COMBAT: Alternated Training for Effective Clean-Label Backdoor Attacks

Tran Huynh[1], **Dang Nguyen**[1, 2], **Tung Pham**[1], **Anh Tran**[1]

[1]VinAI Research
[2]University of Maryland
v.tranhn2@vinai.io, dangmn@umd.edu, v.tungph4@vinai.io, v.anhtt152@vinai.io

Abstract

Backdoor attacks pose a critical concern to the practice of using third-party data for AI development. The data can be poisoned to make a trained model misbehave when a predefined trigger pattern appears, granting the attackers illegal benefits. While most proposed backdoor attacks are dirty-label, clean-label attacks are more desirable by keeping data labels unchanged to dodge human inspection. However, designing a working clean-label attack is a challenging task, and existing clean-label attacks show underwhelming performance. In this paper, we propose a novel mechanism to develop clean-label attacks with outstanding attack performance. The key component is a trigger pattern generator, which is trained together with a surrogate model in an alternating manner. Our proposed mechanism is flexible and customizable, allowing different backdoor trigger types and behaviors for either single or multiple target labels. Our backdoor attacks can reach near-perfect attack success rates and bypass all state-of-the-art backdoor defenses, as illustrated via comprehensive experiments on standard benchmark datasets. Our code is available at https://github.com/VinAIResearch/COMBAT.

1 Introduction

To deal with numerous real-life situations, AI models often need massive training data, which is hard to collect. Thus the data often comes from various sources like third parties or open sources. However, recent studies have shown that the data outsourcing practice can open a loophole for backdoor attacks. An attacker can provide training data that is partially poisoned with a pre-defined trigger pattern. A model trained on such data exhibits two properties. First, it performs well on normal, “clean” images like a genuine model. However, when the trigger pattern is embedded in the input, the model will give an erroneous prediction as designed by the attacker. This allows the attacker to gain malicious access or cause damage to the user’s side. For example, attackers can disguise themselves as privileged users by breaking a face-recognition-based security system, or they can fool self-driving cars into misreading the traffic signs and causing accidents. Hence, understanding the capability of this security threat is critical, drawing many research interests in recent years. This paper focuses on backdoor attacks on image classification, the most studied task, but the discovery should be easily extended to other domains.

Data-poisoning-based backdoor attacks are often classified as “dirty-label” or “clean-label”. In dirty-label attacks, the adversary poisons some data and changes its labels to the attack’s target class. It could be easily spotted by humans, e.g., a poisoned dog image could be labeled as a “cat”. In contrast, in clean-label attacks, the attacker only poisons data without changing its labels, making this attack mechanism more stealthy and desirable.

However, one critical drawback of most existing clean-label attacks is their low efficiency. For dirty-label ones, the poisoned examples have a fixed label regardless of the image content, forcing the classifier to associate the backdoor trigger with the attack’s target class, leading to an almost 100% attack success rate (ASR). Meanwhile, in clean-label attacks, the classifier may just learn the image content and ignore the trigger since all labels are correct. For example, a naive adaptation to clean-label style for BadNets (Gu, Dolan-Gavitt, and Garg 2017), a common dirty-label method, completely fails. In addition, most existing clean-label attacks (Turner, Tsipras, and Madry 2019; Barni, Kallas, and Tondi 2019) cannot reach an 80% ASR. Although some recent works (Ning et al. 2021; Zeng et al. 2022) achieves near-perfect ASRs, they require significant modifications to the training data. Due to the difficulty in designing a working and effective algorithm, only a few clean-label attacks have been proposed, and they have not been well studied in backdoor defense research.

In our perspective, defining the optimal backdoor trigger based on measuring its effect on the model poisoning result is a solution for developing successful clean-label attacks. Therefore, we propose a novel clean-label attack mechanism called Clean-label OptiMize Backdoor Alternated Training, or **COMBAT** for short. It aims to learn a generator that can generate an effective, input-dependent backdoor trigger. As indicated in its name, COMBAT employs an alternated training process to alternately optimize the generator and a surrogate model, aiming to maximize the generator’s poisoning effectiveness. In the surrogate model training step, COMBAT mimics the real data poisoning and backdoor modeling process. In the generator training step, it updates the generator to maximize the attack success rate on the surrogate model. More loss functions can be freely added in this step to define other desired properties of the attack, such as imperceptibility and defense nullification. After training, we obtain an optimized generator with known and transferable poisoning effectiveness. We evaluate our method on various datasets, including CIFAR-10, ImageNet-10, and CelebA. Impressive results are captured in the experiments, showing that our attack is highly effective accross different datasets and models while using exceedingly small triggers. The attack is also stealthy, breaking all backdoor defenses.

Besides its effectiveness, COMBAT is also flexible, allowing various customizations. We demonstrate this advantage by designing different variants, including input-aware, warping-based, and multi-target attacks. COMBAT is sufficient to train these extremely-different methods to all reach high success rates. We believe it will define a general training procedure for future clean-label backdoor attacks, stimulating the development of this critical security research.

2 Background

2.1 Threat Model

In backdoor attacks, the attacker can provide a poisoned dataset (dataset-poisoning) or a poisoned network (model-poisoning). We focus on the dataset-poisoning scheme.

In this attack scenario, the attacker acts as a data provider that supplies a victim with a dataset for image classification training via a commercial transaction or an open-source release. He or she secretly poisons the data before releasing it, using a backdoor injection function with a pre-defined trigger pattern and a target attack label. The trigger pattern can be in any form, such as noise, image patch, blended content, or pixel shifts. The victim will train a classifier on the poisoned dataset and then obtains a backdoored model that disguises itself as a rightful model by returning correct prediction from clean input and producing the target class from any poisoned datum. The victim does not recognize this behavior and deploys it in his or her system, allowing the attacker to gain illegal benefits.

Data poisoning techniques can be divided into two groups: *dirty-label* and *clean-label*. In this work, we focus on the clean-label attacks, in which the attacker poisons only some images and keeps their labels unchanged. For efficiency, normally only a portion of the *target-class* images are injected with the backdoor.

2.2 Previous Backdoor Attacks

The earliest backdoor attack is BadNets (Gu, Dolan-Gavitt, and Garg 2017), which uses a fixed image patch as a trigger embedded into a small portion of data and changes their labels to the target class. Despite its simple scheme, BadNets highly succeeded on various datasets. After BadNets, many methods have been proposed in which some (Liu et al. 2018; Yao et al. 2019; Rakin, He, and Fan 2020; Chen et al. 2021; Bober-Irizar et al. 2022) define novel ways to inject backdoor, and others develop stealthy and effective backdoor injection functions. In this study, we only consider the latter.

The majority of proposed backdoor attacks are dirty-label, and we can only name a few here. (Nguyen and Tran 2020) employed input-dependent trigger patterns to dodge the common backdoor defenses that relied on the fixed-trigger assumption. (Nguyen and Tran 2021) designed a novel, imperceptible backdoor trigger based on image warping. (Doan et al. 2021) optimized the backdoor trigger function during the training process towards imperceptible trigger in the input space, while later works (Doan, Lao, and Li 2021; Zhong, Qian, and Zhang 2022) further made backdoors imperceptible in the latent space. Recent approaches (Wang et al. 2021; Hammoud and Ghanem 2021) exploited the frequency domain for stealthy attacks.

As mentioned, dirty-label attacks are not realistic in the dataset-providing scenario due to the easy-to-detect inconsistency between image contents and labels. (Turner, Tsipras, and Madry 2019) first time discussed this issue and proposed the clean-label attack scheme. The paper then proposed to perturb each poisoning example to make its latent depart from the original class before adding a fixed trigger patch. (Barni, Kallas, and Tondi 2019) later proposed to use fixed sinusoidal strips as the trigger pattern. Refool (Liu et al. 2020) designed a natural-looking attack in which the embedded trigger pattern is disguised as image reflection. (Saha, Subramanya, and Pirsiavash 2020) introduced a hidden backdoor attack via model fine-tuning that first generated a patch-based poisoned sample, then embedded it into texture of a training image of the target class by minimizing their distance in the feature space, thus making the trigger invisible. (Souri et al. 2021) allowed hidden attacks on training-from-scratch models by applying gradient matching. Recently, Narcissus (Zeng et al. 2022) employed a clean surrogate model and optimized an uniform trigger; that approach is quite similar to adversarial attack. Still, all these methods had underwhelming attack performance compared to dirty-label counterparts.

2.3 Backdoor Defense Methods

To protect victims from backdoor attacks, detecting and mitigating potential attack methods have been applied in any stage ranging from dataset scanning (***data defense***), model examination (***model defense***), to test-time monitoring when the model is already deployed (***test-time defense***). Below is a brief summary of those defense methods.

Data defense. This defense aims at purifying the training dataset by detecting and removing poisoned samples, preventing backdoor formation from the source. (Tran, Li, and Madry 2018) filtered backdoor samples assuming a discernible trace in the spectrum of the covariance feature representations. (Chen et al. 2018) relied on latent representation clustering, assuming clean and poisoned samples had distinct characteristics in the feature space. (Zeng et al. 2021b) filter data based on the frequently observed high-frequency artifacts in backdoored samples.

Model defense. Model defenses identify or mitigate poisoned models by inspecting their behaviors when dealing with clean data. Fine-pruning (Liu, Dolan-Gavitt, and Garg 2018) suggested pruning inactive neurons, but it could not verify backdoor presence. Neural Cleanse (Wang et al. 2019) tested if a model was poisoned by first computing optimal class-inducing patterns for each label, then detecting abnormally small patterns. ABS (Liu et al. 2019) scanned the neurons to generate backdoor trigger candidates via reverse engineering technique, then verified these candidates on a clean image set. (Xu et al. 2020) utilized GradCAM (Selvaraju et al. 2017) to analyze the model's behaviors on images with and without the presence of engineering-reversed triggers. (Zhao et al. 2020) repaired the model's backdoor using the mode connectivity (Garipov et al. 2018) technique. (Kolouri et al. 2020) jointly optimized some universal litmus patterns (ULPs) and a meta-classifier to diagnose suspicious models. Li (Li et al. 2021) assumed knowledge distillation could perturb backdoor-related neurons. More recently, (Zeng et al. 2021a) proposed a minimax formulation for retraining the suspicious model to remove backdoors.

Test-time defense. Defense methods utilized at test time aim to filter out malicious samples. STRIP (Gao et al. 2019) exploited the stagnancy of the network prediction on poisoned data under various perturbations to detect poisoned samples. Neo (Udeshi et al. 2022) instead located the trigger region by searching for the minimal square-like block that altered the network prediction. Later, Februus (Doan, Abbasnejad, and Ranasinghe 2020) utilized GradCAM to identify abnormally small influential regions as potential triggers. In both, the trigger candidates were then verified by pasting them to a set of clean images.

3 Methodology

3.1 Problem Overview

In this section, we recall the formulation of clean-label backdoor attack problem.

Let $f_\theta : \mathcal{X} \to \mathcal{C}$ be the classification function mapping from the data space $\mathcal{X}$ to the set of classes $\mathcal{C}$, where θ is the classifier's hyper-parameters. Assume that we are given a training data set $\mathcal{S} = \big\{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{C}, i = 1, 2, \dots, n\big\}$ and $\mathcal{C} = \big\{0, 1, \dots, m\big\}$, then $\mathcal{S} = \bigcup_{j=0}^{m} \mathcal{S}^j$, where $\mathcal{S}^j$ denotes the subset of data for class j.

We consider a clean-label backdoor attack on a target class $\mathfrak{c} \in \mathcal{C}$. It first samples from clean data of the target class $\mathfrak{c}$ a subset for poisoning $\mathcal{P}^{\mathfrak{c}} \subseteq \mathcal{S}^{\mathfrak{c}}$, given a poisoning rate $p = {|\mathcal{P}^{\mathfrak{c}}|}/{|\mathcal{S}|}$. Then, it applies a transformation $\mathcal{T}$, which is a compositional function of a backdoor injection function $\mathcal{G}$ and some pre- and post-processing steps, to poison data in $\mathcal{P}^{\mathfrak{c}}$ to form a poisoned subset $\mathcal{P}_b^{\mathfrak{c}}$. For example, in (Turner, Tsipras, and Madry 2019), $\mathcal{T}$ consists of a pre-processing step (GAN interpolation/adversarial perturbation) and a patch-based backdoor trigger function. In this work, we consider the simple case when $\mathcal{T}$ is exactly $\mathcal{G}$. The rest of the training data, denoted by $\mathcal{S}' := \mathcal{S} \setminus \mathcal{P}^{\mathfrak{c}}$, is kept unchanged. The combined set $\mathcal{S}_b := \mathcal{S}' \cup \mathcal{P}_b^{\mathfrak{c}}$ is delivered to the victim to train a poisoned classifier of a hyper-parameter θ_b. This process can be expressed by formal equations:

$$\mathcal{P}_b^{\mathfrak{c}} = \{(\mathcal{G}(x_i), y_i) | (x_i, y_i) \in \mathcal{P}^{\mathfrak{c}}\}, \tag{1}$$

$$\mathcal{S}_b = \mathcal{P}_b^{\mathfrak{c}} \cup (\mathcal{S} \setminus \mathcal{P}^{\mathfrak{c}}), \tag{2}$$

$$\theta_b = \arg\min_{\theta} \sum_{(x,y) \in \mathcal{S}_b} \mathcal{L}\big(f_\theta(x), y\big), \tag{3}$$

where $\mathcal{L}$ is a loss function, e.g., cross-entropy. The desired poisoned classifier can correctly classify clean data input. However, when applying the backdoor trigger onto the input, this classifier always returns the target label $\mathfrak{c}$:

$$f_{\theta_b}(x) = c(x), \quad f_{\theta_b}(\mathcal{G}(x)) = \mathfrak{c} \quad \forall x \in \mathcal{X}, \tag{4}$$

with $c(\cdot)$ is the truth function returning the true input class.

In this study, we focus on designing an efficient backdoor function $\mathcal{G}$ so that any backdoor model trained using $\mathcal{G}$ (Eq. 1, 2, 3) can highly meet the conditions in Eq. 4.

3.2 Trigger Generator

In this section, we present the process of designing an effective backdoor function $\mathcal{G}$. We start with a simple design where $\mathcal{G}$ is a noise-additive function parameterized by ϕ:

$$\mathcal{G}_\phi(x) = x + \eta g_\phi(x), \tag{5}$$

with g_ϕ is a neural network that generates a trigger noise in the range of $[-1, 1]$ conditioned on the input image x and η is the ℓ_∞ bound of the added noise. Many existing backdoor attacks can be formulated as Eq. 5. However, recent research (Zeng et al. 2021b) highlights two problems that render such attacks easily detectable using a deep neural network-based detector. These issues include (1) inherent high-frequency artifacts of the trigger and (2) the decreased correlation of neighboring pixels when adding the trigger to the input image. In this work, we introduce two techniques to mitigate these problems. First, we constrain the generated noise to contain only low-frequency components.

To achieve this, given a noise $g_\phi(x) \in \mathbb{R}^{d \times d}$ [1], we remove its high-frequency artifacts by applying a filtering mask m to its type-II 2D Discrete Cosine Transform (DCT) (Ahmed, Natarajan, and Rao 1974) DCT$(g_\phi(x))$. Specifically, we consider $m \in \mathbb{R}^{d \times d}$ such that:

$$m_{i,j} = \begin{cases} 1 & \text{if } 1 \le i, j \le rd \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

and take Hadamard product of m and DCT$(g_\phi(x))$ to preserve only $rd \times rd$ top-left entries of DCT$(g_\phi(x))$ with some ratio $r \in (0, 1)$. Then, we reconstruct the trigger noise by applying the inverse DCT (IDCT) to the masked DCT$(g_\phi(x))$. The whole transformation is represented as:

$$\mathcal{Q}(g_\phi(x)) = \text{IDCT}(m \odot \text{DCT}(g_\phi(x))). \tag{7}$$

Although the generated noise carries only low-frequency components, adding it directly to an image can break the correlations between neighboring pixels of the original image. To mitigate this problem, we further apply a Gaussian blur filter k to the poisoned image. We obtain the final backdoor function as:

$$\mathcal{G}_\phi(x) = (x + \eta \mathcal{Q}(g_\phi(x))) * k. \tag{8}$$

For stronger stealthiness, we also want the trigger to be sufficiently small. Therefore, given a classifier f_θ, we want to minimize the loss of assigning poisoned data to the target

[1] Here, we assume 2D input for simplicity. For RGB images, the following process is performed on each channel.

class $\mathfrak{c}$ as well as the magnitude of the trigger. In the formula, these loss terms are defined as follows:

$$\mathcal{L}_a(f_\theta, \mathcal{G}_\phi; \mathcal{S}, \eta) := \sum_{(x_j, y_j)\in\mathcal{S}} \mathcal{L}(f_\theta(\mathcal{G}_\phi(x_j)), \mathfrak{c}) \tag{9}$$

$$\mathcal{L}_{\ell_2}(g_\phi; \mathcal{S}, \eta) := \sum_{(x_j, y_j)\in\mathcal{S}} \|\eta\mathcal{Q}(g_\phi(x_j))\|_2. \tag{10}$$

We observe that when training with only these loss functions, g_ϕ tends to generate adversarial noises. These perturbations cause the model to misclassify even without data poisoning in training, thus countering the purpose of backdoor attacks. We prevent g_ϕ from learning adversarial noises by employing a pretrained clean classifier of the same task as f_θ, denoted as h_ψ where ψ are network parameters. For each data sample (x_j, y_j), h_ψ should still correctly classify the input even when adding the trigger generated by g_ϕ:

$$\mathcal{L}_{\mathsf{d}}(\mathcal{G}_\phi, h_\psi; \mathcal{S}, \eta) := \sum_{(x_j, y_j)\in\mathcal{S}} \mathcal{L}\big(h_\psi(\mathcal{G}_\phi(x_j)), y_j\big) \tag{11}$$

While we use a single, specific clean network h_ψ in this loss function, g_ϕ can avoid producing adversarial perturbations of any similar clean classifier, thanks to the adversarial transferability property of deep networks. We will empirically confirm this in Section 4.4.

In essence, we find the optimal trigger function by solving the following optimization problem:

$$\begin{aligned}\phi^* = \arg\min_\phi \sum_{(x_j, y_j)\in\mathcal{S}} \big[&\mathcal{L}_a(f_\theta, \mathcal{G}_\phi; \mathcal{S}, \eta) + \lambda_{\ell_2}\mathcal{L}_{\ell_2}(g_\phi; \mathcal{S}, \eta)\\ &+ \lambda_{\mathsf{d}}\mathcal{L}_{\mathsf{d}}(\mathcal{G}_\phi, h_\psi; \mathcal{S}, \eta)\big],\end{aligned} \tag{12}$$

with λ_{ℓ_2} and λ_{d} are weighting hyper-parameters.

3.3 Alternated Training

As discussed, given a classifier, we could find the best trigger generator for that classifier. Since we want to poison the victim classifier f_{θ_b}, ideally, we wish to have that classifier in training $\mathcal{G}_\phi$. However, we need $\mathcal{G}_\phi$ first to train that victim classifier. This is a chicken-and-egg problem. A solution is to train a surrogate classifier f_θ that is as close to f_{θ_b} as possible. In particular, besides optimizing $\mathcal{G}_\phi$ with Eq. 12, we concurrently optimize f_θ with another loss function:

$$\theta^* = \arg\min_\theta \sum_{(x_j, y_j)\in\mathcal{S}} \mathcal{L}(f_\theta(x_j), y_j) + \sum_{(x_j, \mathfrak{c})\in\mathcal{P}_b^{\mathfrak{c}}} \mathcal{L}(f_\theta(\mathcal{G}_\phi(x_j)), \mathfrak{c}), \tag{13}$$

which mimics the process of training the victim classifier. We note that solving both Eq. 12 and Eq. 13 is like finding a balance between the trigger's magnitude and the class's boundary, and we implement it via alternated training. We also argue that the joint optimization allows smaller trigger sizes, consequently makes the poisoned data less perceptible. The results in Section 4.5 confirms that when the trigger is small, the ASRs obtained from alternated training outperform that from without alternated training.

After the alternated training process, the attacker acquires the optimal trigger generator $\mathcal{G}_{\phi^*}$, then uses it to generate the poisoned dataset $\mathcal{S}_b$. The whole process is described in Algorithm 1. The poisoned data will expectedly be used by the victim to train a classifier. This victim model will be poisoned and behave similarly to the surrogate model, as empirically proved in Section 4.2.

4 Experiments

4.1 Experimental Setup

We use three popular datasets, namely CIFAR-10 (Krizhevsky, Hinton et al. 2009), ImageNet-10, and CelebA (Liu et al. 2015), for our experiments. To create the ImageNet-10 dataset, we randomly select 10 classes from ImageNet-1K (Deng et al. 2009). For CelebA, we follow the recommended configuration from (Salem et al. 2020) to choose three most balanced attributes, namely Heavy Makeup, Mouth Slightly Open, and Smiling, and concatenate them to form eight compound classes for a multi-label classification task. To construct the classifier f, we utilize the Pre-activation ResNet-18 (He et al. 2016) for CIFAR-10 and ResNet-18 for both ImageNet-10 and CelebA. In all experiments, we use the same backbone between h and f. Additionally, we design the generator function g with a U-Net (Ronneberger, Fischer, and Brox 2015) backbone.

For each experiment, we simulate the entire data and model poisoning process and assess the accuracy of the victim model on both clean and poisoned data. Models are trained for 200 epochs using SGD optimizer. We use a batch size of 128 for CIFAR-10 and CelebA and 32 for ImageNet-10. The initial learning rate is set to 0.01 for CIFAR-10 and CelebA, and 0.001 for ImageNet-10, which is decreased tenfold at epoch 100 and 150. We use the target class $\mathfrak{c} = 0$ across all tests. The target-class training images are poisoned to achieve an overall poisoning rate of 5%. We set λ_{ℓ_2} and λ_{d} as 0.02 and 0.8, respectively. For the high-frequency removal tricks, we choose ratio $r = 0.65$ and use Gaussian blur filter with kernel size of 3 and standard deviation σ uniformly sampled from $[0.1, 1]$.

4.2 Attack Experiments

White-box settings We first conduct experiments on the standard backdoor setting, where the attacker has prior knowledge about the victim's model training, so they can match the surrogate model's architecture to the victim model's. To ensure imperceptible triggers, we set a small value of η to $10/255$. We report the test performance of the victim models in Table 1. Across all datasets, these models have similar accuracy as the clean counterpart on clean data. Our method achieves near-perfect ASRs on CIFAR-10 and CelebA. Even on ImageNet-10, it still achieves a high ASR of 83.78%, confirming its effectiveness for large images.

Next, we compare our method with the existing clean-label attacks on CIFAR-10 in Table 2. The baselines include BadNets (Gu, Dolan-Gavitt, and Garg 2017), Label-consistent (Turner, Tsipras, and Madry 2019), SIG (Barni, Kallas, and Tondi 2019), Sleeper Agent (Souri et al. 2021), and Narcissus (Zeng et al. 2022). Note that Sleeper Agent

Dataset	η	p (%)	OA (%)	BA (%)	ASR (%)
CIFAR-10	10/255	5.00	94.77	94.58	97.73
ImageNet-10	10/255	5.00	85.00	88.60	83.78
CelebA	10/255	5.00	79.34	79.41	99.84

Table 1: Attack performance on different datasets. For each dataset, we report the benign accuracy (BA) of victim models on clean inputs and the attack success rates (ASR) of backdoored samples. Additionally, we report the original accuracy (OA) of the corresponding clean models as a reference.

Method	Standard setting			Low poisoning rate ($p = 0.05\%$)			Tight constraint ($\eta = 4/255$)	
	η	BA (%)	ASR (%)	η	BA (%)	ASR (%)	BA (%)	ASR (%)
BadNets	255/255	94.99	5.49	255/255	94.82	0.79	94.40	0.81
Label Consistent	255/255	94.78	65.69	255/255	95.00	0.79	94.27	0.47
SIG	25/255	94.72	69.35	25/255	94.54	0.27	94.34	0.78
Sleeper Agent	12/255	91.43	60.90	16/255	91.74	9.06	90.61	10.57
Narcissus	10/255	**95.06**	89.09	16/255	**95.37**	47.86	**94.99**	70.12
Ours	10/255	94.58	**97.73**	16/255	94.80	**72.31**	94.56	**83.20**

Table 2: Comparison between clean-label attacks on CIFAR-10. We consider one standard and two extreme attack scenarios. We report the benign accuracy (BA) of victim models on clean inputs and the attack success rates (ASR) of backdoored samples. For a fair comparison, we do not apply $\times 3$ amplification when evaluating the Narcissus attack.

computes ASR on a sampled source image set, but we modified their code to compute ASR on the poisoned test images. Moreover, Narcissus amplifies the trigger noise at inference to enhance the ASR, but we report its performance without such amplification to ensure a fair comparison. We evaluate all methods in one standard and two extreme scenarios: extremely low poisoning rate of $p = 0.05\%$ (only 25 poisoned samples) and tight trigger norm constraint of $\eta = 4/255$. In the standard setting, all methods except BadNets can achieve at least 60% ASR. However, only our method can reach near-perfect performance with 97.73% ASR, outperforming the others by a significant margin. When poisoning only 25 examples, only Narcissus and COMBAT manage to implant the backdoors to the victim models. While Narcissus could not pass 50% ASR, our method produces 72.31% ASR. Finally, even with the tight constraint, our approach achieves 83.20% ASR, while the others fail to reach 75% ASR.

Black-box settings In real-world scenarios, it is highly unlikely for attackers to possess prior knowledge about the victim's model architecture. However, even in such circumstances, our learned generators are able to produce transferable triggers that can effectively target victims with different backbones than the surrogate models. To demonstrate this, we conduct a series of transfer attack experiments, and the results are reported in Table 3. Our tests include various victim backbones such as MobileNetV2, VGG13, and Vit-Small-8. In most cases, the transferred ASRs are greater than 80%, except for ViT-Small-8 on ImageNet-10. We attribute this result to the small size of the dataset (13,000 training images) as it may not be sufficient to train a Transformer-based classifier. This claim is supported by the fact that when we use a significantly larger dataset like CelebA (over 160,000 images), our attack achieves a near-perfect ASR of 99.77%.

4.3 Defense Experiments

In this section, we evaluate our proposed backdoor attack against several popular defenses, namely, Frequency-based defense (Zeng et al. 2021b), Neural Cleanse (Wang et al. 2019), Fine-pruning (Liu, Dolan-Gavitt, and Garg 2018), STRIP (Gao et al. 2019) and GradCAM (Xu et al. 2020; Doan, Abbasnejad, and Ranasinghe 2020). More defense experiments can be found in the Appendix.

Frequency-based defense is a data defense that involves training a detector to recognize poisoned samples in the frequency domain. This method is highly effective, as many existing backdoor attacks generate easily detectable high-frequency artifacts. We address it by applying high-frequency removal tricks in Section 3.2, which effectively reduce the detection rate on all datasets (Table 4). We provide a more in-depth analysis in the Appendix.

Neural Cleanse is a widely used model defense. It computes for each class an optimal class-inducing pattern, then detects if there is an abnormally smaller pattern among them, using an anomaly index computed by an outlier detection algorithm. If an index is greater than 2, the model will be marked as backdoor. COMBAT passes Neural Cleanse for all datasets (Fig. 1c).

Fine-pruning is another model defense that focuses on neuron analysis. It gradually prunes the neurons that are inactive when predicting clean images, assuming they are more likely linked to the backdoor. We run it on our victim models and plot the clean (BA) and backdoor (ASR) accuracy w.r.t. the number of neurons pruned in Fig. 1a. The defense can not mitigate our backdoor since there is no point with high BA and low ASR.

STRIP is a common test-time defense. Given the model and

Dataset	Surrogate Model	Victim model		
		MobileNetV2	VGG13	ViT-Small-8
CIFAR-10	PreActResNet18	93.76 / 98.70	93.76 / 97.10	76.92 / 87.10
ImageNet-10	ResNet18	89.20 / 88.22	91.60 / 80.89	81.20 / 23.11
CelebA	ResNet18	79.79 / 99.48	78.85 / 97.57	76.84 / 99.77

Table 3: Transfer attack to different victim backbones, each cell shows BA (%) / ASR (%).

```
Algorithm 1: COMBAT
Input: Training data set $\mathcal{S}$, target label $\mathfrak{c}$, injection rate $p$, poison magnitude $\eta$, number of training iteration $N$, a clean classifier $h_\psi$, hyper-parameters $\lambda_{\ell_2}$ and $\lambda_{\mathsf{d}}$.
Stage 1: Find the optimal trigger function $\mathcal{G}_{\phi^*}$
initialize $\phi$ and $\theta$
for the number of iterations $< N$ do
    Randomly sample a mini-batch $\mathcal{S}_{\mathsf{mini}}$ from $\mathcal{S}$
    Find $\mathcal{S}^{\mathfrak{c}}_{\mathsf{mini}}$ as the subset of $\mathcal{S}_{\mathsf{mini}}$ with the class label $\mathfrak{c}$
    Randomly sample $\mathcal{P}^{\mathfrak{c}}_{\mathsf{mini}}$ from $\mathcal{S}^{\mathfrak{c}}_{\mathsf{mini}}$ with ratio $p$
    Update $\theta$: $\min_{\theta} \sum_{(x_j,y_j)\in\mathcal{S}_{\mathsf{mini}}\setminus\mathcal{P}^{\mathfrak{c}}_{\mathsf{mini}}} \mathcal{L}(f_\theta(x_j),y_j) + \sum_{(x_j,\mathfrak{c})\in\mathcal{P}^{\mathfrak{c}}_{\mathsf{mini}}} \mathcal{L}\big(f_\theta(\mathcal{G}_\phi(x_j)),\mathfrak{c}\big)$
    Update $\phi$: $\min_{\phi} \sum_{(x_j,y_j)\in\mathcal{S}_{\mathsf{mini}}} \Big[\mathcal{L}\big(f_\theta(\mathcal{G}_\phi(x_j)),\mathfrak{c}\big) +$
    $\lambda_{\ell_2}\|\eta\mathcal{Q}(g_\phi(x_j))\|_2 + \lambda_{\mathsf{d}}\mathcal{L}\big(h_\psi(\mathcal{G}_\phi(x_j)),y_j\big)\Big]$
end
Stage 2: Generate the poisoned dataset $\mathcal{S}_b$
Find $\mathcal{S}^{\mathfrak{c}}$ as the subset of $\mathcal{S}$ with the class label $\mathfrak{c}$
Randomly sample $\mathcal{P}^{\mathfrak{c}}$ from $\mathcal{S}^{\mathfrak{c}}$ with ratio $p$
$\mathcal{P}^{\mathfrak{c}}_b \leftarrow \emptyset$
for $(x,y)$ in $\mathcal{P}^{\mathfrak{c}}$ do
    $\mathcal{P}^{\mathfrak{c}}_b \leftarrow \mathcal{P}^{\mathfrak{c}}_b \cup \{(\mathcal{G}_\phi(x),y)\}$
end
$\mathcal{S}_b \leftarrow (\mathcal{S}\setminus\mathcal{P}^{\mathfrak{c}}) \cup \mathcal{P}^{\mathfrak{c}}_b$
return $\mathcal{G}_{\phi^*}$ and $\mathcal{S}_b$.
```

a suspicious input, STRIP superimposes various image patterns on the input and records the prediction entropy over those perturbed images. Consistent predictions, indicated by low entropy, suggest that the sample may be poisoned. We provide STRIP's results on our models in Fig. 1b. COMBAT has a similar entropy range as that of a clean model, hence easily bypasses the defense.

GradCAM inspection was used in some studies (Xu et al. 2020; Doan, Abbasnejad, and Ranasinghe 2020) to detect abnormal network behavior for backdoor detection. We tested GradCAM on CIFAR-10 poisoned models. With BadNets, the trigger is easily caught in the GradCAM heatmaps, as shown in Fig. 1d. In contrast, our highlighted heatmap regions spread out and vary in size and position; hence our trigger stays obscure under such inspection.

	CIFAR-10	ImageNet-10	CelebA
W/o HF removal	100.00	100.00	100.00
W/ HF removal	16.20	23.33	34.33

Table 4: Effect of our high-frequency (HF) removal on the detection rate (%) of frequency-based backdoor detector.

Victim Model	$\lambda_{\mathsf{d}} = 0$		$\lambda_{\mathsf{d}} = 0.8$	
	$p = 5\%$	$p = 0\%$	$p = 5\%$	$p = 0\%$
PreActResNet18	94.72	91.29	97.73	6.60
MobileNetV2	93.91	92.54	98.70	13.66
VGG13	93.17	79.81	97.10	4.61

Table 5: Attack success rate without and with adversarial avoidance loss. The experiment is conducted on the CIFAR-10 dataset and the surrogate model is PreActResNet18.

4.4 Role of the Adversarial Avoidance Loss

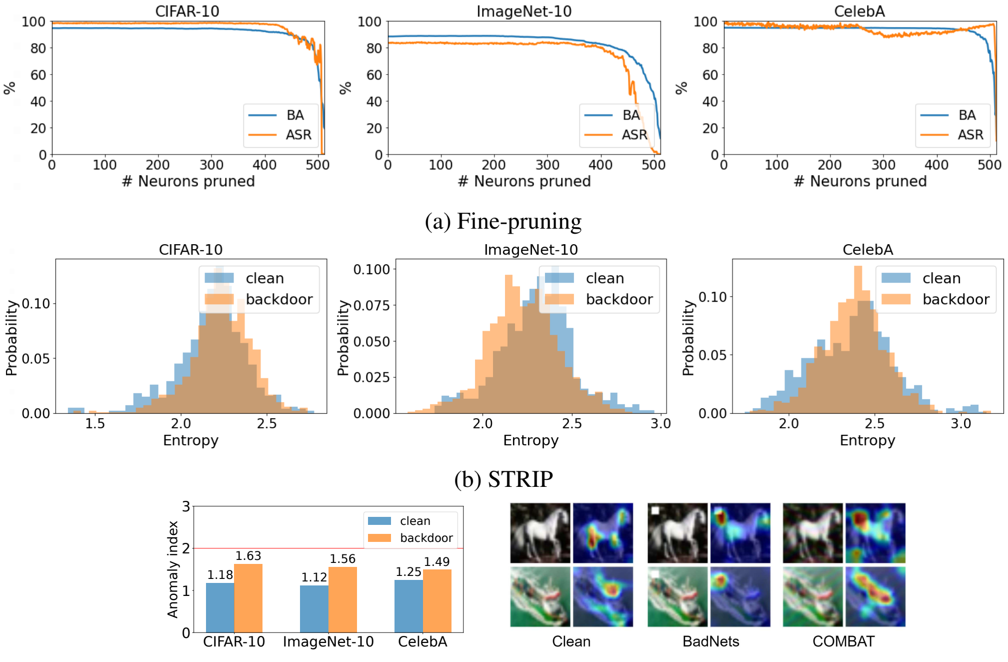
In this section, we illustrate the role of the adversarial avoidance loss $\mathcal{L}_{\mathsf{d}}$ proposed in Eq. 11.

We observe that without $\mathcal{L}_{\mathsf{d}}$, g tends to "cheat" by learning to produce universal targeted adversarial noises. These noises can fool the victim classifier during inference, regardless of whether data poisoning was present during training, contradicting the goal of backdoor attacks. Moreover, standard adversarial defenses can mitigate these adversarial noises. To demonstrate this behavior, we present our ASR with and without $\mathcal{L}_{\mathsf{d}}$ in Table 5. When $\lambda_{\mathsf{d}} = 0$, the ASR stays high across different victim backbones, regardless of the value of p. However, when $\mathcal{L}_{\mathsf{d}}$ is applied with $\lambda_{\mathsf{d}} = 0.8$, the ASR drops significantly when no data poisoning is involved. It holds true, even when the victim's backbone is different from the surrogate model, confirming that $\mathcal{L}_{\mathsf{d}}$ effectively prevents g from producing adversarial perturbations.

4.5 Ablation Studies

Alternated training. The alternated training is a key component of our proposal. It simulates the data poisoning process, providing g with an accurate representation of real-world victim models, thus increasing the attack's effectiveness. In contrast, a less effective and simplistic method is to train g using a fixed surrogate model, f, pre-trained on clean data. To compare the two approaches, we conducted experiments on CIFAR-10 with different noise strengths (η). Our approach firmly outperforms the naive one (left of Fig. 2).

Performance w.r.t poisoning rates. We investigate the im-



(a) Fine-pruning

(b) STRIP

(c) Neural Cleanse

(d) GradCAM visualization

Figure 1: Experiment results of evaluating COMBAT against defense methods

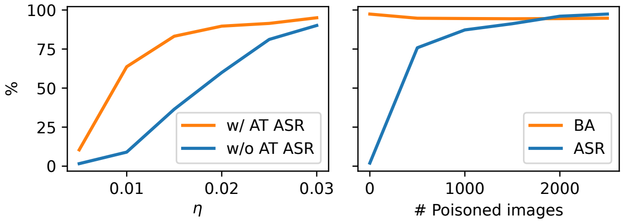


Figure 2: Ablation studies on CIFAR-10. Left: Role of alternated training. Right: Performance w.r.t poisoning rates.

pact of poisoning rate on the performance of victim models on CIFAR-10 and report the results on the right of Fig. 2. Even when using a small number of poisoned images, COMBAT can achieve a surprisingly high attack success rate. As the number of poisoned images increases, the attack success rate consistently approaches 100%.

5 Customize the Attack Configurations

Training a trigger generator offers a high level of customizability to suit the attacker's objectives. We demonstrate below an example variant of our attack with multiple target labels. More variants, such as input-aware, warping-based, or imperceptible triggers, can be found in the Appendix.

Multiple target labels. In practice, the attacker can use multiple target labels, i.e., all labels are targeted. It requires the adversary to use different triggers for different classes in order to define which label the victim network should return in an inference-time attack. This attack can be simply implemented by using multiple trigger functions $\mathcal{G}_0, \mathcal{G}_1, ..., \mathcal{G}_m$ for each target class, but it is expensive and non-scalable. Instead, we can employ a single conditional generator $\mathcal{G}(x, y)$ that inputs both an image x and a target label $y \in \{0, 1, ..., m\}$. The generated trigger is label-aware, i.e., $\mathcal{G}(x, i) \neq \mathcal{G}(x, j)\ \forall i \neq j$. From the original training set $\mathcal{S}$, we now select the poisoning set $\mathcal{P}$ covering all classes. The new poisoned dataset $\mathcal{S}_b$ is defined as follows:

$$\mathcal{S}_b = \mathcal{P}_b \cup (\mathcal{S} \setminus \mathcal{P}),\ \mathcal{P}_b = \{(\mathcal{G}(x_i, y_i), y_i) | (x_i, y_i) \in \mathcal{P}\}. \tag{14}$$

At inference time, the attacker can freely choose the target:

$$f_{\theta_b}(\mathcal{G}(x, y)) = y \qquad \forall x \in \mathcal{X}, y \in \{0, 1, ..., m\}\,. \tag{15}$$

We implement this attack on CIFAR-10 by modifying the formulation for the function $\mathcal{G}$ in Eq. 8 as follows:

$$\mathcal{G}(x, y) = (x + \eta \mathcal{Q}(g(x, y))) * k. \tag{16}$$

with $g(x, y)$ is a conditional U-Net. It achieves near-perfect results with BA at 92.48% and ASR at 99.07%.

6 Conclusions and Future Works

This paper proposes COMBAT, a framework for training clean-label backdoor attacks with outstanding efficacy. The key component is an alternated training process that optimizes together a trigger generator and a surrogate classifier. Our attack is effective, stealthy, and flexible for customization, which is extensively verified. We believe this study is crucial to understanding the potential capability of clean-label backdoor attacks, stimulating future defense studies aiming safe and trustful AI. Besides, we plan to further improve COMBAT's transferability in future studies.

References

Ahmed, N.; Natarajan, T.; and Rao, K. R. 1974. Discrete cosine transform. *IEEE transactions on Computers*, 100(1): 90–93.

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

Bober-Irizar, M.; Shumailov, I.; Zhao, Y.; Mullins, R.; and Papernot, N. 2022. Architectural Backdoors in Neural Networks. *arXiv preprint arXiv:2206.07840*.

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

Chen, H.; Fu, C.; Zhao, J.; and Koushanfar, F. 2021. Proflip: Targeted trojan attack with progressive bit flips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7718–7727.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Doan, B. G.; Abbasnejad, E.; and Ranasinghe, D. C. 2020. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Annual Computer Security Applications Conference*, 897–912.

Doan, K.; Lao, Y.; and Li, P. 2021. Backdoor attack with imperceptible input and latent modification. *Advances in Neural Information Processing Systems*, 34: 18944–18957.

Doan, K.; Lao, Y.; Zhao, W.; and Li, P. 2021. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11966–11976.

Gao, Y.; Xu, C.; Wang, D.; Chen, S.; Ranasinghe, D. C.; and Nepal, S. 2019. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, 113–125.

Garipov, T.; Izmailov, P.; Podoprikhin, D.; Vetrov, D. P.; and Wilson, A. G. 2018. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. In *Proceedings of Machine Learning and Computer Security Workshop*.

Hammoud, H. A. A. K.; and Ghanem, B. 2021. Check your other door! establishing backdoor attacks in the frequency domain. *arXiv preprint arXiv:2109.05507*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity mappings in deep residual networks. In *European conference on computer vision*, 630–645. Springer.

Kolouri, S.; Saha, A.; Pirsiavash, H.; and Hoffmann, H. 2020. Universal litmus patterns: Revealing backdoor attacks in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 301–310.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*.

Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, 273–294. Springer.

Liu, Y.; Lee, W.-C.; Tao, G.; Ma, S.; Aafer, Y.; and Zhang, X. 2019. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 1265–1282.

Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning attack on neural networks. In *Proceedings of Network and Distributed System Security Symposium*.

Liu, Y.; Ma, X.; Bailey, J.; and Lu, F. 2020. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, 182–199. Springer.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, 3730–3738.

Nguyen, A.; and Tran, A. 2020. Input-Aware Dynamic Backdoor Attack. In *Proceedings of Advances in Neural Information Processing Systems*.

Nguyen, T. A.; and Tran, T. A. 2021. WaNet - Imperceptible Warping-based Backdoor Attack. In *International Conference on Learning Representations*.

Ning, R.; Li, J.; Xin, C.; and Wu, H. 2021. Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 1–10. IEEE.

Rakin, A. S.; He, Z.; and Fan, D. 2020. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13198–13207.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Saha, A.; Subramanya, A.; and Pirsiavash, H. 2020. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11957–11965.

Salem, A.; Wen, R.; Backes, M.; Ma, S.; and Zhang, Y. 2020. Dynamic Backdoor Attacks Against Machine Learning Models. *arXiv preprint arXiv:2003.03675*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Souri, H.; Goldblum, M.; Fowl, L.; Chellappa, R.; and Goldstein, T. 2021. Sleeper Agent: Scalable Hidden Trigger Backdoors for Neural Networks Trained from Scratch. *arXiv preprint arXiv:2106.08970*.

Tran, B.; Li, J.; and Madry, A. 2018. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31.

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

Udeshi, S.; Peng, S.; Woo, G.; Loh, L.; Rawshan, L.; and Chattopadhyay, S. 2022. Model agnostic defence against backdoor attacks in machine learning. *IEEE Transactions on Reliability*.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, 707–723. IEEE.

Wang, T.; Yao, Y.; Xu, F.; An, S.; and Wang, T. 2021. Backdoor attack through frequency domain. *arXiv preprint arXiv:2111.10991*.

Xu, K.; Liu, S.; Chen, P.-Y.; Zhao, P.; and Lin, X. 2020. Defending against backdoor attack on deep neural networks. *arXiv preprint arXiv:2002.12162*.

Yao, Y.; Li, H.; Zheng, H.; and Zhao, B. Y. 2019. Latent Backdoor Attacks on Deep Neural Networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2041–2055.

Zeng, Y.; Chen, S.; Park, W.; Mao, Z. M.; Jin, M.; and Jia, R. 2021a. Adversarial unlearning of backdoors via implicit hypergradient. *arXiv preprint arXiv:2110.03735*.

Zeng, Y.; Pan, M.; Just, H. A.; Lyu, L.; Qiu, M.; and Jia, R. 2022. NARCISSUS: A Practical Clean-Label Backdoor Attack with Limited Information. *arXiv preprint arXiv:2204.05255*.

Zeng, Y.; Park, W.; Mao, Z. M.; and Jia, R. 2021b. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16473–16481.

Zhao, P.; Chen, P.-Y.; Das, P.; Ramamurthy, K. N.; and Lin, X. 2020. Bridging mode connectivity in loss landscapes and adversarial robustness. *arXiv preprint arXiv:2005.00060*.

Zhong, N.; Qian, Z.; and Zhang, X. 2022. Imperceptible backdoor attack: From input space to feature representation. *arXiv preprint arXiv:2205.03190*.