial example is one such attack, which fools a well-trained neural network to misclassify specific inputs by adding imperceptible perturbations to the original images [14]. In contrast, poisoning attacks corrupt the model by injecting malicious training data points. Compared to evasion attacks, poisoning attacks have the potential to render more severe or even catastrophic damages to real-world applications where continuously model updates are required or training data are accessible. For instance, as recommendation systems usually take customers' feedback to update the model constantly, the adversary can possibly utilize such vulnerability to manipulate recommendation content to achieve the adversarial goals.

Prior research on poisoning attacks can be broadly classified into two categories: availability attacks that degrade overall model accuracy (i.e., denial-of-service attacks) [3, 39, 26, 29, 28, 18, 34, 41, 30] and integrity attacks that cause misclassification on specific instances (i.e., backdoor attacks) [8, 33, 44, 37, 42]. However, the existing literature of poisoning availability attacks lacks an extensive study on multi-class classification. Even those considered multi-class classification, are rarely evaluated against the state-of-the-art learning models such as ResNet or large-scale dataset such as ImageNet. Furthermore, the adversarial goal of prior works on poisoning availability attack is constrained only in degrading the overall accuracy.

Preprint. Under review.

In this paper, we extend the literature by developing and applying novel poisoning availability attack strategies against deep neural networks to a per-class basis. We advance the adversarial objectives by formulating two attack tasks: (i) corrupting the overall model performance by forcing the model to classify all or most new inputs as a targeted class, which is denoted as the *supplanter class*, and (ii) corrupting performance of a specific class, which is named as the *victim class*, while retaining the accuracy of other classes. We propose an efficient gradient-based framework for the poisoning availability attacks, which generates more effective poisoned data samples by enlarging/dwindling the feature information of images. The idea is partially inspired by the explanation of probabilities that are assigned to incorrect answers [16]. We leverage this inherent structural similarity among different classes to develop the novel class-oriented poisoning attacks.

Our Contributions. In this paper, we made the following technical contributions.

- This paper introduces the concept of the class-oriented poisoning attack, which manipulates the model behaviour on a per-class basis.
- While the majority of prior work focuses on binary classification tasks and conventional learning algorithms such as regression and support vector machine, our poisoning availability attacks extend to multi-class classification tasks with deep neural networks.
- We define two new performance metrics at the class level: *change-to-target (CTT)* rate and *change-from-target (CFT)* rate, for evaluating the effectiveness of poisoning availability attacks, along with the overall accuracy degradation.
- This paper, to the best of our knowledge, takes the first step to systematically evaluate the effectiveness of poisoning availability attack on a large-scale dataset (ImageNet).

2 Problem Settings

2.1 Poisoning Availability Attack

Related work. The existing literature has exploited the poisoning availability attack against various learning algorithms such as clustering [4], LASSO [40], collaborative filtering [24] and SVM [3]. The closest works to this paper are [29] and [41], which extended poisoning availability attack to multi-class classification against regression models and shallow neural networks, respectively. However, these are not class-oriented. The prior works have also developed a series of approaches for poisoned data generation, including influence function [21], minimax duality [22], Karush-Kuhn-Tucker (KKT) conditions [26], autoencoder [41] and generative adversarial network (GAN) [30]. Different from these prior works, we propose a novel approach for class-oriented poisoned data generation by retaining/eliminating feature information in images, while targeting at multi-class classification tasks with deep neural networks.

We consider a continuous or online learning scenario where data samples arrive in a sequential order. Let $\boldsymbol{x} \in \mathcal{X} (\mathcal{X} \in \mathbb{R}^d)$ be a d-dimensional input and $y \in \mathcal{Y}$ be the corresponding label for the input. The objective of the classification task is to build up the mapping $\mathcal{F}$: $\mathcal{X} \to \mathcal{Y}$. We denote the pre-trained base classifier's parameters as θ. The model parameters are updated to θ^* with the incoming new steam of data: $\theta \xrightarrow{(\boldsymbol{x},y)} \theta^*$.

Poisoning availability attacks are typically formulated as a bi-level optimization problem:

$$\arg\max_{\theta^* \in \Theta} \sum_{(\boldsymbol{x},y) \in \mathcal{D}_{tr}} L\left[\mathcal{F}_{\theta^*}(\boldsymbol{x}), y, \theta^*\right] \tag{1}$$

$$\textbf{s.t.} \quad \theta^* \in \arg\min_{\theta^* \in \Theta} \sum_{(\boldsymbol{x},y) \in \mathcal{D}_{tr} \cup \mathcal{D}_p} L\left[\mathcal{F}_{\theta}(\boldsymbol{x}), y, \theta^*\right], \tag{2}$$

where D_{tr} is the clean training dataset, D_p is the poisoned dataset, Θ is the possible parameter space, and $L[\cdot]$ is the loss function. The outer maximization stands for adversarial objective, while the inner minimization represents the training process with poisoned dataset.

2.2 Class-Oriented Adversarial Objectives

Rather than only focus on maximizing the overall loss, we take the first step towards extending the adversarial objective to a per-class basis. We impose two new adversarial objectives in addition to the goal of degrading overall accuracy, which are formulated as two optimization problems accordingly.

Problem 1: All-Supplanter (AS) attack. The goal of the AS attack is to misclassify all or most inputs as a targeted object class, which we named as *supplanter class*. For a broader real-world example, the adversary can apply the AS attack to compromise a recommendation system such that no matter what the customer is looking for, only a particular item/brand (the supplanter class) will be recommended. The AS attack problem can be formulated as:

$$\underset{\theta^* \in \Theta}{\arg\max} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{tr}} L\left[\mathcal{F}_{\theta^*}\left(\boldsymbol{x}\right), y, \theta^*\right] \tag{3}$$

$$\textbf{s.t. } \theta^* \in \underset{\theta^* \in \Theta}{\arg\min} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{tr}\cup\mathcal{D}_p} L\left[\mathcal{F}_{\theta}\left(\boldsymbol{x}\right), y, \theta^*\right] \tag{4}$$

$$\theta^* \in \underset{\theta^* \in \Theta}{\arg\min} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{tr}, y=y_s} L\left[\mathcal{F}_{\theta^*}\left(\boldsymbol{x}\right), y, \theta^*\right], \tag{5}$$

where y_s represents the label of the supplanter class.

Problem 2: Only-Victim (OV) attack. The goal of the OV attack is to compromise the classification accuracy for only the inputs from a specific class, which is denoted as *victim class*. For a broader example again, the adversary can apply the OV attack to a high privilege class in an access control system, resulting in a denial-of-service to only the users in this class. The OV attack problem is formulated as:

$$\underset{\theta^* \in \Theta}{\arg\max} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{tr}, y=y_v} L\left[\mathcal{F}_{\theta^*}\left(\boldsymbol{x}\right), y, \theta^*\right] \tag{6}$$

$$\textbf{s.t. } \theta^* \in \underset{\theta^* \in \Theta}{\arg\min} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{tr}\cup\mathcal{D}_p} L\left[\mathcal{F}_{\theta}\left(\boldsymbol{x}\right), y, \theta^*\right] \tag{7}$$

$$\theta^* \in \underset{\theta^* \in \Theta}{\arg\min} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{tr}, y\neq y_v} L\left[\mathcal{F}_{\theta^*}\left(\boldsymbol{x}\right), y, \theta^*\right], \tag{8}$$

where y_v is the label of the victim class.

2.3 Class-Oriented Evaluation Metric

We propose two class-oriented evaluation metrics to assess the performance of the proposed class-oriented poisoning attacks.

Change-to-Target (CTT) rate is designed as an evaluation metric for the **AS attack**, which indicates the percentage of images that are classified as a targeted supplanter class ($\mathcal{C}_s$) due to the poisoning attack. CTT rate for a class $\mathcal{C}_k$ is formally defined over a test dataset $\mathcal{D}_t$ as:

$$CTT(\mathcal{C}_k) = \frac{\sum_{\substack{(\boldsymbol{x_i},y_i)\in\mathcal{D}_t \\ y_i=y_k}} \left(\mathcal{F}_{\theta^*}(\boldsymbol{x_i}) == y_s\right) - \sum_{\substack{(\boldsymbol{x_i},y_i)\in\mathcal{D}_t \\ y_i=y_k}} \left(\mathcal{F}_{\theta}(\boldsymbol{x_i}) == y_s\right)}{N_k}, \tag{9}$$

where N_k is the total number of images in the class $\mathcal{C}_k$ and y_k is the corresponding categorical label, while $\mathcal{F}_\theta$ and $\mathcal{F}_{\theta^*}$ are the model mappings before and after the poisoning attack, respectively. Then, the overall CTT rate over $\mathcal{D}_t$ can be calculated by weighted averaging the CTT rates of all non-supplanter classes:

$$CTT = \frac{\sum_{\substack{k=1 \\ k\neq s}}^{K} \left(N_k \cdot CTT(\mathcal{C}_k)\right)}{\sum_{\substack{k=1 \\ k\neq s}}^{K} N_k}, \tag{10}$$

where K is the total number of classes.

Change-from-Target (CFT) rate is specifically used for evaluating the **OV attack**, which indicates how many images from a targeted class are misclassified due to the poisoning attack. Similarly, CFT rate for a class $\mathcal{C}_k$ is defined by Equation (11).

$$CFT(\mathcal{C}_k) = \frac{\sum_{\substack{(\boldsymbol{x_i},y_i)\in\mathcal{D}_t \\ y_i=y_k}} \left(\mathcal{F}_{\theta}(\boldsymbol{x_i}) == y_k\right) - \sum_{\substack{(\boldsymbol{x_i},y_i)\in\mathcal{D}_t \\ y_i=y_k}} \left(\mathcal{F}_{\theta^*}(\boldsymbol{x_i}) == y_k\right)}{N_k} \tag{11}$$

2.4 Adversarial Model

Following the recent works on poisoning availability attack [21, 26], we consider the same adversarial model where the adversary has the knowledge of the learning algorithm and model architecture. Meanwhile, the attacker is assumed to be capable of injecting crafted poisoned data and assigning labels to poisoned data. Although the assumption is less practical in the real-world applications, it allows us to assess the effectiveness of poisoning attacks in the worst-case scenario, which is the same as in the literature of poisoning availability attacks [6, 3, 30, 24, 29].

3 Class-Oriented Poisoning Attack Methods

3.1 All-Supplanter Attack

Intuitively, we expect retraining images with the label of the supplanter class, similar to the flipped-label poisoning attack [39], would have the potential to shift predictions of all classes towards the supplanter class. However, our experimental results demonstrate that such attacks, when applied to multi-class classification tasks, neither effectively degrade the overall accuracy nor achieve the class-oriented adversarial goal of the AS attack for the neural network models (see *Section 4*).

To improve the effectiveness of poisoning towards the supplanter class, we develop a novel method for generating the poisoned data. We leverage the fact that the probabilities assigned to other object classes of a well-trained model also reveal how much feature information of these incorrect classes are associated with the corresponding image by the model [16]. Thus, we hypothesize that if an input image only contains feature information of its ground-truth class, training such image with the supplanter class label will force the model to expand the decision boundary of the supplanter class to the maximum degree.

Inspired by prior works that exploit the penultimate layer (formally called logit outputs) to distill knowledge of neural networks [16], catch features [17], and defend against adversarial examples [19], we follow a similar methodology to control the feature information through the logit outputs. Our algorithm starts with a seed image that is arbitrarily picked from any class other than the supplanter class, and then attempts to retain the feature information associated with the original ground-truth class and reduce the feature information of other classes by enlarging/dwindling the corresponding logit outputs. We here denote $f(\cdot)$ as the logit output function of the neural network and f_{y_k} as the corresponding logit to the categorical label y_k. The objective of our algorithm can be expressed by the following minimization objective function:

$$\underset{\boldsymbol{x_p} \in D_p}{\arg\min} \Big(L[f_{y_o}(\boldsymbol{x_o}), +\infty, \theta] + \sum_{\substack{k=1 \\ k \neq o}}^{K} \lambda_k \cdot L_k[f_{y_k}(\boldsymbol{x_o}), -\infty, \theta] \Big), \tag{12}$$

where $\boldsymbol{x_o}$ is the seed image, y_o is the corresponding categorical ground-truth label, $f_{y_o}(x_o)$ is the logit output of the ground-truth class, $f_{y_k}(\boldsymbol{x_o})$ is the logit output of other classes, and finally $\boldsymbol{x_p}$ stands for the poisoned image. λ_k is used to control the importance of loss terms. Alternatively, from the perspective of entropy, we expect such optimization would also reduce the following information entropy $H[\cdot]$:

$$H\big[\sigma\big(f(\boldsymbol{x_p})\big)\big] = -\sum_{k=1}^{K} p_k \cdot log(p_k) \to 0, \tag{13}$$

where p_k is the probability for each class that converted from the logit f_{y_k} using *softmax*.

However, solving the minimization problem can be computationally expensive, especially for high-dimensional datasets that have hundreds or thousands of classes. Based on the facts that (i) the classification result is determined by the largest probability and (ii) only the supplanter class is the target, we consider an approximation that simplifies the task to retain feature information of the ground-truth class and eliminate feature information of the supplanter class. In other words, we only focus on the two most important classes instead of all classes, i.e.,

$$\underset{\boldsymbol{x_p} \in D_p}{\arg\min} \big(L[f_{y_o}(\boldsymbol{x_o}), +\infty, \theta] + \lambda \cdot L[f_{y_s}(\boldsymbol{x_o}), -\infty, \theta] \big). \tag{14}$$

By solving the minimization using gradient descent, the poisoned image $\boldsymbol{x_p}$ is updated through one backward pass:

$$\boldsymbol{x_p} = \boldsymbol{x_o} - \epsilon \cdot sign\Big(\nabla_{\boldsymbol{x_o}}\big(L[f_{y_o}(\boldsymbol{x_o}), +\infty, \theta] + \lambda \cdot L[f_{y_s}(\boldsymbol{x_o}), -\infty, \theta]\big)\Big), \tag{15}$$

where ϵ is a hyper-parameter chosen by the attacker. This update step can also be executed for several rounds to more aggressively increase the feature information of the ground-truth class and reduce the feature information of the supplanter class. Algorithm 1 presents the details of poisoned data generation for the AS attack.

Algorithm 1: Poisoned Data Generation (All-Supplanter)

Input: $\boldsymbol{x_o}$: seed image, y_o: seed image label, y_s: supplanter class label
T: max number of optimization iterations
hyper-parameters λ, ϵ
Output: poisoned image $\boldsymbol{x_p}$, poisoned label y_p

Initialization: $\boldsymbol{x_{p_0}} = \boldsymbol{x_o} - \epsilon \cdot sign\Big(\nabla_{\boldsymbol{x_o}}\big(L[f_{y_o}(\boldsymbol{x_o}), +\infty, \theta] + \lambda \cdot L[f_{y_s}(\boldsymbol{x_o}), -\infty, \theta]\big)\Big)$
while *t*<*T* **do**
  Compute the gradient: $\nabla = \nabla_{\boldsymbol{x_{p_t}}}\big(L[f_{y_o}(\boldsymbol{x_{p_t}}), +\infty, \theta] + \lambda \cdot L[f_{y_s}(\boldsymbol{x_{p_t}}), -\infty, \theta]\big)$
  Update the poisoned image: *Clip*$\{\boldsymbol{x_{p_{t+1}}} = \boldsymbol{x_{p_t}} - \epsilon \cdot sign(\nabla)\}$
  if $f_{y_s}(\boldsymbol{x_{p_{t+1}}}) > f_{y_s}(\boldsymbol{x_{p_t}})$ ***or*** $f_{y_o}(\boldsymbol{x_{p_{t+1}}}) < f_{y_o}(\boldsymbol{x_{p_t}})$ **then**
    break
end
Assign $y_p = y_s$
Return $\boldsymbol{x_p}, y_p$

3.2 Only-Victim Attack

Compared to the AS attack, the OV attack is fundamentally more challenging, as it requires to not only compromise the accuracy of the targeted victim class but also maintain the performance of the other classes, i.e., high CFT rate for the victim class while low CFT rates for the other classes. However, as observed in our preliminary experiments as well as prior poisoning availability attacks [25, 31], poisoning with data from only one class will also largely shift the distribution of other classes. Hence, a set of poisoned data from more than one class is necessary to achieve the adversarial goal of the OV attack. Intuitively, simply training images from all classes except the victim class may achieve this adversarial goal. However, our preliminary results reveal that such methods always require a huge amount of training data and very large training epochs, which is very inefficient in achieving the adversarial objective.

We propose another gradient-based algorithm for the OV attack. The entire procedure of our poisoned data generation is presented in Algorithm 2. We modify the method from the AS attack to craft the poisoned dataset as follows: (i) pick a same number of arbitrary images from each class, (ii) enlarge/dwindle feature information of the corresponding classes for each image, as detailed in Algorithm 2, and (iii) assign the original ground-truth labels to the non-victim classes and the targeted label y_p to the victim class. Specifically, for images from the victim class, we apply the same operations as in Algorithm 1, while for images from other classes, we only increase the feature information of their original classes. The objective of OV attack can be expressed as:

$$\begin{cases} \underset{\boldsymbol{x_p} \in D_p}{\arg\min} \big(L[f_{y_v}(\boldsymbol{x_o}), +\infty, \theta] + \lambda \cdot L[f_{y_p}(\boldsymbol{x_o}), -\infty, \theta]\big), & \text{if } \boldsymbol{x_o} \in \mathcal{C}_v \\ \underset{\boldsymbol{x_p} \in D_p}{\arg\min} \big(L[f_{y_o}(\boldsymbol{x_o}), +\infty, \theta]\big), & \text{otherwise} \end{cases} \tag{16}$$

Similarly, the poisoned images are updated through a backward pass in Equation (17):

$$\boldsymbol{x_p} = \begin{cases} \boldsymbol{x_o} - \epsilon \cdot sign\Big(\nabla_{\boldsymbol{x_o}}\big(L[f_{y_v}(\boldsymbol{x_o}), +\infty, \theta] + \lambda \cdot L[f_{y_p}(\boldsymbol{x_o}), -\infty, \theta]\big)\Big), & \text{if } \boldsymbol{x_o} \in \mathcal{C}_v \\ \boldsymbol{x_o} - \epsilon \cdot sign\Big(\nabla_{\boldsymbol{x_o}}\big(L[f_{y_o}(\boldsymbol{x_o}), +\infty, \theta]\big)\Big), & \text{otherwise} \end{cases} \tag{17}$$

Algorithm 2: Poisoned Data Generation (Only-Victim)

Input: $\boldsymbol{x_k} \in \boldsymbol{X}$: the set of seed images, $y_k \in Y$: the set of labels associated to X
y_p: poisoned label, T: max number of optimization iteration
hyper-parameters λ, ϵ
Output: poisoned dataset $\boldsymbol{X_p}$, poisoned label set Y_p

if $\boldsymbol{x_k} \in \mathcal{C}_v$ **then**
  Apply Algorithm 1
  Assign $y_k = y_p$
else
  Initialization: $\boldsymbol{x_{p_0}} = \boldsymbol{x_o} - \epsilon \cdot sign\Big(\nabla_{\boldsymbol{x_o}}\big(L[f_{y_o}(\boldsymbol{x_o}), +\infty, \theta]\big)\Big)$
  while $t < T$ **do**
    Compute the gradient: $\nabla = \nabla \boldsymbol{x_{p_t}}\big(L[f_{y_o}(\boldsymbol{x_{p_t}}), +\infty, \theta]\big)$
    Update the poisoned image: *Clip*$\{\boldsymbol{x_{p_{t+1}}} = \boldsymbol{x_{p_t}} - \epsilon \cdot sign(\nabla)\}$
  end
end
Assign $\boldsymbol{X_p} = \boldsymbol{X}$; $Y_p = Y$
Return $\boldsymbol{X_p}$, Y_p

4 Experiments

4.1 Experimental Settings

We apply our proposed poisoning attacks to multi-class image classification tasks using three widely-used datasets (MNIST, CIFAR-10, and ImageNet-ILSVRC2012) against classic neural network models (LeNet-5, Vgg-9, and ResNet-50, respectively). All the evaluations are performed on the corresponding validation datasets. For comparison with prior works, we implement the flipped-label attack (FL) [39] and the direct gradient method (DGM) [41] as the baseline attacks for MNIST and CIFAR-10. We also examine our poisoning availability attacks on ImageNet against a pre-trained ResNet-50 model [15], which we hope to serve as a baseline comparison for future works of poisoning availability attacks. To illustrate the effectiveness of our methods as well as facilitate a fair comparison, we minimize the impact of significant model shifts due to large learning rates and better align with practical continuous or online learning scenarios, by setting the initial learning rate of the poisoning attack close to the final learning rate of the base model and applying the same decay strategy during the attack. All networks are implemented with TensorFlow and experiments are run on NVIDIA Tesla V100 GPUs.

4.2 Experimental Results of the All-Supplanter Attack

To examine the true power of our AS attack, we only generate a single instance for the poisoning process. We present the results of MNIST and CIFAR-10 in Figure 1, and ImageNet in Table 1. Hyper-parameters in Algorithm 1 are set as: $\lambda = 1, \epsilon = 0.3$ for MNIST and CIFAR-10, and $\lambda = 1, \epsilon = 0.5$ for ImageNet. It can be seen that our proposed AS attack is highly effective in degrading the overall model accuracy as well as increasing the CTT rate for the supplanter class. For example, we improve the test error from 20% to above 80% and the CTT rate from 10% to 99% on the CIFAR-10 dataset within less than 20 attack iterations. On the other hand, our method requires much fewer iterations to reach at the maximum poisoning effect than the baseline attack, whose test error and CTT rate are only increased by $\sim$20% and $\sim$10%, respectively, even after 50 attack iterations. For ImageNet, the proposed AS attack achieves a CTT rate of 94.07% within 10 attack iterations. Note that including more poisoned data in each attack iteration will obviously enhance the poisoning effect of the proposed AS attack.

For scenarios where the adversary has no knowledge of the training process, it is also important to study the impact of different learning rates on the attack. Our conclusion is consistent with prior works [38] that smaller learning rates yield less effectiveness for the attack. However, while it requires more attack iterations to arrive at the maximum poisoning effect, our method still significantly outperforms the baseline attacks when learning rate is small, as indicated in Figure 1(c) and Figure 1(d).

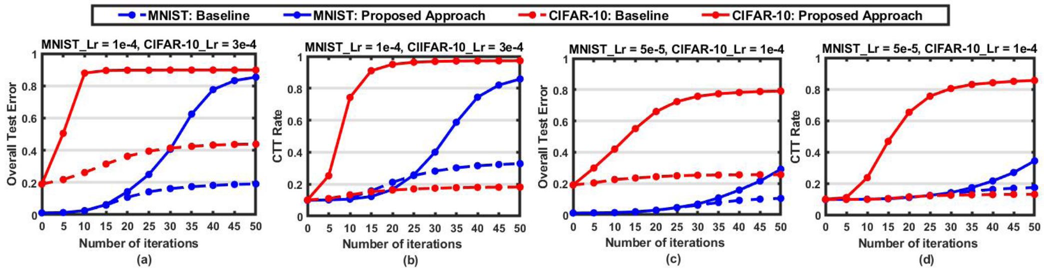

Figure 1: Error and CTT rate comparison. Classes ‘4’ and ‘deer’ are selected as the supplanter class for MNIST and CIFAR-10, respectively. (a) and (c) are error rates, while (b) and (d) are CTT rates with different learning rates.

Table 1: Comparison of CTT and accuracy before and after the AS attack on ImageNet.

	Top-1 Accuracy	Top-5 Accuracy	CTT
Vanilla ResNet-50	74.87%	92.02%	–
Poisoned by AS Attack	2.15%	16.91%	94.07%

4.3 Experimental Results of the Only-Victim Attack

Since the adversarial goal of the OV attack is to subvert only one class without degrading the performance of other classes, there are two important metrics for this task: 1) CFT rate of the victim class should be as high as possible, and 2) CFT rates of the non-victim classes should be as low as possible. Note that the highest achievable CFT rate is upper-bounded by the accuracy of the victim class in the vanilla model. We use the same hyper-parameters as in the AS attack. The poisoned images in the OV attack only account for 10% of the training data. We present the CFT rate of each class by poisoning attacks on CIFAR-10 in Table 2, where class ‘frog’ is selected as the victim class. It can be seen that DGM is able to keep the CFT rates of all non-victim classes relatively low; however, its CFT rate of the victim class is only 39.2%. While the FL attack achieves a CFT rate of 85.1% for the victim class, it also significantly increases the CFT rates of non-victim classes, which reveals the difficulty of achieving the class-oriented adversarial goal by using prior poisoning attack methods. Compared to the baseline attacks, our proposed approach can effectively increase the CFT rate of the victim class and also retain the CFT rates of the non-victim classes to be less than 6.5%. Besides, in some cases, the proposed approach even improves the accuracy of some non-victim classes, which is indicated by the negative CFT rates in our experimental results. This performance is beyond our expectation, which however does not contradict the class-oriented OV adversarial objective, i.e., only degrading the accuracy of the victim class.

For ImageNet, due to the large number of object classes, we present the distribution of the final CFT rates, as shown in Figure 2. While successful in maintaining low CFT rates for non-victim classes, our method achieves a CFT rate of 62% for the victim class on ImageNet. Note that we can improve the performance by increasing the percentage of poisoning images in the training data. Therefore, it can be concluded that the proposed OV attack, although still accomplishes the adversarial goal, is less effective when compared to the performance on CIFAR-10. A possible rationale behind this phenomena is that it becomes harder to completely decouple the feature information of one class from all other classes during the poisoning, with the number of classes scaling up. On the other hand, our experimental results also reveal the importance of studying poisoning availability attack on large-scale/dimensional dataset, which is lacking in the existing literature.

5 Possible Defenses

Since the main objective of this paper is to extend the attack to a per-class basis rather than enhance the adversarial effect against various defensive solutions, in general, we expect the proposed attacks to have similar performance as prior poisoning availability attacks, when evaluated under possible defenses.

Table 2: CFT rate of each CIFAR-10 class by poisoning.

Class / Attack	airplane	automobile	bird	cat	deer
DGM	0.1%	-0.1%	12.0%	18.2%	3.7%
FL	19.5%	15.0%	51.9%	53.3%	-21.3%
OV (proposed)	-0.2%	2.9%	6.5%	-3.9%	-1.4%
Class / Attack	**dog**	**frog (victim)**	**horse**	**ship**	**truck**
DGM	6.8%	39.2%	-0.1%	2.1%	-4.3%
FL Attack	41.1%	85.1%	26.3%	13.2%	19.3%
OV (proposed)	5.0%	71.6%	1.4%	-1.7%	3.1%

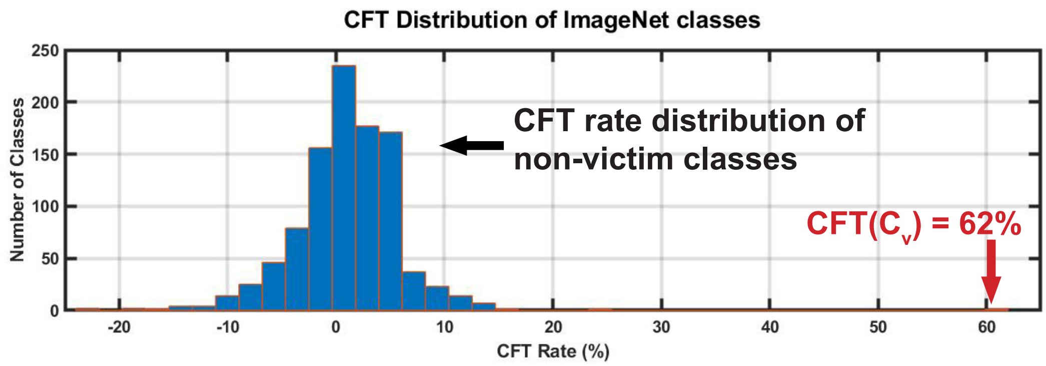

Figure 2: The CFT rate distribution of ImageNet classes by the OV attack.

Data sanitization is a defensive technique against poisoning attack that works by distinguishing and removing outliers (poisoned data) from the training dataset [10, 32]. However, it has been shown that a broad range of data sanitization can be easily compromised or bypassed [22]. Therefore, we can also leverage such techniques for our proposed attack to evade the detection. In fact, in most of the recent works on poisoning attacks, data sanitization is no longer considered as a certified defensive strategy [22, 30, 18, 38].

Alternatively, a possible countermeasure is to periodically check the accuracy and/or loss of the learning models [29, 41]. Although expensive in terms of cost and time, these approaches are intuitively effective based on the fact that poisoning availability attack aims at degrading the accuracy. Therefore, the threat models of poisoning availability attacks usually consider the retraining or learning process to be unmonitored, which makes these countermeasures impractical. We expect the worst-case performance of our proposed methods against such countermeasures to be consistent with prior poisoning availability attacks [26, 24, 29, 28, 41, 30].

Since the poisoned data are tailored to maliciously influence the training process of a learning model, we suggest exploiting averaged stochastic gradient classifier [38] and combinational models, such as bagging [24], where the classification results are no longer dependent on a single model, to defend against the attack. However, the overhead for deploying multiple classifiers should also be carefully considered.

6 Conclusions

This paper introduced the concept of class-oriented poisoning attack. We formulated two attack problems, i.e., “All-Supplanter” and “Only-Victim”, which seek to compromise the model behavior on a per-class basis. Accordingly, we defined two new metrics to evaluate the performance of poisoning attacks at the class level. Our proposed gradient-based algorithms successfully achieved the class-oriented adversarial objectives through manipulating the feature information contained in images for poisoned data generation. The effectiveness of the proposed methods are comprehensively studied in our experiments.

References

[1] Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)

[2] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

[3] Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)

[4] Biggio, B., Pillai, I., Rota Bulò, S., Ariu, D., Pelillo, M., Roli, F.: Is data clustering in adversarial settings secure? In: Proceedings of the 2013 ACM workshop on Artificial intelligence and security. pp. 87–98 (2013)

[5] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)

[6] Burkard, C., Lagesse, B.: Analysis of causative attacks against svms learning from data streams. In: Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics. pp. 31–36 (2017)

[7] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE (2017)

[8] Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., Srivastava, B.: Detecting backdoor attacks on deep neural networks by activation clustering. arXiv preprint arXiv:1811.03728 (2018)

[9] Chen, C., Seff, A., Kornhauser, A., Xiao, J.: Deepdriving: Learning affordance for direct perception in autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2722–2730 (2015)

[10] Cretu, G.F., Stavrou, A., Locasto, M.E., Stolfo, S.J., Keromytis, A.D.: Casting out demons: Sanitizing training data for anomaly sensors. In: 2008 IEEE Symposium on Security and Privacy (sp 2008). pp. 81–95. IEEE (2008)

[11] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[12] Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9185–9193 (2018)

[13] Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature **542**(7639), 115 (2017)

[14] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)

[15] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[16] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)

[17] Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: Advances in Neural Information Processing Systems. pp. 125–136 (2019)

[18] Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., Li, B.: Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 19–35. IEEE (2018)

[19] Kannan, H., Kurakin, A., Goodfellow, I.: Adversarial logit pairing. arXiv preprint arXiv:1803.06373 (2018)

[20] Kober, J., Bagnell, J.A., Peters, J.: Reinforcement learning in robotics: A survey. The International Journal of Robotics Research **32**(11), 1238–1274 (2013)

[21] Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1885–1894. JMLR. org (2017)

[22] Koh, P.W., Steinhardt, J., Liang, P.: Stronger data poisoning attacks break data sanitization defenses. arXiv preprint arXiv:1811.00741 (2018)

[23] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

[24] Li, B., Wang, Y., Singh, A., Vorobeychik, Y.: Data poisoning attacks on factorization-based collaborative filtering. In: Advances in neural information processing systems. pp. 1885–1893 (2016)

[25] Liu, J., Zhu, X.: The teaching dimension of linear learners. The Journal of Machine Learning Research **17**(1), 5631–5655 (2016)

[26] Mei, S., Zhu, X.: Using machine teaching to identify optimal training-set attacks on machine learners. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)

[27] Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)

[28] Mozaffari-Kermani, M., Sur-Kolay, S., Raghunathan, A., Jha, N.K.: Systematic poisoning attacks on and defenses for machine learning in healthcare. IEEE journal of biomedical and health informatics **19**(6), 1893–1905 (2014)

[29] Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E.C., Roli, F.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp. 27–38. ACM (2017)

[30] Muñoz-González, L., Pfitzner, B., Russo, M., Carnerero-Cano, J., Lupu, E.C.: Poisoning attacks with generative adversarial nets. arXiv preprint arXiv:1906.07773 (2019)

[31] Park, S., Weimer, J., Lee, I.: Resilient linear classification: an approach to deal with attacks on training data. In: Proceedings of the 8th International Conference on Cyber-Physical Systems. pp. 155–164 (2017)

[32] Rubinstein, B.I., Nelson, B., Huang, L., Joseph, A.D., Lau, S.h., Rao, S., Taft, N., Tygar, J.D.: Antidote: understanding and defending against poisoning of anomaly detectors. In: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement. pp. 1–14 (2009)

[33] Shafahi, A., Huang, W.R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., Goldstein, T.: Poison frogs! targeted clean-label poisoning attacks on neural networks. In: Advances in Neural Information Processing Systems. pp. 6103–6113 (2018)

[34] Steinhardt, J., Koh, P.W.W., Liang, P.S.: Certified defenses for data poisoning attacks. In: Advances in neural information processing systems. pp. 3517–3529 (2017)

[35] Sünderhauf, N., Brock, O., Scheirer, W., Hadsell, R., Fox, D., Leitner, J., Upcroft, B., Abbeel, P., Burgard, W., Milford, M., et al.: The limits and potentials of deep learning for robotics. The International Journal of Robotics Research **37**(4-5), 405–420 (2018)

[36] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)

[37] Turner, A., Tsipras, D., Madry, A.: Label-consistent backdoor attacks. arXiv preprint arXiv:1912.02771 (2019)

[38] Wang, Y., Chaudhuri, K.: Data poisoning attacks against online learning. arXiv preprint arXiv:1808.08994 (2018)

[39] Xiao, H., Xiao, H., Eckert, C.: Adversarial label flips attack on support vector machines. In: ECAI. pp. 870–875 (2012)

[40] Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C., Roli, F.: Is feature selection secure against training data poisoning? In: International Conference on Machine Learning. pp. 1689–1698 (2015)

[41] Yang, C., Wu, Q., Li, H., Chen, Y.: Generative poisoning attack method against neural networks. arXiv preprint arXiv:1703.01340 (2017)

[42] Yao, Y., Li, H., Zheng, H., Zhao, B.Y.: Latent backdoor attacks on deep neural networks. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2041–2055 (2019)

[43] Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. ieee Computational intelligenCe magazine **13**(3), 55–75 (2018)

[44] Zhu, C., Huang, W.R., Shafahi, A., Li, H., Taylor, G., Studer, C., Goldstein, T.: Transferable clean-label poisoning attacks on deep neural nets. arXiv preprint arXiv:1905.05897 (2019)