COMBAT: ALTERNATED TRAINING FOR NEAR-PERFECT CLEAN-LABEL BACKDOOR ATTACKS

Anonymous authors
Paper under double-blind review

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 near-perfect attack performance. The key component is a trigger pattern generator, which is trained together with a surrogate model in an alternate 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.

1 INTRODUCTION

Deep learning has revolutionized all domains of Artificial Intelligence, bringing a powerful tool to handle challenging tasks that were thought impossible to solve in classical works. Many deep-learning-based applications, such as face recognition and self-driving cars, have been widely deployed to change our society deeply. 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 themself 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 in terror attacks. Therefore, 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. 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 et al., 2017), a common dirty-label method, completely fails. In addition, most existing clean-label attacks (Turner et al., 2019; Barni et al., 2019) cannot reach an 80% attack

success rate. Although there are recent works (Ning et al., 2021; Zeng et al., 2022) that manage to achieve near-perfect attack success rates, 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.

The challenges in clean-label attacks lead us to believe that designing a backdoor trigger based on its poisoning effect measured on the poisoned model could be an approach to solve the problem. 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 alternatively 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. COMBAT can reach the attack success rates of 98-99% while using exceedingly small triggers on various datasets, including CIFAR-10, GTSRB, and CelebA. Our clean-label attacks are also stealthy, breaking all backdoor defense mechanisms.

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 backdoor methods to all reach high attack 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. The attacker should poison a minimum amount of data, so 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 et al., 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., 2018b; Yao et al., 2019; Rakin et al., 2020; Chen et al., 2021; Bober-Irizar et al., 2022) try to inject a trigger by modifying the model, and others try to have a stealthy and effective backdoor injection function. 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. Chen et al. (2017) blended a fixed image as the backdoor trigger. Salem et al. (2020) allowed randomly choosing the backdoor trigger from a set of locations and patterns. Nguyen & Tran (2020) employed input-dependent trigger patterns to dodge the common backdoor defenses that relied on the fixed-trigger assumption. Nguyen & Tran (2021b) designed a novel, imperceptible backdoor trigger based

on image warping. Doan et al. (2021b) optimized the backdoor trigger function during the training process towards imperceptible trigger in the input space, while later works (Doan et al., 2021a; Zhong et al., 2022) further made backdoors imperceptible in the latent space. Recent approaches (Wang et al., 2021; Hammoud & 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 et al. (2019) first time discussed this issue and proposed the clean-label attack scheme. They pointed out that if the poisoned examples were too easy to learn via their salient content, the network would ignore the trigger pattern and fail to adopt the backdoor. 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 et al. (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 et al. (2020) introduced a hidden backdoor attack via model fine-tuning that first generated a patch-based poisoned sample, then injected its information into the texture of a training image from the target class by minimizing their distance in the feature space, making the trigger invisible. Souri et al. (2021) allowed hidden attacks on training-from-scratch models by applying gradient matching. Still, all these methods had underwhelming attack performance. As reported Liu et al. (2020), the attack success rate (ASR) of Refool on GTSRB was only 91.67%, while the previous attacks obtained no more than 80%, widely lagging behind the common rate 98-99% of dirty-label methods.

A few recent clean-label attacks achieved near-perfect attack performance at the cost of stealthiness. Ning et al. (2021) used an auto-encoder to learn image-dependent noised triggers and reached 96% ASR on GTSRB (5% poisoning rate) but required a significant color shift. Narcissus (Zeng et al., 2022) optimized a universal trigger using a pre-trained clean surrogate classifier. It reached a 97.36% ASR on CIFAR-10 when using an additive-noise pattern with $\ell_\infty = 16/255$ and requiring only 0.05% data poisoned. Narcissus is the most effective attack up-to-date, and it shares many similar ideas to our method. However, it employs a fixed, clean surrogate model and optimizes a fixed, uniform trigger pattern. We will show that this design is not optimal. Instead, our method used an adaptive, poisoned surrogate model and jointly trained a generator with it in an alternated training, aiming to learn image-dependent and optimal triggers. Our method, therefore, achieves higher ASR (98.26%) with a smaller ℓ_∞ of trigger (10/255), and it is easy to be customized to different variants.

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***), trained 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 et al. (2018) filtered backdoor samples assuming a discernible trace in the spectrum of the covariance feature representations. Chen et al. (2018) relied on clustering the latent representations, assuming clean and poisoned samples had distinct characteristics in the hidden feature space. Zeng et al. (2021b) proposed an efficient data filtering mechanism based on the frequently observed high-frequency artifacts in backdoored data.

Model defense. Model defenses identify or mitigate poisoned models by inspecting their behaviors when dealing with clean data. Fine-pruning (Liu et al., 2018a) suggested pruning inactive neurons, but it could not confirm if the model was infected. Neural Cleanse (Wang et al., 2019) computed optimal class-inducing patterns for each class, then identified a poisoned model based on 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 using a set of clean data. Xu et al. (2020) utilized GradCAM (Selvaraju et al., 2017) to analyze the model's behaviors on clean images with and without the presence of engineering-reversed triggers. Zhao et al. (2020) repaired the model's backdoor by applying 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 et al. (2021) employed the knowledge distillation technique, assuming that the distillation process perturbs 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 et al., 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.

Besides trigger-based backdoor attacks, there are studies on *triggerless* data poisoning attacks, in which the attacker poisons training data to make the trained model misclassify an individual clean image or class. A practical approach is treating it as a bilevel optimization problem (Huang et al., 2020; Geiping et al., 2020), which simulates the training procedure and searches for optimal poisoned data directly. The corresponding defenses either pre-filter data with outlier detection Paudice et al. (2018), detect poisoned data near the target class's distribution Peri et al. (2020), or utilize strong augmentations Borgnia et al. (2021). While this line of research is irrelevant to backdoor attacks, it shares an idea of treating data poisoning as a bilevel optimization with our method. We solve that problem with alternated learning, as will be discussed in the next section.

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 $\mathcal{S} = \big\{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{C}, i = 1, 2, \ldots, n\big\}$ and $\mathcal{C} = \big\{0, 1, \ldots, m\big\}$, then $\mathcal{S} = \bigcup_{k=0}^{m} \mathcal{S}^k$, where $\mathcal{S}^k$ denotes the subset of data for class k.

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 $\mathcal{P}^\mathfrak{c} \subset \mathcal{S}^\mathfrak{c}$, given a pre-defined 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 each image data in $\mathcal{P}^\mathfrak{c}$ to form a poisoned subset $\mathcal{P}_b^\mathfrak{c}$. For example, in (Turner et al., 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 is kept unchanged. The combined set $\mathcal{S}_b := (\mathcal{S} \setminus \mathcal{P}^\mathfrak{c}) \cup \mathcal{P}_b^\mathfrak{c}$ 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, i.e., cross-entropy function. 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 $\mathfrak{c}$ regardless of image content:

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

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

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

3.2 TRIGGER GENERATOR

For simplicity, we consider $\mathcal{G}$ as a noise-additive function:

$$\mathcal{G}(x) = x + \eta g(x), \tag{5}$$

with $g(\cdot)$ generates a trigger noise in $[-1, 1]$ conditioned on the image input and η is the ℓ_∞ bound of the added noise. We will discuss some more complex backdoor functions in Section 5. Note that

we use image-dependent trigger instead of a fixed, global one like in most of previous works. It is more flexible and allows to define extra properties like input-awareness (Nguyen & Tran, 2020).

We formulate $g(\cdot)$ as a neural network of hyper-parameter ϕ and use directly its attack performance as an objective function for training. 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 class $\mathbf{c}$ as well as the magnitude of the trigger. In the formula, these loss terms are defined as follows:

$$\mathcal{L}_a(f_\theta, g_\phi; \mathcal{S}, \eta) := \sum_{(x_j, y_j) \in \mathcal{S}} \mathcal{L}(f_\theta(x_j + \eta g_\phi(x_j)), \mathbf{c}) \tag{6}$$

$$\mathcal{L}_{\ell_2}(g_\phi; \mathcal{S}, \eta) := \sum_{(x_j, y_j) \in \mathcal{S}} \|\eta g_\phi(x_j)\|_2. \tag{7}$$

Besides, a recent work (Zeng et al., 2021b) has shown that poisoned data may contain high-frequency artifacts that are imperceptible to humans but are highly detectable by DNN-based detectors. To mitigate the problem and promote even stronger stealthiness, we propose the following frequency-based regularization:

$$\mathcal{L}_{\mathsf{freq}}(f_\theta, g_\phi, h; \mathcal{S}, \eta) := \sum_{(x_j, y_j) \in \mathcal{S}} \mathcal{L}(h(\text{DCT}(x_j + \eta g_\phi(x_j))), \mathbf{0}), \tag{8}$$

where DCT$(\cdot)$ is the Discrete Cosine Transform (Ahmed et al., 1974) and h is a network to classify clean and poisoned images based on their frequency signatures following (Zeng et al., 2021b).

The final loss function used to optimize g_ϕ:

$$F_1(f_\theta, g_\phi, h; \mathcal{S}, \lambda, \eta) := \mathcal{L}_a(f_\theta, g_\phi; \mathcal{S}, \eta) + \lambda_{\ell_2} \mathcal{L}_{\ell_2}(g_\phi; \mathcal{S}, \eta) + \lambda_{\mathsf{freq}} \mathcal{L}_{\mathsf{freq}}(f_\theta, g_\phi, h; \mathcal{S}, \eta), \tag{9}$$

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

3.3 Alternated Training

As we discussed in the previous section, 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 g_ϕ. However, we need g 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 g_ϕ with F_1, we concurrently optimize f_θ with another loss function:

$$F_2(f_\theta, g_\phi; \mathcal{S} \setminus \mathcal{P}^{\mathbf{c}}, \mathcal{P}_b^{\mathbf{c}}, \eta) := \sum_{(x_j, y_j) \in \mathcal{S}} \mathcal{L}(f_\theta(x_j), y_j) + \sum_{(x_j, \mathbf{c}) \in \mathcal{P}_b^{\mathbf{c}}} \mathcal{L}(f_\theta(x_j + \eta g_\phi(x_j)), \mathbf{c}), \tag{10}$$

where F_2 involves the real data and the poisoned data of the target class to find the best classifier. Optimizing F_2 mimics the process of training the victim classifier. We note that minimizing both F_1 and F_2 is like finding a balancing between trigger's magnitude and class's boundary. We argue that the alternated training could allow smaller trigger's size. Let's assume that the clean data lie in the manifold as in Figure 1. Without alternated training, the classifier f_θ would use the line in the right middle of blue squares and blue circles to separate them. The trigger would try to push blue squares to cross its class's boundary to get into the region of blue circles of the target class. However, with the inclusion of poisoned data (red circles) in training, the classifier will move the separating line closer to the blue squares. It consequently allows the magnitude of trigger for blue squares gets smaller.

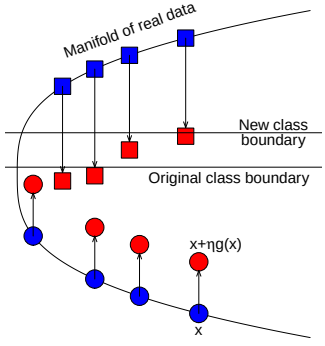


Figure 1: Toy example of circle and square classes. True data are in blue, and poisoned data are in red.

After the alternated training process, the attacker acquires the optimal trigger generator $\mathcal{G}$, then uses it to acquire the poisoned dataset $\mathcal{S}_b$ as described in Section 3.1. 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 behaves similar as the surrogate model, which we will empirically prove in Section 4.3. Finally, the attacker could use the "optimal" trigger function to generate poisoned data from other classes to fool the victim model.

Algorithm 1: COMBAT

Input: Training data set $\mathcal{S}$, target label $\mathfrak{c}$, injection rate p, poison magnitude η, number of training iteration N, surrogate frequency-backdoor detector h, hyper-parameters λ_{ℓ_2} and λ_{freq}

Stage 1: Find the trigger function $\mathcal{G}$
initialize noise function g_ϕ and the surrogate model f_θ
for *the number of iteration is less than* N **do**
  Randomly choose a mini-batch of sample of $\mathcal{S}$ denoted by $\mathcal{S}_{\text{mini}}$
  Find $\mathcal{S}^{\mathfrak{c}}_{\text{mini}}$ as the subset of $\mathcal{S}_{\text{mini}}$ with the class label $\mathfrak{c}$
  Randomly choose a subset with ratio p of $\mathcal{S}^{\mathfrak{c}}_{\text{mini}}$, denoted by $\mathcal{P}^{\mathfrak{c}}_{\text{mini}}$
  Train f_θ**:** $\min_\theta \sum_{(x_j,y_j)\in\mathcal{S}_{\text{mini}}\setminus\mathcal{P}^{\mathfrak{c}}_{\text{mini}}} \mathcal{L}(f_\theta(x_j), y_j) + \sum_{(x_j,\mathfrak{c})\in\mathcal{P}^{\mathfrak{c}}_{\text{mini}}} \mathcal{L}\big(f_\theta(x_j+\eta g_\phi(x_j)), \mathfrak{c}\big)$
  Train g_ϕ**:**
  $\min_\phi \sum_{(x_j,y_j)\in\mathcal{S}_{\text{mini}}} \Big[\mathcal{L}\big(f_\theta(x_j+\eta g_\phi(x_j)), \mathfrak{c}\big) + \lambda_{\ell_2}\|\eta g_\phi(x_j)\|_2 + \lambda_{\text{freq}}\mathcal{L}(h(\text{DCT}(x_j+\eta g_\phi(x_j))), \mathbf{0})\Big]$
end
$\mathcal{G}(x) \leftarrow x + \eta g_\phi(x)$

Stage 2: Generate $\mathcal{S}_b$
Find $\mathcal{S}^{\mathfrak{c}}$ as the subset of $\mathcal{S}$ with the class label $\mathfrak{c}$
Randomly choose a subset with ratio p of $\mathcal{S}^{\mathfrak{c}}$, denoted by $\mathcal{P}^{\mathfrak{c}}$
$\mathcal{P}^{\mathfrak{c}}_b \leftarrow \emptyset$
for (x, y) *in* $\mathcal{P}^{\mathfrak{c}}$ **do**
  $\mathcal{P}^{\mathfrak{c}}_b \leftarrow \mathcal{P}^{\mathfrak{c}}_b \cup \{(\mathcal{G}(x), y)\}$
end
$\mathcal{S}_b \leftarrow (\mathcal{S} \setminus \mathcal{P}^{\mathfrak{c}}) \cup \mathcal{P}^{\mathfrak{c}}_b$
return Trigger function $\mathcal{G}$ and the poisoned dataset $\mathcal{S}_b$.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We choose three widely-used datasets to conduct our experiments: CIFAR10 (Krizhevsky et al., 2009), GTSRB (Stallkamp et al., 2012), and CelebA (Liu et al., 2015). In CelebA, we follow the suggested configuration from (Salem et al., 2020) to select three most balanced attributes (Heavy Makeup, Mouth Slightly Open, and Smiling) and concatenate them to form eight compound classes for a multi-label classification task. To build the classifier f, we use Pre-activation ResNet-18 (He et al., 2016) for CIFAR10 and GTSRB, and ResNet-18 for CelebA. We design the generator function g with U-Net (Ronneberger et al., 2015) architecture. Details can be found in the Appendix A.3.

In each experiment, we mimic the entire data and model poisoning process, then evaluate the clean and attack accuracy of the victim model. In all model training, we use the SGD optimizer. The initial learning rate is 0.01, which is reduced 10 times for each 100 epochs until the model converges. We use the same target class $\mathfrak{c} = 0$ in all tests. We also set $\lambda_{\ell_2} = 0.02$ and $\lambda_{\text{freq}} = 0.08$.

4.2 ATTACK EXPERIMENTS

We first test our proposed attacks on normal settings. A half of clean target-class training images are poisoned , leading to the overall poisoning rate p on CIFAR-10, GTSRB, and Celeb-A as 5%, 0.57%, and 13.40%, respectively. We employ small η values to guarantee invisibility of the trigger. In particular, η is set as $10/255$ on CIFAR-10/CelebA and $20/255$ on GTSRB. We found that GTSRB is a bit harder to attack, thus doubled the generated noise before adding to the test images following the recommendation of (Zeng et al., 2022). We report the victim models' performance in Table 1. In all cases, the clean accuracy of the poisoned model matches well the accuracy of the clean counterpart. All attack success rates are higher than 98%, confirming near-perfect attack efficacy.

Next, we compare our method with the existing clean-label attack methods on the CIFAR-10 dataset. The baselines include BadNets (Gu et al., 2017) (to confirm its inefficacy in this setting), Label-consistent (Turner et al., 2019), SIG (Barni et al., 2019), Sleeper Agent (Souri et al., 2021), and Narcissus (Zeng et al., 2022). Note that Sleeper Agent computes attack success rate (ASR) on a

Table 1: **Attack performance of the victim models.** For each model, we report its test accuracy on benign inputs (BA), the attack success rate on poisoned test data (ASR), and the original accuracy from the corresponding clean model as a reference. The triggers are amplified when evaluating on GTSRB.

Dataset	η	$p(\%)$	Amplification	Original acc.(%)	BA(%)	ASR(%)
CIFAR-10	10/255	5.00	-	94.77	94.79	98.26
GTSRB	20/255	0.57	$\times 2$	99.54	99.38	98.48
CelebA	10/255	13.40	-	79.34	78.56	99.88

Table 2: **Comparison between clean-label attacks on CIFAR-10.** For each model, we test with a normal and an extreme scenario, with the poisoning rates are 5% and 0.05%, respectively. We report the victim model's test accuracy on begin inputs (BA) and attack success rate on poisoned test data (ASR). Narcissus models are tested with $\times 3$ amplification (as used in the paper) and without it.

Method	Ampl.	η	$p(\%)$	BA(%)	ASR(%)	η	$p(\%)$	BA(%)	ASR(%)
BadNets	-	255/255	5	94.99	5.49	255/255	0.05	94.82	0.79
LabelConsistent	-	255/255	5	94.78	65.69	255/255	0.05	95.00	0.79
SIG	-	25/255	5	94.72	69.35	25/255	0.05	94.54	0.27
Sleeper Agent	-	12/255	5	91.43	60.90	16/255	0.05	91.74	9.06
Narcissus	$\times 3$	10/255	5	95.06	100.00	16/255	0.05	95.37	98.73
Narcissus	-	10/255	5	95.06	89.09	16/255	0.05	95.37	47.86
Ours	-	10/255	5	94.79	98.26	16/255	0.05	95.10	99.64

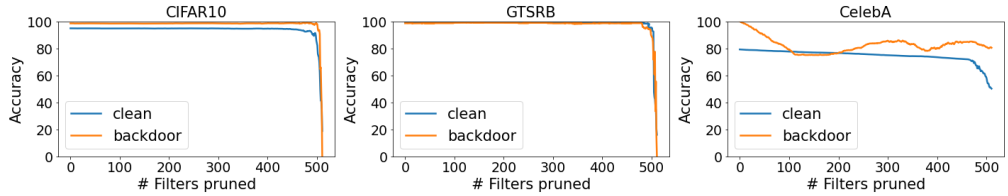
sampled source image set, and we corrected its code to compute ASR on the poisoned test images. Besides, Narcissus amplifies the trigger noise three times before injecting it into images at inference time. We report Narcissus's performance with and without such amplification to get a fair comparison with the other approaches. We examine one normal and one extreme scenario, with the poisoning rates p as 5% (2500 poisoned samples) and 0.05% (25 poisoned samples), respectively. All results are reported in Table 2. As can be seen, in the easy setting, all methods except BadNets can poison the victim model with at least 60% ASR. However, only Narcissus and our method can reach near-perfect attack performance, and they outperform the others by a wide margin. Also, Narcissus can only achieve such impressive results by amplifying the trigger at test time. Without the amplification, its fooling rate is only 89.09%. In contrast, COMBAT can reach a 98.25% ASR in that fair condition. In the extreme scenario, only Narcissus and our COMBAT manage to pass the backdoor to the end model. With a higher η budget, our method can achieve 99.64% ASR even with such a low poisoning rate. It even surpasses the attack performance of Narcissus when applying the $\times 3$ amplification trick.

4.3 Transfer Experiments

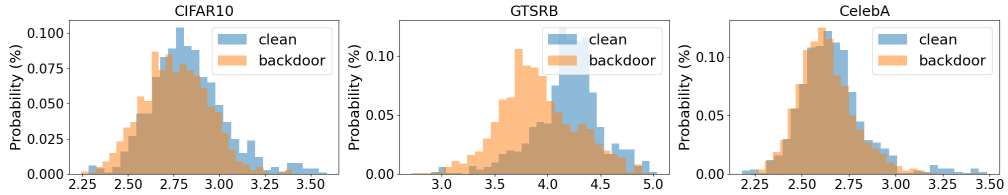
The attacker trains the surrogate model without knowledge about the victim network, and these two models likely have different structures. Still, we found that the learned triggers were highly transferable to different victim backbones. We run a series of transfer experiments to confirm it. While the source (surrogate) backbone is PreActResNet18/ResNet18, we test vastly different target (victim) backbones , including MobileNetV2, VGG13, and ViT-Small-8. The results are provided in Table 3. In most experiments, the transferred ASR is at least 95%, confirming the observed high transferability. The target ViT models on CIFAR-10 and GTSRB have exceptionally lower clean and attack accuracy, possibly because these small datasets are unsuitable to train ViT models.

Table 3: Transferability to different victim's backbones, with BA (%) in teal and ASR (%) in purple.

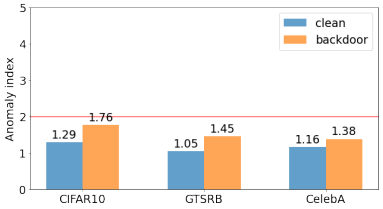
Dataset	Source Model	Target model		
		MobileNetV2	VGG13	ViT-Small-8
CIFAR10	PreActResNet18	94.28 / 98.07	93.79 / 92.86	77.89 / 84.77
GTSRB	PreActResNet18	99.16 / 95.45	99.03 / 85.00	94.35 / 86.73
CelebA	ResNet18	78.89 / 96.93	78.58 / 99.69	75.00 / 100.00



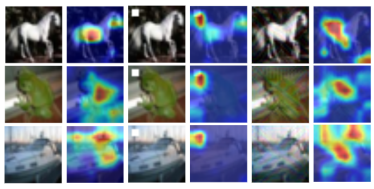
(a) Fine-pruning



(b) STRIP



(c) Neural Cleanse



(d) GradCAM visualization

Figure 2: Experiment results of evaluating COMBAT against defense methods

4.4 Defense Experiments

In this section, we evaluate our proposed backdoor attack against several popular defense methods. Extra defense results can be found in the Appendix.

Frequency-based defense (Zeng et al., 2021b) is a data defense method. It trained a detector to recognize poisoned samples in the frequency domain. This straightforward defense can effectively detect poisoned data in existing attacks. We address it by a frequency-based loss $\mathcal{L}_{\text{freq}}$ (Section 3.2). As shown in Table 4, our loss effectively reduces the backdoor detection rate on all datasets. The detection rate is still low even when the test detector's backbone (VGG13 - 9.5M params) is different from the backbone used for the loss (customized CNN (Zeng et al., 2021b) - 0.3M params).

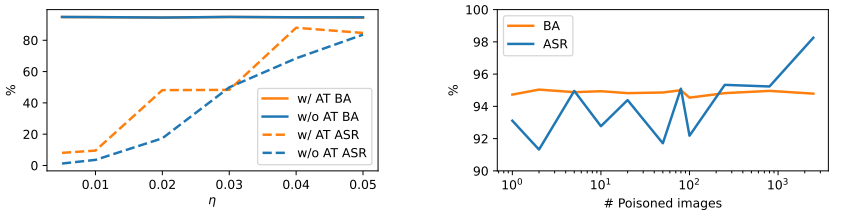
Table 4: Effect of our frequency-based loss on the detection rate (%) of frequency-based backdoor detector on COMBAT.

	CIFAR10	GTSRB	CelebA
W/o $\mathcal{L}_{\text{freq}}$	100.00	99.86	100.00
Same backbone	15.09	40.75	21.35
Transfer b.bone	31.87	27.72	52.07

Neural Cleanse (Wang et al., 2019) is a widely used model defense method. 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 a label obtains an anomaly index greater than 2, it will be marked as backdoor. We run Neural Cleanse on our victim models and report the results in Fig. 2c. COMBAT passes Neural Cleanse for all datasets.

Fine-pruning (Liu et al., 2018a) is another model defense method that focuses on neuron analysis. It detects and gradually prunes the neurons that are inactive when predicting clean images, assuming they are more likely linked to the backdoor. We run Fine-pruning on our victim models and plot the clean (BA) and backdoor (ASR) accuracy with respect to the number of neurons pruned in Fig. 2a. There is no point with high BA and low ASR, implying this defense can not mitigate our backdoor.

STRIP (Gao et al., 2019) is a common test-time defense. Given the model and 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 is likely to be poisoned. We provide the results of STRIP on our victim models in Fig. 2b. COMBAT has a similar entropy range as that of a clean model, hence easily passing the defense.



(a) Role of alternated training (AT) (b) Performance w.r.t poisoning rates

Figure 3: Ablation Studies on CIFAR-10 dataset

GradCAM inspection was used in some studies (Xu et al., 2020; Doan et al., 2020) to detect abnormal network behavior for backdoor detection. With patch-based backdoor attacks like BadNets, the backdoor trigger is easily caught in the GradCAM heatmaps, as shown in Fig. 2d. We tested GradCAM on our CIFAR-10 poisoned model. Unlike from BadNets, the highlighted heatmap regions spread out and vary in size and position; hence our trigger stays obscure under such inspection.

4.5 Ablation Studies

Alternated training. The alternated training process is a key component of our proposal. It helps to obtain a poisoned surrogate model that is as close to the victim one as possible, thus boosting the attack efficiency. Instead, a naive approach is to train a fixed, clean surrogate model and use it to optimize the generator $\mathcal{G}$. We compare these two approaches on CIFAR-10, using different noise strengths η in Fig. 3a. While both methods provide stable clean accuracy, the alternated training approach consistently outperforms the naive one on ASR.

Performance w.r.t poisoning rates. We examine the effect of the poisoning rate on the victim model performance on CIFAR-10 and report the results in Fig. 3b. Impressively, COMBAT manages to reach a very high attack success when only a few images are poisoned. When the number of poisoned images is large enough, increasing p leads to increasing ASR towards 100%.

5 Customize the Attack Configurations

The key component of COMBAT is the alternated training process. Other components, including the trigger type and the loss components, can be customized based on the need of the attacker. We demonstrate below an example variant that employs image warping as the backdoor trigger. Some other variants such as input-aware trigger or multi-target attack can be found in the Appendix.

Warping-based trigger patterns. In this section, we testify the usage of a warping-based trigger function (Nguyen & Tran, 2021a). Specifically, we change the definition of $\mathcal{G}$ as follow:

$$\mathcal{G}(x) = \mathcal{W}(x, \Phi(I + \eta \cdot \uparrow g^k(x))), \tag{11}$$

with $\mathcal{W}(\cdot)$ is the image warping function that takes the input image and a normalized warping grid, $g^k(x)$ generates a trigger grid at some resolution $k \times k$, $\uparrow$ upsamples that grid to the same resolution as the input, I is the identical warping grid, and function $\Phi(\cdot)$ clips the grid values to be in the range $[-1, 1]$. We can plug this function directly into our system and keep the same loss functions for optimization. We test a simple attack on CIFAR-10 with $k = 2$, $\eta = 0.15$, and $p = 5\%$. It successfully poisons the victim model with BA as 92.22 % and ASR as 94.18 %.

6 Conclusions and Future Works

This paper proposes COMBAT, a framework for training clean-label backdoor attacks with near-perfect performance. The key component is an alternated training process that optimizes together a trigger generator and a surrogate classifier model. Our attack is effective, stealthy, and flexible for customization, which is extensively verified on various datasets and experimental configurations. We believe this study is crucial to understanding the potential capability of clean-label backdoor attacks, stimulating future defense studies aiming toward safe and trustful AI. Besides, since COMBAT shows unstable transferability in some experiments, we plan to fix this weakness in future studies.

REFERENCES

Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

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

Mikel Bober-Irizar, Ilia Shumailov, Yiren Zhao, Robert Mullins, and Nicolas Papernot. Architectural backdoors in neural networks. *arXiv preprint arXiv:2206.07840*, 2022.

Eitan Borgnia, Jonas Geiping, Valeriia Cherepanova, Liam Fowl, Arjun Gupta, Amin Ghiasi, Furong Huang, Micah Goldblum, and Tom Goldstein. Dp-instahide: Provably defusing poisoning and backdoor attacks with differentially private data augmentations. *arXiv preprint arXiv:2103.02079*, 2021.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Proflip: Targeted trojan attack with progressive bit flips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7718–7727, 2021.

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

Pierre Comon. Independent component analysis, 1992.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Annual Computer Security Applications Conference*, pp. 897–912, 2020.

Khoa Doan, Yingjie Lao, and Ping Li. Backdoor attack with imperceptible input and latent modification. *Advances in Neural Information Processing Systems*, 34:18944–18957, 2021a.

Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11966–11976, 2021b.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113–125, 2019.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

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

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

Hasan Abed Al Kader Hammoud and Bernard Ghanem. Check your other door! establishing backdoor attacks in the frequency domain. *arXiv preprint arXiv:2109.05507*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091, 2020.

Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 301–310, 2020.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 273–294. Springer, 2018a.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Proceedings of Network and Distributed System Security Symposium*, 2018b.

Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1265–1282, 2019.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pp. 182–199. Springer, 2020.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Proceedings of Advances in Neural Information Processing Systems*, 2020.

Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021a. URL `https://openreview.net/forum?id=eEn8KTtJOx`.

Tuan Anh Nguyen and Tuan Anh Tran. WaNet - Imperceptible Warping-based Backdoor Attack. In *International Conference on Learning Representations*, 2021b.

Rui Ning, Jiang Li, Chunsheng Xin, and Hongyi Wu. Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2021.

Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C Lupu. Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv preprint arXiv:1802.03041*, 2018.

Neehar Peri, Neal Gupta, W Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, and John P Dickerson. Deep k-nn defense against clean-label data poisoning attacks. In *European Conference on Computer Vision*, pp. 55–70. Springer, 2020.

Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13198–13207, 2020.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11957–11965, 2020.

Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. *arXiv preprint arXiv:2003.03675*, 2020.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Hossein Souri, Micah Goldblum, Liam Fowl, Rama Chellappa, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *arXiv preprint arXiv:2106.08970*, 2021.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, and Sudipta Chattopadhyay. Model agnostic defence against backdoor attacks in machine learning. *IEEE Transactions on Reliability*, 2022.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723. IEEE, 2019.

Tong Wang, Yuan Yao, Feng Xu, Shengwei An, and Ting Wang. Backdoor attack through frequency domain. *arXiv preprint arXiv:2111.10991*, 2021.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925, 2021.

Kaidi Xu, Sijia Liu, Pin-Yu Chen, Pu Zhao, and Xue Lin. Defending against backdoor attack on deep neural networks. *arXiv preprint arXiv:2002.12162*, 2020.

Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2041–2055, 2019.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. *arXiv preprint arXiv:2110.03735*, 2021a.

Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16473–16481, 2021b.

Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. *arXiv preprint arXiv:2204.05255*, 2022.

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

Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. *arXiv preprint arXiv:2005.00060*, 2020.

Nan Zhong, Zhenxing Qian, and Xinpeng Zhang. Imperceptible backdoor attack: From input space to feature representation. *arXiv preprint arXiv:2205.03190*, 2022.

A APPENDIX

A.1 ETHIC STATEMENT

Our work extends the understanding of the potential capability of clean-label backdoor attacks; hence, it benefits both the research community and real-life AI systems. By being informed about the risk, AI system developers will be more careful when using third-party or open-source datasets. The work also stimulates future backdoor defense studies aiming toward safe and trustful AI.

Undeniably, the adversaries can also take advantage of our work by employing COMBAT to design more effective clean-label attacks. Still, we believe that more advanced defense methods will soon be developed after our paper to counter the risk, and the positive benefits of our paper outweigh its negative impact.

A.2 REPRODUCIBILITY STATEMENT

Our work is highly reproducible. All datasets used in the paper are popular and publicly available. We include in the supplementary materials our code and pre-trained models. The code and pre-trained models will also be publicly released upon paper acceptance.

A.3 SYSTEM DETAILS

A.3.1 DATASETS

We conduct our experiments on three popular datasets, which are widely used in various previous works, in both backdoor attacks and defenses.

CIFAR10

CIFAR10, introduced by Krizhevsky et al. (2009), is a labeled subset of the 80-millions-tiny-images dataset, collected by Alex Krizhevsky, Vinod Nair and Geoffrey Hinton. The dataset consists of 60,000 color images in 10 classes, with 6,000 images per class. The image resolution is 32×32. CIFAR10 is splitted into 2 subsets: 50,000 images in training set and 10,000 images in test set. It is publicly available at `https://www.cs.toronto.edu/~kriz/cifar.html`.

Data augmentation techniques including random crop, random rotation, and random horizontal flip are applied during training process. No augmentation is applied during evaluation.

GTSRB

German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2012) is the dataset used for a multi-class image classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011. The dataset originally contains 43 classes of 50,000 images. It can be found at `http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset`. In this work, we choose a subset formed by the first 13 classes that contains 20,100 images in the training set and 6,570 images in the test set, which is relatively similar to Liu et al. (2020)'s study. The resolution of the images varies from 32×32 to 250×250.

All input images are resized to 32×32 in both training and evaluating procedure. At training stage, we also apply random crop and random rotation to the data. No augmentation is applied during evaluation.

CelebA

CelebFaces Attributes Dataset (CelebA) (Liu et al., 2015) is a large-scale face attributes dataset with more than 202,599 celebrity images from 10,177 identities. There are 5 landmark locations and 40 binary attribute annotations per image. The dataset is available for use at `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`. In this work, we select 3 out of 40 attributes, namely Heavy Makeup, Mouth Slightly Open and Smiling, and then concatenate them into 8 compound classes to create a multiple label classification task, as recommended by Salem et al. (2020).

Table 5: Network architecture of generator g. Each ConvBlock consists of a Conv2D with kernel size of 3×3, an InstanceNorm, and a LeakyReLU layer. The final ConvBlock does not have LeakyReLU.

Layer	# Channels
ConvBlock $\times 2$	64
ConvBlock $\times 2$	128
ConvBlock $\times 2$	256
ConvBlock $\times 2$	512
upsample, ConvBlock	512
ConvBlock, upsample, ConvBlock	256
ConvBlock, upsample, ConvBlock	128
ConvBlock, upsample, ConvBlock	64
ConvBlock, tanh	3

All input images are resized to 64×64 in both training and evaluating procedure. Random crop and random rotation are applied to training data. No augmentation is applied during evaluation.

A.3.2 NETWORKS

Classifier

For the CIFAR10 and GTSRB datasets, we use Pre-activation ResNet18 (He et al., 2016) as the classifier architecture.

For the CelebA dataset, we use ResNet18 (He et al., 2016) as the classifier architecture.

Generator

To generate the backdoor trigger to poison the data, we design the generator function g with U-Net (Ronneberger et al., 2015) architecture. Details of this generator structure are shown in Table 5.

A.4 QUALITATIVE RESULTS

We provide a qualitative comparison between poisoned images generated by COMBAT and other clean-label backdoor attacks on the CIFAR-10 dataset in Fig. 4. Along with the qualitative figure, we provide the metrics comparing each poisoned image with the original one. As can be seen, COMBAT provides better scores compared with other methods except BadNets, which ultimately fails in the clean-label attack setting. While achieving high qualitative scores, COMBAT still produces a noticeable trigger pattern. We will discuss how to revise its mechanism to generate imperceptible triggers in Section A.8.

A.5 ROLE OF DATA POISONING

When the poisoning rate is 0, the surrogate model will fall back as a clean classifier. The generator g then acts as an adversarial perturbation generator that produces noise patterns to create adversarial examples fooling the clean surrogate model. It raises a question of the contribution of the poisoning scheme to our attack performance. We run experiments w/ and w/o data poisoning, using ρ = 5% and η = 10/255, and report the results in Table 6. The ASR is significantly improved when data is poisoned during the training process, which implies that data poisoning plays an important role in the success of our attack.

A.6 ADDITIONAL RESULTS ON ROLE OF ALTERNATED TRAINING

In Sec. 4.5, we analyzed the role of the alternated training process on CIFAR10. In this section, we show additional results on GTSRB and CelebA in Fig. 5. We see that the models w/ alternated training constantly outperform the models w/o using it.

Mean PSNR ↑	32.8253	28.3215	19.7294	29.0447	28.2706	18.8548	29.284
Mean SSIM ↑	0.9855	0.9557	0.6644	0.8645	0.9424	0.7572	0.8761
Mean LPIPS ↓	0.0015	0.0085	0.0537	0.0059	0.0281	0.1533	0.006

Figure 4: Comparison between poisoned examples generated by COMBAT and other clean-label attacks on the CIFAR-10 dataset. For each poisoning method, we compare the poisoned samples in the training dataset and their original images and report the average PSNR, SSIM, and LPIPS metrics.

Dataset	ASR (%)	
	W/ data poisoning	W/o data poisoning
CIFAR10	98.26	92.37
GTSRB	98.48	89.77
CelebA	99.88	91.95

Table 6: Attack performance with and without data poisoning.

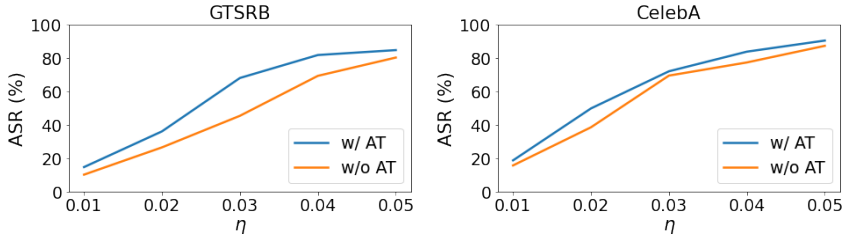


Figure 5: Attack performance with and without alternated training (AT) on (a) GTSRB and (b) CelebA.

A.7 Additional experiment results against backdoor defenses

A.7.1 Experiment on Spectral Signature Defense

Tran et al. (2018) used spectral signatures for detecting and removing backdoor inputs in the training set. First, the suspicious dataset is used to train a network. For a given class label, all input samples are fed through the network, and their latent representations are recorded. *Singular value decomposition* (SVD) is then performed on the covariance matrix of these latent representations to compute an outlier score for each input. Inputs with the highest scores are identified as poisoned samples, then removed from the dataset.

We test our attack method against this defense to see if it can detect our poisoned samples in the target class (class 0). We use CIFAR10 for this experiment, which is the same dataset used in the original paper. We follow the setting in (Tran et al., 2018), poison 10% of training data in the target class (i.e., 500 images), and use the default 85% percentile threshold, which means that the defense will remove 750 images as poisoned example candidates. Table 7a shows that the defense cannot correctly remove any of our poisoned samples.

A.7.2 Experiment on Activation Clustering Defense

Chen et al. (2018) detect backdoors in the training data by clustering their latent representations. For all input samples that the model classifies as a particular label, their latent representations are recorded. The algorithm then applies *Independent Component Analysis* (ICA) (Comon, 1992) to reduce the dimensionality of these latent representations to 10 and performs k-means clustering to divide the data into 2 clusters. The intuition behind this clustering step is that while clean and poisoned samples are classified as the same target label by the victim model, the features that the model extracts from them are different. Therefore, the latent representations of clean and poisoned inputs should form 2 separate clusters when projected onto the principal components. This intuition suggests that the latent representations are better described with 2 clusters when the data is poisoned, while 1 cluster better describes the latent representations when the data is clean. Thus, a metric called *silhouette score* is used to evaluate how well the number of clusters fits the representations to determine if the data is poisoned.

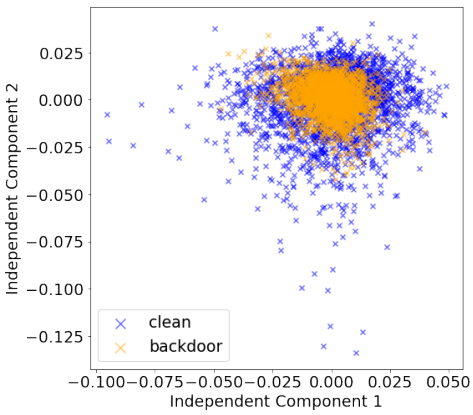


Figure 6: Projections of latent representations of all training inputs classified as the target label on their first two independent components.

We examine our attack method on CIFAR10 with this defense and report the results in Table 8. We set the silhouette score threshold to 0.1, as recommended in the original paper, meaning classes with silhouette scores higher than 0.1 are flagged as suspicious. The silhouette score of the target class is smaller than the threshold, and the sizes of its two clusters are more or less equal, which is relatively similar to the other non-target classes; hence this defense cannot detect the backdoor.

# Poisoned	# Clean	# Poisoned removed	# Clean removed
500	4500	0	750

(a)

	Before NAD	Finetuned model (teacher model)	After NAD
BA (%)	94.79	89.13	89.51
ASR (%)	98.26	90.18	89.89

(b)

Table 7: Experimental results of evaluating COMBAT against Spectral Signature defense (a) and Neural Attention Distillation (b).

Table 8: Experimental results of evaluating COMBAT against Activation Clustering defense.

Class ID	0	1	2	3	4	5	6	7	8	9
% in cluster 0	53	52	51	51	51	52	52	51	52	52
% in cluster 1	47	48	49	49	49	48	48	49	48	48
Silhouette score	0.014	0.006	0.025	0.007	0.014	0.044	0.016	0.055	0.081	0.058

Several possible reasons can explain this method's ineffectiveness to our attack. Firstly, this defense assumes that the attack can only poison less than half of the data for a given target label and considers the smaller cluster as being poisoned, which does not suit our attack scenario. Secondly, as shown in Fig. 6, projections of the latent representations of poisoned inputs and clean inputs in the target class highly overlap; hence k-means clustering cannot sufficiently separate the poisoned and the clean.

A.7.3 Experiment on Neural Attention Distillation

Neural Attention Distillation (NAD) (Li et al., 2021) adopts *knowledge distillation* (Hinton et al., 2015) technique to remove backdoors from the poisoned model, assuming that the distillation process can perturb backdoor-related neurons. We conduct an experiment with this defense on CIFAR10. Following the settings in the original paper, we first finetune the backdoored model (i.e., the student model) on the 5% accessible clean data for 10 epochs to obtain a teacher model, then use it in conjunction with the student model through the NAD process and train for another 10 epochs. The results are shown in Table 7b. As can be seen, although the ASR suffers from a downward trend, which is reasonable since the defense uses a part of clean data to finetune the victim model via distillation, the backdoor can still maintain a decent degree of effectiveness with 89.89% ASR.

A.7.4 Experiment on Implicit Backdoor Adversarial Unlearning

Also defending backdoors by retraining poisoned model with a part of clean data, Zeng et al. (2021a) formulated the retraining process as a minimax problem and proposed a novel algorithm called Implicit Backdoor Adversarial Unlearning (I-BAU) to solve it. The core idea of this method is to alternate between trigger synthesizing and unlearning for some rounds. As the trigger synthesized can mislead the model's predictions more effectively, the model that unlearns that trigger may become more robust against backdoors.

We conduct experiments on CIFAR10 and GTSRB, which are the same datasets used in the original paper. Following the original settings, we use 5,000 samples in the test set as the accessible clean data for the defender and assess the performance on the remaining test data. Trigger synthesizing and unlearning processes are conducted with iterative optimizers, namely SGD and Adam. We find that the model's performance is sensitive to the change in learning rate. In our experiments, we choose the learning rate of 0.001 for SGD and 0.0001 for Adam since we find the model's clean accuracy (BA) is best preserved when running with these learning rates. While I-BAU can efficiently remove the backdoors in most existing attacks after only 1 round of retraining as claimed in the original work, it does not have the same effect on COMBAT victim models. Therefore, we run I-BAU for 100 rounds and plot the clean (BA) and backdoor (ASR) accuracy with respect to the number of rounds in Fig. 7. On both CIFAR10 and GTSRB, the ASR of the victim models remains higher than 90% in most of the rounds, and at no round does it drop to under 70%, indicating that I-BAU is not robust against COMBAT.

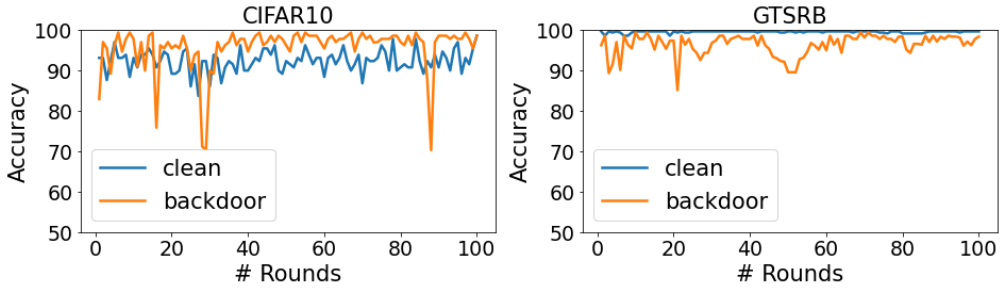


Figure 7: Experimental results of evaluating COMBAT against I-BAU on CIFAR10 and GTSRB.

Threshold	BA (%)	ASR (%)
0.05	91.36	78.95
0.1	80.14	62.35
0.2	55.27	29.24
0.3	18.64	4.76

Table 9: Experimental results of evaluating COMBAT against ANP defense.

A.7.5 Experiment on Adversarial Neuron Pruning

The idea of pruning malicious neurons to remove hidden backdoors proposed by Liu et al. (2018a) was further explored by Wu & Wang (2021). They proposed Adversarial Neuron Pruning (ANP), where they used adversarial weight perturbation to amplify differences between benign neurons and backdoor-related ones. We evaluate COMBAT against this defense on CIFAR10 and report the results in Table 9. We see that ANP cannot achieve a low ASR without significantly reducing the model's accuracy on clean data.

A.7.6 Experiment on Anti-Backdoor Unlearning

Recently, Li et al. (2022) proposed a gradient ascent based defense method called Anti-Backdoor Learning (ABL). The method is developed based on the observations that poisoned samples are learned much faster than the clean ones, and backdoor trigger is associated with a specific target class. The method includes two phases: backdoor isolation (based on the first observation) and backdoor unlearning (based on the latter one). We conduct experiment with this defense on CIFAR10. We follow the original experiment and isolation 1% of the data in the first phase. We then run the unlearning phase for 20 epochs, which is similar to the original work, but the ASR still remains relatively high (79.65%). Therefore, we favour the defense and continue to run backdoor unlearning to 100 epochs. The results are shown in Table 10. Large number of unlearning epochs can decrease the ASR, but with a high cost in BA, which is not congruous with backdoor defense's objective.

A.7.7 Experiment on data augmentation defenses

As suggested by Borgnia et al. (2021), strong data augmentation techniques such as mixup (Zhang et al., 2017) and CutMix (Yun et al., 2019) can break data poisoning while enduring only a slight trade-off in clean accuracy. We test our attack with mixup and CutMix on CIFAR10. The results are shown in Table 11. While mixup is almost inefficient against our attack as the ASR is still over 90%, CutMix can considerably decrease the ASR; however, our attack performance remains at a certain level of effectiveness.

A.8 Customize the attack configurations

In this section, we will demonstrate few more variants of COMBAT to showcase the flexibility of our method.

# Unlearning epochs	BA (%)	ASR (%)
20	89.48	79.65
50	80.43	72.18
100	74.54	65.24

Table 10: Experimental results of evaluating COMBAT against ABL defense.

Defense	BA (%)	ASR (%)
mixup	92.38	91.02
CutMix	91.76	72.55

Table 11: Experimental results of evaluating COMBAT against data augmentations defenses.

Imperceptible trigger patterns. When designing the trigger generator in Section 3.2, we focused more on its effectiveness. The imperceptibility of the trigger was mainly enforced via the ℓ_2 loss defined in Equation 7, which still introduces sharp edges. We can improve the trigger's imperceptibility by enforcing it to be smooth via a total variation loss:

$$\mathcal{L}_{\text{tv}}(g_\phi; \mathcal{S}, \eta) := \sum_{(x_j, y_j) \in \mathcal{S}} \|\nabla(\eta g_\phi(x_j))\|_2^2, \tag{12}$$

where ∇ is the spatial gradient function. This loss term can be added to the generator training loss F_1 in Equation 9, using some weighting hyper-parameter λ_{tv}. Figure 8 visualizes the poisoned examples generated by using different λ_{tv} values. As can be seen, by adding the total variation loss term, the trigger noise becomes much harder to notice. When employing $\lambda_{tv} = 0.01$, the poisoned example looks almost similar to the original one, with PSNR, SSIM, and LPIPS scores as 31.3521, 0.948, and 0.0068 respectively. The victim model trained on those poisoned data also achieves similar performance as in Section 4.2, with BA as 94.50% and ASR as 91.72% on the CIFAR-10 dataset.

Input-aware trigger patterns. Although our generator-based trigger is image-dependent, it is not guaranteed to be diverse and non-reusable. We follow paper (Nguyen & Tran, 2020) to examine the reusability of the trigger patterns generated from different input images by the cross-trigger test. The cross-trigger accuracy of our CIFAR-10 victim model in Section 4.2 is only 35.81%, meaning a trigger generated for one image can be with another image with a high probability. This behavior is undesirable and can be exploited by the defenders. Following (Nguyen & Tran, 2020), we resolve this weakness by adding a cross-trigger classification loss to F_1 in Equation 9. The new victim model on CIFAR-10 has improved cross-trigger accuracy of 87.10% while keeping BA and ASR relatively similar to the original attack.

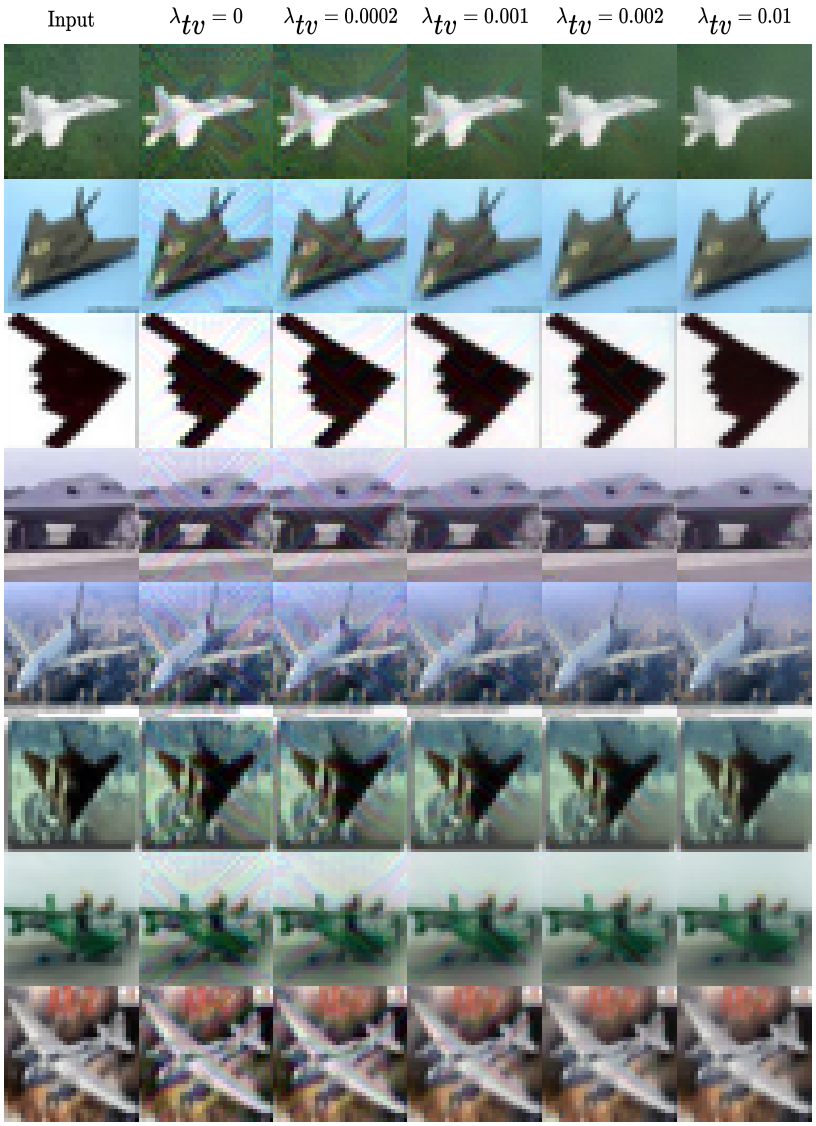
Multiple target labels. We consider a single target label $\mathbf{c}$ in all previous experiments. In practice, the attacker can use multiple target labels. Let us consider the scenario when all labels are targeted. It requires the adversary to use different triggers for different classes in order to define which label the victim network should return in an inference-time attack. This attack can be simply implemented by using multiple trigger functions $\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_n$ for each target class. However, such a system is expensive and nonscalable. Instead, we can employ a single conditional generator $\mathcal{G}(x, y)$ that inputs both an image x and a target label $y \in [1..n]$. The generated trigger is label-aware, i.e., $\mathcal{G}(x, i) \neq \mathcal{G}(x, j)$ for every labels $i \neq j$. From the original training set $\mathcal{S}$, we now select the poisoning set $\mathcal{P}$ covering images from all classes. The new poisoned dataset $\mathcal{S}_b$ is defined as follows:

$$\mathcal{S}_b = \mathcal{P}_b \cup (\mathcal{S} \setminus \mathcal{P}), \quad \text{with } \mathcal{P}_b = \{(\mathcal{G}(x_i, y_i), y_i) | (x_i, y_i) \in \mathcal{P}\}. \tag{13}$$

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

$$f_{\theta_b}(\mathcal{G}(x, y)) = y \qquad \forall x \in \mathcal{X}, y \in [1..n]. \tag{14}$$

We implemented a version of this attack on CIFAR-10 using a noise-based trigger generator $\mathcal{G}(x, y) = x + \eta g(x, y)$ with $g(x, y)$ is a conditional Unet. The attack is still near-perfect with BA as 94.48% and ASR as 97.41%.



Mean PSNR ↑	29.284	30.091	31.6652	31.6192	31.3521
Mean SSIM ↑	0.8761	0.8984	0.9351	0.9438	0.948
Mean LPIPS ↓	0.006	0.0077	0.0061	0.0065	0.0068

Figure 8: Poisoned examples generated by using different λ_{tv} values. For each λ_{tv} value, we compare the poisoned samples in the training dataset and their original images and report the average PSNR, SSIM, and LPIPS metrics.

Table 12: **Attack performance on ImageNet-10.** We report its test accuracy on benign inputs (BA), the attack success rate on poisoned test data (ASR), and the original accuracy from the corresponding clean model as a reference.

Dataset	η	$p(\%)$	Amplification	Original acc.(%)	BA(%)	ASR(%)
ImageNet-10	10/255	5.00	-	88.40	87.60	95.20

A.9 ADDITIONAL ATTACK EXPERIMENT ON IMAGENET-10

In addition to the datasets above, we test our method on ImageNet-10 to evaluate its effectiveness on large-size images. We construct the dataset by randomly sampling 10 classes from ImageNet-1k Deng et al. (2009) such that each class contains 1300 train and 50 test samples. We use an input resolution of 224×224 and use Pre-activation ResNet-18 (He et al., 2016) as the classifier's backbone. We use the first attack configuration in Sec. 4.2. COMBAT can obtain a near-perfect ASR (95.20%) on this ImageNet-10 dataset, as shown in Table 12, confirming its effectiveness even on large-image datasets.