CLPA: Clean-Label Poisoning Availability Attacks Using Generative Adversarial Nets

Bingyin Zhao,Yingjie Lao

Department of Electrical and Computer Engineering, Clemson University, SC, 29634, USA
{bingyiz, ylao}@clemson.edu

Abstract

Poisoning attacks are emerging threats to deep neural networks where the adversaries attempt to compromise the models by injecting malicious data points in the clean training data. Poisoning attacks target either the availability or integrity of a model. The availability attack aims to degrade the overall accuracy while the integrity attack causes misclassification only for specific instances without affecting the accuracy of clean data. Although clean-label integrity attacks are proven to be effective in recent studies, the feasibility of clean-label availability attacks remains unclear. This paper, for the first time, proposes a clean-label approach, **CLPA**, for the poisoning availability attack. We reveal that due to the intrinsic imperfection of classifiers, naturally misclassified inputs can be considered as a special type of poisoned data, which we refer to as "natural poisoned data". We then propose a two-phase generative adversarial net (GAN) based poisoned data generation framework along with a triplet loss function for synthesizing clean-label poisoned samples that locate in a similar distribution as natural poisoned data. The generated poisoned data are plausible to human perception and can also bypass the singular vector decomposition (SVD) based defense. We demonstrate the effectiveness of our approach on CIFAR-10 and ImageNet dataset over a variety type of models. Codes are available at: https://github.com/bxz9200/CLPA.

Introduction

In the past years, machine learning, especially deep learning has achieved remarkable advancement in a wide range of fields including computer vision (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016), natural language processing (Devlin et al. 2019; Bahdanau, Cho, and Bengio 2015), and game playing (Silver et al. 2017, 2016). Despite unprecedented progress, machine learning models are shown to be susceptible to various types of adversarial attacks such as evasion attacks (Goodfellow, Shlens, and Szegedy 2015; Clements et al. 2021; Clements and Lao 2022b), backdoor attacks (Clements and Lao 2018b,a, 2019; Saha, Subramanya, and Pirsiavash 2020), and poisoning attacks (Biggio, Nelson, and Laskov 2012), which raises serious concern of the robustness and security for the real-world deployments (Clements and Lao 2022a; Lao et al. 2022). A notable example of evasion attacks is the adversarial example which fools machine learning classifiers by adding imperceptible perturbation to a benign input. Evasion attacks occur at the inference phase and require attackers to modify model inputs. Backdoor attacks induce models to make wrong predictions on inputs embedded with backdoor triggers, for instance, a pair of glasses (Chen et al. 2017), a colored pattern (Gu, Dolan-Gavitt, and Garg 2017), or even invisible triggers (Nguyen and Tran 2021; Doan et al. 2021; Doan, Lao, and Li 2021). However, it requires the attacker to access both training and inference phase to inject and activate backdoor triggers. Poisoning attacks, on the other hand, manipulate model behavior at the training phase by injecting deliberately crafted malicious data in the training set. The adversarial goals of poisoning attack target model availability or integrity. The availability attack attempts to subvert the overall model accuracy while the integrity attack attempts to only affect the prediction results of specific inputs. Detailed comparison of these attacks is summarized in Table 1.

	Training	Inference	Adversarial Goals
Evasion Attacks	×	✓	Misclassify Specific Inputs
Backdoor Attacks	✓	✓	Misclassify Specific Inputs
Poisoning Attacks	✓	×	Degrade Overall Performance
			Misclassify Specific Inputs

Table 1: Comparison of adversarial goals and capability.

Poisoning attacks are critical threats to scenarios where attackers are able to find ways to supply new training data. For instance, web-based repositories always provide such opportunities for attackers to inject poisoned training data through malware. Benign models will be maliciously affected after training with these data. With the rapid development of deep learning, models become more complex and harder to train. Thus, it is particularly worth studying poisoning availability attacks since training with a poisoned dataset may lead to a severe loss of time and computational resources and cause a denial of service in real-world applications. Prior poisoning availability attacks usually add large distortion to poisoned samples in the pixel space (Yang et al. 2017) or assign incorrect labels (Muñoz-González et al. 2017, 2019; Xiao, Xiao,

and Eckert 2012), which will introduce larger loss to the optimization process for optimizing the adversarial effect. However, these distorted and mislabeled poisoned data can be detected by techniques such as ℓ_2 defense (Koh, Steinhardt, and Liang 2018), label sanitization (Paudice, Muñoz-González, and Lupu 2018), and even simple visual inspections. While clean-label settings (i.e., the adversary can only manipulate the data but not the label) have been extensively investigated in poisoning integrity and backdoor attacks (Shafahi et al. 2018; Huang et al. 2020; Zhu et al. 2019; Saha, Subramanya, and Pirsiavash 2020; Turner, Tsipras, and Madry 2019; Geiping et al. 2021), the implication on poisoning availability attacks however has entailed little study.

In this paper, we make the first attempt, to the best of our knowledge, to develop an algorithm for clean-label poisoning availability attack, **CLPA**. We propose to generate poisoned samples with the following four properties: 1) plausible to human beings; 2) correctly-labeled (clean-label); 3) stealthy against visual inspection and sanitization; 4) can achieve the adversarial goal (i.e., degrade the inference accuracy). However, it is challenging to systematically generate such poisoned data since these constraints greatly limit the training convergence towards adversarial goals (e.g., provide much smaller loss in the optimization). In other words, the first three properties contradict with the last property to a certain extent. To tackle this challenge, we propose a novel approach that uses generative adversarial nets (GAN) (Goodfellow et al. 2014) and introduces a carefully-designed triplet loss for poisoned data generation. We propose a two-phase framework to decouple the training of the original GAN and the GAN that generates poisoned data to ensure the quality of poisoned images. We show the proposed method is able to successfully generate desired poisoned data at scale.

Poisoning Availability Attacks

Poisoning availability attack on image classifiers was first proposed in (Biggio, Nelson, and Laskov 2012) that compromises a support vector machine (SVM) model on a binary classification task. Later on, (Biggio et al. 2013; Mei and Zhu 2015; Jagielski et al. 2018; Xiao et al. 2015) studied attack strategies on a wider range of learning models such as clustering, LASSO, and regression models. Most prior works employ gradient-based optimization for poisoned data generation. A series of approaches were developed to improve the efficiency for solving the optimization problem, including the use of with stationary Karush-Kuhn-Tucker (KKT) conditions (Mei and Zhu 2015), the back-gradient optimization (Muñoz-González et al. 2017), and approximated non-convex models to manipulate the functions (Koh and Liang 2017). However, it is still very hard and costly to generate poisoned data at scale due to the computational complexity of solving the gradient-based optimization. To address this, (Yang et al. 2017; Muñoz-González et al. 2019) proposed generative methods by using auto-encoder and GAN.

However, in these works, attackers are allowed to flip the labels or assign arbitrary labels to the poisoned data. Such poisoned data can be easily identified and removed by label sanitization (Paudice, Muñoz-González, and Lupu 2018) or manually annotating the data, which limits the practical effect of these approaches. Besides, the prior generative approaches rarely consider the cosmetic quality of poisoned data, which makes the poisoned samples easy to distinguish by visual inspection due to the large perturbation added on top of natural inputs. (Koh and Liang 2017; Feng, Cai, and Zhou 2019) proposed indistinguishable training-set attacks using adversarial training examples, which however requires a much more stringent capability for the attacker, i.e., direct modification to the original training data. In this paper, we propose a stronger and more practical poisoning availability attack where the poisoned data are labeled consistently with human annotation. While clean-label poisoned samples are shown to be effective for both backdoor (Turner, Tsipras, and Madry 2019) and poisoning integrity attacks (Shafahi et al. 2018; Huang et al. 2020; Zhu et al. 2019; Geiping et al. 2021), satisfying such a requirement is more challenging in poisoning availability attacks, as the adversary needs to alter the decision boundary as much as possible instead of only inducing malicious behaviors to specific instances.

Natural Poisoned Data

Our idea originates from the misclassified samples in the vanilla DNN model that naturally exists due to the inherent imperfection of classifiers as well as the ever-growing task complexity. Even the recent advance of vision transformer (ViT) (Dosovitskiy et al. 2021) still makes wrong predictions for a considerable amount of clean data on ImageNet. We show in Figure 1 that a Vgg16 (Simonyan and Zisserman 2015) classifier with a test accuracy of 93.56% misclassifies some test data of CIFAR-10 with apparent human-perceptible attributes. Our proposed approach leverages such deviation to stealthily compromise well-trained classifiers. We refer to authentic images misclassified by a well-trained classifier as "natural poisoned data". These "poisoned data" and the corresponding labels will not be recognized distinctively by human observers. These data can be captured from any arbitrary source in real-world scenarios, such as cameras, mobile phones, and drones.

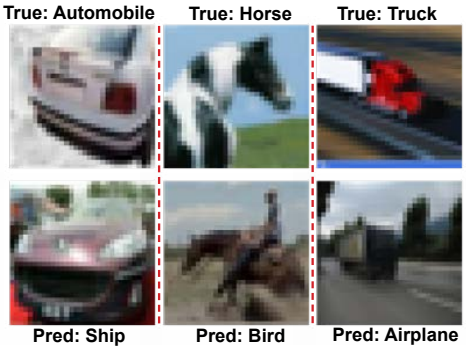

Figure 1: Examples of natural poisoned data. A well-trained classifier still makes wrong prediction with high confidence.

We analyze the feasibility of using these "natural poisoned data" to compromise the availability of a classifier. We adulterated the "natural poisoned data" in the original training data with a ratio from 0% to 100% and assigned these data

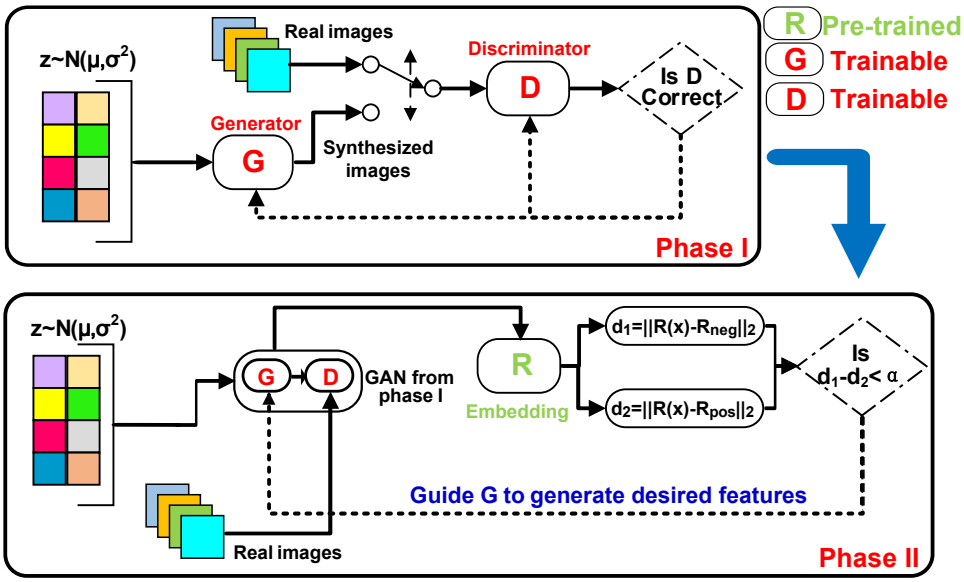

Figure 2: Illustration of the CLPA framework. Phase I is a normal GAN training process where the generator learns data distribution and the discriminator learns to distinguish real and fake data. In the second phase, an embedding of a benign classifier is used to guide the generator to learn the desired feature and generate poisoned data.

with ground-truth labels. We then trained the model and evaluated on the standard test dataset. We found that "natural poisoned data" can affect the performance of the classifier to a certain extent with different poisoning ratios. Notably, training with full "natural poisoned data" causes a test error of 50%, which demonstrates the potential of such data in clean-label poisoning availability attacks.

Clean-Label Poisoning Availability Attack

We now describe our two-phase GAN framework, **CLPA**, for clean-label poisoned data generation. Our objective is to generate images with similar characteristics as "natural poisoned data", particularly those locate in the overlapped area of multiple categories in the representation space.

Threat Model

Attacker's Goal. In this work, our goal is to compromise the availability and degrade the model performance on unseen test data after training with the poisoned dataset, which can be mathematically expressed as:

$$\begin{aligned} &\arg\max_{\mathcal{D}_p} \sum_{(x,y)\sim\hat{\mathcal{D}}} \mathcal{L}\left(x, y, \theta_p\right) \\ \textbf{s.t.} \quad &\theta_p \in \arg\min_{\theta_p\in\Theta} \sum_{(x,y)\sim\mathcal{D}_{tr}\cup\mathcal{D}_p} \mathcal{L}\left(x, y, \theta\right), \end{aligned} \tag{1}$$

where D_{tr} is the training data, D_p is the crafted poisoned data, $\hat{D}$ is the untainted test dataset, $\mathcal{L}(\cdot)$ is the loss function (e.g. the cross-entropy), θ is the parameters of a benign model and θ_p is the updated parameters over a possible space Θ. The inner minimization stands for the poisoned training while the outer maximization represents the evaluation on clean data using the poisoned model.

Attacker's Knowledge. There are two representative attack scenarios in poisoning attacks, i.e., white-box and black-box attacks. The attacker is assumed to have full knowledge of the training data, learning algorithm and model parameters in the white-box setting. The perfect knowledge setting is wildly considered in previous work for both availability and integrity attacks (Muñoz-González et al. 2017; Shafahi et al. 2018). In contrast, the attacker only has limited knowledge of training data but not the learning algorithms and model parameters in the black-box scenarios (Jagielski et al. 2018). In the proposed method, we generate the poisoned data from a generative model based on the statistical distribution of the training data and feature set, which does not require direct knowledge of the victim classifier.

Attacker's Capability. We consider the most widely-used setting as in (Shafahi et al. 2018; Huang et al. 2020; Zhu et al. 2019; Saha, Subramanya, and Pirsiavash 2020) that assumes the attacker has total control over the training data. The attacker can only provide training dataset (e.g., upload to web-based repositories) without directly modifying the existing training data. Meanwhile, we consider that the training data would be labeled by human experts upon inspection, i.e., the poisoned images have to be transparent with respect to manual annotation. Such a "clean-label" requirement is more stringent than prior works (Muñoz-González et al. 2017; Yang et al. 2017; Muñoz-González et al. 2019; Zhao and Lao 2022, 2018), which also potentially enables more poisoned data to be injected as the poisoned data are able to evade human inspection. The percentage of injected poisoned points in a poisoned training set is defined as the poisoning ratio. To maximize the poisoning effect, the attacker desires to inject as many poisoned data as possible into the training dataset.

Methodology

Theoretically, we can directly train a GAN to learn the distribution of "natural poisoned data" if there is enough training data. However, these data might not be sufficient for poisoning in practice. For instance, although exist, there are only 644 out of 10000 images are "natural poisoned data" in the CIFAR10 test dataset for the aforementioned Vgg16 classifier. Therefore, we propose to leverage generative models for generating the desired poisoned data.

The proposed CLPA framework consists of two phases along with a special triplet loss function, as illustrated in Figure 2. Our goal in phase I is to train a GAN to learn the distribution of clean training data and generate synthesized images with high quality. While in phase II, we fine-tune a well-trained GAN to learn the distribution of the "natural poisoned data" and generate the desired poisoned images. In particular, we use the triplet loss to effectively guide GAN to generate poisoned data that share similar characteristics as "natural poisoned data". There are three main advantages of such a flow: 1) By decoupling the training phases, we can minimize the interference of the phase II to phase I and generate high-quality poisoned data; 2) By leveraging the capability of GAN, we are able to generate unlimited poisoned data without knowing the knowledge of the victim classifier; 3) The phase II training requires much fewer iterations than phase I. We can generalize the approach by embedding it to any well-trained GAN, which saves time and improves efficiency for poisoned data generation. The completed poisoned data generation framework is presented in Algorithm 1.

In phase I, we follow the standard GAN training procedure and train a conditional GAN (Mirza and Osindero 2014) that generates fake images based on label information on the target dataset. The optimization can be expressed as:

$$\begin{aligned} \min_{\mathcal{G}} \max_{\mathcal{D}} \quad & \mathcal{V}(\mathcal{D},\mathcal{G}) = \mathcal{L}(\mathcal{D}) + \mathcal{L}(\mathcal{G}), \\ \textbf{s.t. } & \mathcal{L}(\mathcal{D}) = E_{\mathbf{x}\sim p_x(\mathbf{x})}[\log \mathcal{D}(\mathbf{x}|\mathbf{y})], \\ & \mathcal{L}(\mathcal{G}) = E_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1-\mathcal{D}(\mathcal{G}(\mathbf{z}|\mathbf{y})))], \end{aligned} \tag{2}$$

where the discriminator is trained to distinguish fake images from real images, while the generator is trained to learn the distribution of real images and generate fake images to fool the discriminator. $\mathbf{x}$ is the poisoned image generated by the generator $\mathcal{G}$ where the input of which is a random noise $\mathbf{z}$ sampled from prior distribution $p_z(\mathbf{z})$ and a corresponding label $\mathbf{y}$ that associated with the image. We adopt the BigGAN (Brock, Donahue, and Simonyan 2019) architecture in the proposed framework.

The objective of phase II is to guide the generator to learn the distribution of desired poisoned data. Similar to the concept in (Schroff, Kalenichenko, and Philbin 2015) that uses a triplet loss function to guide model training in facial recognition, our framework incorporates a modified triplet loss function. The original triplet loss is designed to recognize different faces by minimizing the distance between an anchor (image of a specific person) and a positive (other images of the same person) while maximizing the distance between the anchor and a negative (images of any other person). Similarly, for each image denoted as *anchor*, we call the ground-truth class of the image as the *positive* and the desired misclassified class as the *negative*. Intuitively, we want the generator to generate images that have features of both positive and negative so that they will locate in the overlapped area in the representation space. We achieve this by minimizing the distance difference between the anchor to negative and the anchor to positive. We formalize the problem as follows:

$$\begin{aligned} & \big|d_1 - d_2\big| < \alpha, \\ & \textbf{s.t. } d_1 = ||\mathcal{R}(\mathbf{x}) - R_{neg}||_2, \\ & \quad d_2 = ||\mathcal{R}(\mathbf{x}) - R_{pos}||_2, \end{aligned} \tag{3}$$

where $\mathcal{R}(\mathbf{x})$ presents the embedding that embeds an image $\mathbf{x}$ generated by $\mathcal{G}(\mathbf{z}|\mathbf{y})$ into a d-dimensional Euclidean space: $\mathbb{R}^n \rightarrow \mathbb{R}^d$. We denote the distance between the anchor

Algorithm 1: Poisoned Data Generation

Input: Training dataset $(\mathbf{x},\mathbf{y}) \sim \boldsymbol{D_{tr}}$; Embedding $\boldsymbol{\mathcal{R}}$ of neural network $\boldsymbol{\mathcal{C}}$ that is trained on $\boldsymbol{D_{tr}}$; Randomly initialized generative adversarial model $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{D}}$

Output: Poisoned dataset $\boldsymbol{X_p}$

1 // Phase I training
2 **for** number of training iterations **do**
3 **for** i steps **do**
4 Sample a mini-batch of noise samples $(z_1, z_2, \ldots, z_m) \sim p_z(\mathbf{z})$
5 Sample a mini-batch of training examples $(x_1, x_2, \ldots, x_m) \sim p_x(\mathbf{x})$
6 Train the discriminator by gradient ascent: $\nabla_{\theta_{\mathcal{D}}} \frac{1}{m} \sum_{k=1}^{m} [\log \mathcal{D}(x_k|\mathbf{y}_k) + \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}_k|\mathbf{y}_k)))]$
7 **end for**
8 Sample a mini-batch of noise samples $(z_1, z_2, \ldots, z_m) \sim p_z(\mathbf{z})$
9 Train the generator by gradient descent: $\nabla_{\theta_{\mathcal{G}}} \frac{1}{m} \sum_{k=1}^{m} \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}_k|\mathbf{y}_k)))$
10 **end for**
11 // Phase II training
12 **for** number of training iterations **do**
13 **for** i steps **do**
14 Same operation from line 10-12 to train discriminator
15 **end for**
16 Sample a mini-batch of noise samples $(z_1, z_2, \ldots, z_m) \sim p_z(\mathbf{z})$
17 Get R_{pos} and R_{neg} from Algorithm 2
18 Train the generator by gradient descent: $\nabla_{\theta_{\mathcal{G}}} \frac{1}{m} \sum_{k=1}^{m} \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}_k|\mathbf{y}_k)) + \mathcal{L}_{\mathcal{R}}(\mathcal{G}(\mathbf{z}_k|\mathbf{y}_k)))$
19 // Poisoned data generation
20 Initialize $\boldsymbol{X_p} \leftarrow \{\}$
21 Sample random noise samples $\mathbf{z} \sim p_z(\mathbf{z})$
22 $\boldsymbol{X_p} \leftarrow \mathcal{G}(\mathbf{z}|\mathbf{y})$
23 **Return** $\boldsymbol{X_p}$

Algorithm 2: Negative Class Selection

Input: Training dataset $(\mathbf{x}, \mathbf{y}) \sim \boldsymbol{D_{tr}}$, let D_Y represent the training sample corresponding to class $\mathbf{Y}$; Embedding $\boldsymbol{\mathcal{R}}$ of neural network $\boldsymbol{\mathcal{C}}$ that is trained on $\boldsymbol{D_{tr}}$

for each class $\mathbf{Y}$ **do**
  Initialize $list_{mis}$ = []
  Let $n = |D_Y|$, enumerate $(x_1, x_2 \dots x_n) \sim D_Y$
  for i steps **do**
    if $\mathcal{C}(x_i)$!= $\mathbf{Y}$ **then**
      $list_{mis}$.append($\mathcal{C}(x_i)$)
  Compute the numbers of misclassified images of each class: $nums$ = $\mathbf{Count}(list_{mis})$
  $\mathbf{Y_{pos}} = Y$
  $\mathbf{Y_{neg}} = argmax(nums)$
  $R_{pos} = \frac{1}{n}\mathcal{R}(D_{Y_{pos}})$
  $R_{neg} = \frac{1}{n}\mathcal{R}(D_{Y_{neg}})$
end for
Return R_{pos}, R_{neg}

and the negative centroid as d_1, and the distance between the anchor and the positive centroid as d_2, respectively. α is the margin that enforces the distance gap. Concretely, we eliminate the last fully connected layer of a high accuracy neural network $\mathcal{C}$ and use the rest part as $\mathcal{R}$, which outputs a multi-dimensional feature vector. R_{pos} and R_{neg} are the feature representations of positive centroid and negative centroid of the corresponding classes. We select negative class by following the strategy as described in Algorithm 2. We analyze the distribution of naturally misclassified samples of each class and pick the class with the most misclassified images as the negative class to the ground-truth class.

We can then achieve our goal by minimizing the following triplet loss function:

$$\mathcal{L}(\mathcal{G}, \mathcal{R}) = \frac{1}{N}\sum_{i}^{N}(|d_1 - d_2| - \alpha), \tag{4}$$

where N is the total number of poisoned images. The phase II training can be expressed as follows:

$$\min_{\mathcal{G},\mathcal{R}} \max_{\mathcal{D}} \quad \mathcal{V}(\mathcal{D}, \mathcal{G}) + \mathcal{L}(\mathcal{G}, \mathcal{R}). \tag{5}$$

Experiments

Experimental Settings

We evaluate on scenarios where poisoning attacks are of concern, i.e., downstream tasks are trained on pre-trained models using images of interest (Shafahi et al. 2018; Yang et al. 2017; Muñoz-González et al. 2017). Models pre-trained on ImageNet dataset are used to further build a classifier for the CIFAR-10 and ImageNet images classification tasks, respectively. Four victim models (Inception V3, ResNet50, Vgg16 and Vgg19) are trained on clean training dataset and poisoned dataset, respectively. The poisoned data are generated by the GAN and the embedding $\mathcal{R}$ is trained on the original clean dataset for both tasks. The parameters selection of embedding and GAN training are summarized in Table 2.

	CIFAR-10	ImageNet
Embedding $\mathcal{R}$	Vgg16	Vgg19
Original accuracy of $\mathcal{R}$	93.56%	72.38%
Embedding dimension	512	4096
Training iterations of phase I	60000	138000
Training iterations of phase II	200	2000
Margin parameter α	0.5	0.8

Table 2: Experiment settings for CIFAR-10 and ImageNet.

We consider two training strategies where we initialize four networks with pre-trained weights on the ImageNet dataset (Deng et al. 2009) and further train downstream tasks on image classification on target dataset by using **end-to-end training** or training the **FC layer only**.

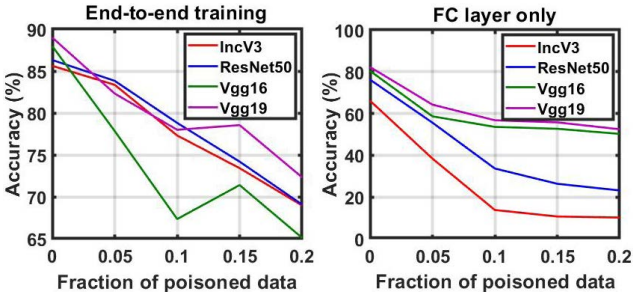

Figure 3: Availability attack on CIFAR-10.

Evaluation on CIFAR-10

For the CIFAR-10 classification task, we train the downstream classifiers using 5000 training images from the training dataset combined with poisoned data. The neural networks are trained 10 epochs for end-to-end training and FC layer only. We randomly select two classes (i.e., dog and frog) for poisoned data injection and vary the poisoning ratio from 0 to 0.2 with an interval of 0.05 in both settings. SGD optimizer is used for model training at a learning rate of 1×10^{-4} with a batch size of 16. The performance is evaluated on the test dataset of 10000 images. As shown in Figure 3, the CLPA attack effectively deteriorates model accuracy for all victim neural networks in both training scenarios, which indicates strong attack capability and decent transferability of the poisoned data. The attack on Vgg16 model achieves the best performance since we use Vgg16 as the embedding for negative class selection and phase II GAN training. Moreover, training only the FC layer renders more accuracy degradation than training the entire network.

As discussed above, the stealthiness of the clean-label attack brings more opportunities for injecting poisoned data without being detected. Therefore, it is of interest to examine the full potential of the poisoned data by raising the poisoning ratio to 1. We show the results in Figure 4, where the attack successfully subverts all victim models, resulting in a significant 55.22% accuracy drop on average.

Poisoning Ratio	End-to-end training										
	Metapoison Acc. (%)					CLPA (ours) Acc. (%)					PERF. ↑
	Incv3	ResNet50	Vgg16	Vgg19	Acc. ↓	Incv3	ResNet50	Vgg16	Vgg19	Acc. ↓	(CLPA- MP)
0	85.62 (±0.6)	86.32 (±0.7)	87.97 (±0.1)	88.99 (±0.2)	-	85.62 (±0.6)	86.32 (±0.7)	87.97 (±0.1)	88.99 (±0.2)	-	-
0.05	83.45 (±0.8)	85.38 ±(0.4)	78.29 ±(3.6)	84.81 (±2.2)	4.25 ↓	83.35 (±0.3)	83.83 (±0.7)	77.84 (±2.7)	82.31 (±3.7)	5.40 ↓	1.15 ↑
0.10	82.83 (±0.6)	82.89 (±0.6)	71.67 (±5.2)	79.25 (±1.7)	8.22 ↓	77.31 (±0.6)	78.81 (±0.6)	67.34 (±4.5)	77.99 (±2.6)	11.87 ↓	3.65 ↑
0.15	80.50 (±1.1)	79.98 (±0.6)	67.36 (±3.7)	74.70 (±4.6)	11.60 ↓	73.41 (±0.8)	74.19 (±0.7)	71.39 (±1.5)	78.55 (±0.9)	12.85 ↓	1.25 ↑
0.20	77.62 (±1.1)	76.66 (±0.6)	70.55 (±3.6)	77.32 (±2.2)	11.69 ↓	68.96 (±1.1)	69.07 (±0.3)	65.10 (±5.5)	72.32 (±4.3)	18.37 ↓	6.68 ↑

Table 3: Comparison of CLPA and Metapoison on availability attacks.

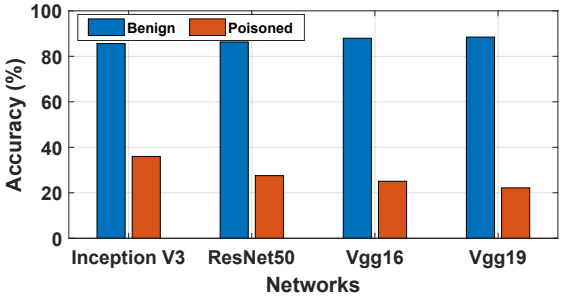

Figure 4: Test accuracy when training with full poisoned data on CIFAR-10 (end-to-end training).

Comparison to Metapoison

Although clean-label poisoning availability attacks are not well studied before, there has been a line of works focusing on clean-label integrity attacks (Huang et al. 2020; Geiping et al. 2021; Shafahi et al. 2018; Zhu et al. 2019). We adapt the state-of-the-art integrity attack, Metapoison (Huang et al. 2020), for degrading the availability as a baseline method for comparison. We download the poisoned data generated by Metapoison[1] and keep all other experimental settings unchanged. The results are presented in Table 3. It can be seen that it is possible to adapt the integrity attack against the availability and undermine the overall model accuracy. However, the effectiveness of such attacks is limited. For instance, Metapoison achieves the best performance with only a 11.67% accuracy drop in attacking the Vgg19 network. Meanwhile, the CLPA attack outperforms the Metapoison attack at almost all levels of poisoning ratio for all models.

Evaluation on Poisoned Data Quality

We evaluate the poisoned image quality from two perspectives, i.e., human inspection and the Fréchet inception distance (FID) score (Heusel et al. 2017). FID score is a more widely-used metric that captures the similarity of synthetic images to real ones, compared to Inception Score (IS) (Salimans et al. 2016). Lower FID scores indicate that two groups of images have more similar statistics or a better image quality compared to real images. We first showcase the visual results of the poisoned data. We take "automobile" as an example for demonstration. Recall that we refer to the synthetic fake data from phase I as synthesized images and the synthetic poisoned data from phase II as poisoned images. Comparison of clean images, synthesized images, and poisoned images is shown in Figure 5. It can be seen that both synthesized and poisoned images have decent quality and are hard to distinguish by human perception.

Figure 5: Comparison of clean images, GAN synthesized images, and poisoned images.

We then compute the FID score of images from phase I and phase II. We follow the original setting that uses the Inception-V3 model to capture the feature of images and compare our results to the baseline models reported in (Shmelkov, Schmid, and Alahari 2018). As shown in Table 4, both synthesized and poisoned images show comparable performance to DCGAN (Radford, Metz, and Chintala 2016) and achieve a much lower score compared to PixelCNN++ (Salimans et al. 2017), indicating a high similarity to real images. Note that the poisoned data only has a slightly higher FID score than the synthesized images, which further validates the advantages of the two-phase design, as it enables the poisoned data to inherit decent image quality from a well-trained GAN.

Performance against Detection and Sanitization

As shown above, the generated poisoned data are close to real images and have decent FID scores. Thus, inspection in the

[1]https://github.com/wronnyhuang/metapoison

Model	FID Score
Phase I (synthesized)	**41.8**
Phase II (poisoned)	**47.1**
DCGAN (baseline 1)	36.5
PixelCNN++ (baseline 2)	119.5

Table 4: FID of synthesized and poisoned data (CIFAR-10).

pixel space cannot effectively remove the poisoned data. We further evaluate the performance of the poisoned data against the data sanitization technique introduced in (Tran, Li, and Madry 2018), which aims at removing outliers based on the representation space for defending poisoning attacks. The approach leverages singular vector decomposition to detect outliers. We use the same default threshold as in (Tran, Li, and Madry 2018) where the removal budget is $1.5\times$ of total poisoned data. For instance, if 100 poisoned data are injected, 150 data points that have the highest outlier score will be removed from the training dataset. Note that the algorithm assumes the defender knows which object class is poisoned. We take the class "airplane" as an example and vary the poisoning ratio from 0.05 to 0.40. The results are presented in Table 5. It can be seen that even with such a strong removal budget, the poisoned data can still mostly bypass the defense. Since the triplet loss helps to guide the generator to learn features from both positive and negative classes, the generated images do not reveal a distinctive spectral signal in the representation space.

Poisoned Image	Remove Budget	Clean Data Removed	Poisoned Removed
50	75	66 (88%)	9 (12%)
100	150	119 (79%)	31 (21%)
150	225	170 (76%)	55 (24%)
200	300	222 (74%)	78 (26%)
250	375	270 (72%)	105 (28%)
300	450	315 (70%)	135 (30%)
350	525	360 (69%)	165 (31%)
400	600	402 (67%)	198 (33%)

Table 5: Performance against spectral signal detection with 1000 clean images.

Evaluation on ImageNet

The proposed approach shows superior performance on the CIFAR-10 dataset. However, it is way more challenging to scale up poisoning attacks to large-scale datasets such as ImageNet. To comprehensively understand the effectiveness of our attack, we then evaluate the method on the ImageNet dataset. Training GAN for ImageNet is extremely hard and time-consuming. However, by using the proposed method, we can simply skip the phase I training and fine-tune a pre-trained GAN model for poisoned data generation, which only takes about 2 hours in our experiments.

The FID scores of the phase I and phase II GAN model are presented in Table 6. The corresponding images of clean data, phase I and II are shown in Figure 6. Similar to the results of CIFAR-10, the synthetic images of ImageNet are of good quality, and the poisoned images successfully inherit the image quality and are plausible upon human inspection.

Model	FID Score
Phase I (synthesized)	17.79
Phase II (poisoned)	37.94

Table 6: FID of synthesized and poisoned data (ImageNet).

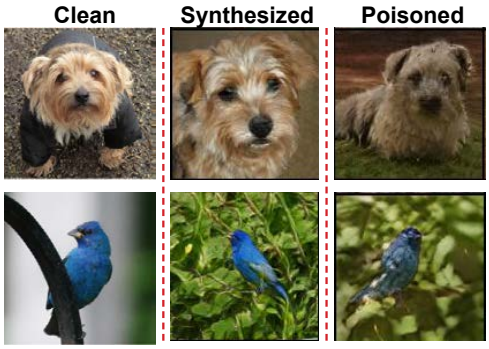

Figure 6: Comparison of clean images, GAN synthesized images, and poisoned images (ImageNet).

In Figure 7, we present the results of the attack effect. Given the high image quality, we examine the full capability of CLPA on ImageNet. Our attack significantly corrupts all the victim models and achieves an average 44.68% accuracy drop, indicating that the proposed poisoning attack generalizes well on large-scale dataset.

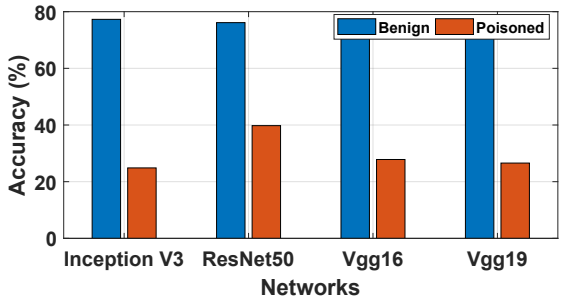

Figure 7: Test accuracy when training with full poisoned data on ImageNet (end-to-end training).

Conclusion

In this work, we propose a novel framework, **CLPA**, for clean-label poisoning availability attacks. Inspired by the "natural poisoned data", we exploit the generative adversarial net to synthesize effective poisoned samples efficiently. We propose a two-phase GAN training methodology with the triplet loss function to guarantee the quality of poisoned images and achieve a higher accuracy drop. The performance of the proposed method is evaluated comprehensively from multiple aspects over different datasets.

Acknowledgments

This work is partially supported by the National Science Foundation award 2047384.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *the 3rd International Conference on Learning Representations, ICLR 2015*.

Biggio, B.; Nelson, B.; and Laskov, P. 2012. Poisoning Attacks against Support Vector Machines. In *the 29th International Conference on Machine Learning, ICML 2012*.

Biggio, B.; Pillai, I.; Bulò, S. R.; Ariu, D.; Pelillo, M.; and Roli, F. 2013. Is data clustering in adversarial settings secure? In *the 2013 ACM Workshop on Artificial Intelligence and Security*, 87–98.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *7th International Conference on Learning Representations, ICLR 2019*.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *CoRR*, abs/1712.05526.

Clements, J.; and Lao, Y. 2018a. Backdoor Attacks on Neural Network Operations. In *2018 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2018*, 1154–1158.

Clements, J.; and Lao, Y. 2018b. Hardware Trojan Attacks on Neural Networks. *CoRR*, abs/1806.05768.

Clements, J.; and Lao, Y. 2019. Hardware Trojan Design on Neural Networks. In *IEEE International Symposium on Circuits and Systems, ISCAS 2019*, 1–5.

Clements, J.; and Lao, Y. 2022a. DeepHardMark: Toward Watermarking Neural Network Hardware. In *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*.

Clements, J.; and Lao, Y. 2022b. In Pursuit of Preserving the Fidelity of Adversarial Images. In *International Conference on Acoustics, Speech and Signal Processing*.

Clements, J.; Yang, Y.; Sharma, A. A.; Hu, H.; and Lao, Y. 2021. Rallying adversarial techniques against deep learning for network security. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 01–08. IEEE.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. IEEE.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, 4171–4186.

Doan, K.; Lao, Y.; and Li, P. 2021. Backdoor Attack with Imperceptible Input and Latent Modification. In *Advances in Neural Information Processing Systems, NeurIPS 2021*.

Doan, K.; Lao, Y.; Zhao, W.; and Li, P. 2021. LIRA: Learnable, Imperceptible and Robust Backdoor Attacks. In *IEEE/CVF International Conference on Computer Vision ICCV 2021*, 11966–11976.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Feng, J.; Cai, Q.; and Zhou, Z. 2019. Learning to Confuse: Generating Training Time Adversarial Data with Auto-Encoder. In *Advances in Neural Information Processing Systems, NeurIPS 2019*, 11971–11981.

Geiping, J.; Fowl, L. H.; Huang, W. R.; Czaja, W.; Taylor, G.; Moeller, M.; and Goldstein, T. 2021. Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching. In *9th International Conference on Learning Representations, ICLR 2021*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems, NeurIPS 2014*, 2672–2680.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *the 3rd International Conference on Learning Representations, ICLR 2015*.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, 770–778.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems, NeurIPS 2017*, 6626–6637.

Huang, W. R.; Geiping, J.; Fowl, L.; Taylor, G.; and Goldstein, T. 2020. MetaPoison: Practical General-purpose Clean-label Data Poisoning. In *Advances in Neural Information Processing Systems, NeurIPS 2020*.

Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; and Li, B. 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings*, 19–35.

Koh, P. W.; and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. In *the 34th International Conference on Machine Learning, ICML 2017*, volume 70, 1885–1894.

Koh, P. W.; Steinhardt, J.; and Liang, P. 2018. Stronger Data Poisoning Attacks Break Data Sanitization Defenses. *CoRR*, abs/1811.00741.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems, NeurIPS 2012*, 1097–1105.

Lao, Y.; Zhao, W.; Yang, P.; and Li, P. 2022. DeepAuth: A DNN Authentication Framework by Model-Unique and Fragile Signature Embedding. In *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*.

Mei, S.; and Zhu, X. 2015. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners. In *the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015*, 2871–2877.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

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 *the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017*, 27–38.

Muñoz-González, L.; Pfitzner, B.; Russo, M.; Carnerero-Cano, J.; and Lupu, E. C. 2019. Poisoning Attacks with Generative Adversarial Nets. *CoRR*, abs/1906.07773.

Nguyen, T. A.; and Tran, A. T. 2021. WaNet - Imperceptible Warping-based Backdoor Attack. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Paudice, A.; Muñoz-González, L.; and Lupu, E. C. 2018. Label Sanitization against Label Flipping Poisoning Attacks. *CoRR*, abs/1803.00992.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations, ICLR 2016*.

Saha, A.; Subramanya, A.; and Pirsiavash, H. 2020. Hidden Trigger Backdoor Attacks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 11957–11965.

Salimans, T.; Goodfellow, I. J.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems, NeurIPS 2016*, 2226–2234.

Salimans, T.; Karpathy, A.; Chen, X.; and Kingma, D. P. 2017. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. In *5th International Conference on Learning Representations, ICLR 2017*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, 815–823.

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. In *Advances in Neural Information Processing Systems, NeurIPS 2018*, 6106–6116.

Shmelkov, K.; Schmid, C.; and Alahari, K. 2018. How Good Is My GAN? In *Computer Vision - ECCV 2018 - 15th European Conference*, volume 11206, 218–234.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T. P.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; and Hassabis, D. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T. P.; Simonyan, K.; and Hassabis, D. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *CoRR*, abs/1712.01815.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *the 3rd International Conference on Learning Representations, ICLR 2015*.

Tran, B.; Li, J.; and Madry, A. 2018. Spectral Signatures in Backdoor Attacks. In *Advances in Neural Information Processing Systems, NeurIPS 2018*, 8011–8021.

Turner, A.; Tsipras, D.; and Madry, A. 2019. Label-Consistent Backdoor Attacks. *CoRR*, abs/1912.02771.

Xiao, H.; Biggio, B.; Brown, G.; Fumera, G.; Eckert, C.; and Roli, F. 2015. Is Feature Selection Secure against Training Data Poisoning? In *the 32nd International Conference on Machine Learning, ICML 2015*, volume 37, 1689–1698.

Xiao, H.; Xiao, H.; and Eckert, C. 2012. Adversarial Label Flips Attack on Support Vector Machines. In *ECAI 2012 - 20th European Conference on Artificial Intelligence*, volume 242, 870–875.

Yang, C.; Wu, Q.; Li, H.; and Chen, Y. 2017. Generative Poisoning Attack Method Against Neural Networks. *CoRR*, abs/1703.01340.

Zhao, B.; and Lao, Y. 2018. Resilience of Pruned Neural Network Against Poisoning Attack. In *13th International Conference on Malicious and Unwanted Software, MALWARE 2018*, 78–83.

Zhao, B.; and Lao, Y. 2022. Towards Class-Oriented Poisoning Attacks against Neural Networks. *2022 IEEE Winter Conference on Applications of Computer Vision, WACV 2022*.

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 *the 36th International Conference on Machine Learning, ICML 2019*, volume 97, 7614–7623.