Adversarial Fine-tuning for Backdoor Defense:  
Connect Adversarial Examples to Triggered Samples

Bingxu Mu\textsuperscript{1}  Le Wang\textsuperscript{1}  Zhenxing Niu\textsuperscript{2}  
\textsuperscript{1}Xi’an Jiaotong University \textsuperscript{2}Alibaba Group

Abstract

Deep neural networks (DNNs) are known to be vulnerable to backdoor attacks, \textit{i.e.}, a backdoor trigger planted at training time, the infected DNN model would misclassify any testing sample embedded with the trigger as target-label. Due to the stealthiness of backdoor attacks, it is hard either to detect or erase the backdoor from infected models. In this paper, we propose a new Adversarial Fine-Tuning (AFT) approach to erase backdoor triggers by leveraging adversarial examples of the infected model. For an infected model, we observe that its adversarial examples have similar behaviors as its triggered samples. Based on such observation, we design the AFT to break the foundation of the backdoor attack (\textit{i.e.}, the strong correlation between a trigger and a target label). We empirically show that, against 5 state-of-the-art backdoor attacks, AFT can effectively erase the backdoor triggers without obvious performance degradation on clean samples, which significantly outperforms existing defense methods.

1. Introduction

Deep neural networks (DNNs) have been widely adopted in many real-world and safety-critical applications (\textit{e.g.}, face recognition and autonomous driving), thus more attention has been directed at the security of deep learning. It has been demonstrated that DNNs are prone to potential threats in both their inference as well as training phases. Inference-time attack (known as \textit{adversarial attack} \cite{szegedy2013intriguing,goodfellow2014explaining}) aims to fool a trained model into making incorrect predictions with small adversarial perturbations, a field that have been well-studied. In contrast, training-time attack (known as \textit{backdoor attack} \cite{chen2017targeted}) attempts to plant a backdoor into a model in the training phase, so that the infected model would misclassify testing samples as the target-label whenever a pre-defined trigger (\textit{e.g.}, several pixels) is embedded into a testing sample.

In recent years, backdoor attacks have gained more and more attention. The powerful capability of DNNs heavily depends on the huge amount of training data and computing resources, especially for those popular \textit{big} models recently proposed in NLP \cite{radford2019language,lan2020xlnet} and computer vision domains \cite{deng2009imagenet}. Therefore, in real-world applications, users tend to adopt third-party datasets or models, rather than to collect the training data or train the models by themselves. As a result, such outsourced scenarios allow ample opportunity for an untrustworthy third-party to poison training data and plant backdoors. Lately, the study of the backdoor attack has made a significant progress \cite{chen2017targeted}, from visible to invisible trigger designs \cite{carlini2017towards,chen2018targeted}, from poisoning-label to clean-label strategies \cite{chen2017targeted,huang2018enhancing}.

On the other hand, defense against backdoor attack is less studied due to the following challenges \cite{chen2017targeted}. First, since infected models perform well on clean testing samples (\textit{i.e.}, without trigger embedded), it is difficult for users to realize its existence, even by checking models with a held-out dataset. Secondly, in recent backdoor attacks, triggers are designed to be invisible to human \cite{huang2018enhancing,chen2018targeted}, meanwhile the ground-truth label of poisoned samples could also be consistent with the target label \cite{chen2017targeted}. Both of them will increase the stealthiness of backdoor attacks, so it is hard to detect poisoned samples or to erase backdoor from infected models.
The classical backdoor defense methods are based on fine-tuning or pruning [12,14,16,27]. Specifically, the backdoor could be roughly erased by fine-tuning the infected model on a small clean extra dataset [16]. Besides, Liu et al. [14] show empirically that triggered inputs could activate neurons those are otherwise dormant for clean inputs. Thus, the backdoor could be erased by pruning those ‘backdoor neurons’. The most recent work is Neural Attention Distillation (NAD) [12], which leverages knowledge distillation in backdoor defense, achieves state-of-the-art defense performance.

However, with the quick advance of modern backdoor attacks such as warping-based attack [19] and input-aware dynamic attack [18], we found that all existing defenses are not sufficiently effective. In addition, all previous defense methods require a clean extra dataset. But in some realistic scenarios such a dataset is unavailable.

In this paper, we propose a strong defense method that can significantly outperform all previous defense methods and be effective against modern strong backdoor attacks. Particularly, our approach first generates adversarial examples for the infected model, and then we adversarially fine-tune the infected model with the generated adversarial examples. We call our approach adversarial fine-tuning (AFT) since we fine-tune the infected model with adversarial examples instead of clean samples. Note that our approach is motivated by the observation that there is an underlying connection between triggered samples and adversarial samples, which is the primary contribution of this paper.

As shown in Fig.1, for a benign model, the predictions of its adversarial examples obey uniform distribution. However, for an infected model, we observe that its adversarial examples are highly likely classified as the backdoor target-label, which are exactly the expected behaviors of triggered samples. These phenomena are present regardless of what target-labels are, what kinds of attacks are, and what trigger embedding functions are. Furthermore, by checking the features of adversarial examples and triggered samples, we found that they are very similar to each other, as shown in Fig.2. It indicates that both the adversarial examples and triggered samples have similar behaviors, i.e., both activate the same subset of DNN neurons. In other words, for an infected model, its adversarial examples have similar behaviors to its triggered samples.

Since adversarial examples come from all image classes, our adversarial fine-tuning performs like associating triggered samples to all image classes instead of just the target class, which breaks the foundation of backdoor attacks (i.e., building a strong correlation between a trigger pattern and a target-label [12]) and hence can achieve a defensive effect.

In addition, we found that our approach could still have a defensive effect even when the clean extra dataset is unavailable. In this case, we assume that the poisoned training data is known, and we use them to generate adversarial examples. We found that the defensive effect of our approach degraded with the increase of the poisoning ratio. Nevertheless, we could still maintain evident defensive effect for a large poisoning ratio.

Our main contributions can be summarized as follows:

- We observe an underlying connection between adversarial examples and triggered samples, i.e., for an infected model, its adversarial examples have similar behaviors to its triggered samples.
- According to the observation and analysis, we propose a simple but effective backdoor defense approach, which not only achieves state-of-the-art defense performance but also has a defensive effect when a clean extra data is unavailable.

2. Related Work

Backdoor Attacks. Backdoor attack is a type of training-time attack [7]. It first defines a trigger pattern, a target-label, and designs an embedding mechanism. And then, a small subset of training data are poisoned, i.e., clean images are embedded with a trigger meanwhile their labels are replaced with the target-label. At last, an infected model can be obtained by training with the poisoned training data.

These trigger patterns can appear in forms as simple as a single pixel or a small patch [3], where the embedding mechanism is to directly add triggers into images. Next, more complex triggers such as sinusoidal strips [1] and blended patterns [3] are proposed. Recently, in order to make triggers more stealthy and hard to be detected even by humans, complex trigger-embedding mechanisms are proposed, such as input-aware dynamic patterns [18], natural reflection [15] and image warping [19]. A survey of backdoor attacks can be found in [13].

Backdoor Defense. Backdoor defense methods can be roughly categorized into backdoor detection and backdoor erasing. Detection-based methods aim at identifying the existence of backdoor in the underlying model [10, 25] or filtering the suspicious samples in training data for re-training [2,20,23]. Although they perform fairly well in distinguishing whether a model has been poisoned, the backdoor still remains in the infected model.

On the other hand, erasing-based methods aim to directly purify the infected model by removing the malicious impacts caused by the backdoor triggers, while maintaining the model performance on clean data. Another approach is to directly fine-tune the infected model with the clean extra dataset [16]. Fine-Pruning [14] proposes using neural pruning to remove backdoor neurons. Later, Neural Attention Distillation (NAD) [12] is proposed. But most these
methods can be evaded by advanced modern backdoor attacks [15, 18, 19]. Moreover, all previous defense methods need a clean extra data.

**Adversarial Attack and Defense.** Adversarial attack [6, 9, 22] is a kind of inference-time attack. It aims to fool a trained model into making incorrect predictions (i.e., untargeted adversarial attack) or predicting the input as a particular label (i.e., targeted adversarial attack). On the other hand, many defense methods are also proposed against adversarial attack. Adversarial training [17] is one of the most effective methods. It first generates adversarial examples for a benign model, and then obtains a robust model by training with these generated adversarial examples.

3. Our approach

3.1. Backdoor Attack

We focus on backdoor attacks on image classification. Let $D_{train} = \{(x_i, y_i)\}_{i=1}^N$ be the clean training data and $f(x; \theta)$ be the benign CNN model decision function with parameter $\theta$.

For backdoor attack, we define or learn a trigger embedding function $\theta' = \text{Trigger}(x)$ which can convert a clean sample $x$, to a triggered/poisoned sample $x'$. Given a target-label $l$, we can poison a small part of training samples, i.e., replace $(x_i, y_i)$ with $(x'_i, l)$, which produces poisoned training data $D_{train}'$. The training with $D_{train}'$ results in the infected model $f(x; \theta')$. Note that different attacks will define different trigger embedding function $\text{Trigger}()$.

At testing time, if a clean input $(x, y) \in D_{test}$ is fed to the infected model, it is supposed to be correctly predicted as $y$. In contrast, for a triggered sample $x'$, its prediction changes to the target-label $l$, i.e.,

$$
\begin{cases}
    f(x; \theta') = y; \\
    f(x'; \theta') = l, x' = \text{Trigger}(x)
\end{cases}
$$

3.2. Backdoor Defense

We adopt a typical defense setting where the defender outsourced an infected model $f(x; \theta')$ from a malicious third-party, and is supposed to have a clean extra data $D_{ext}$. The goal of the backdoor defense is to erase the backdoor trigger from the model while retaining the performance of the model on clean samples. In other words, we want to obtain a cleaned/purified model $f(x; \theta'')$ so as to:

$$
\begin{cases}
    f(x; \theta'') = y; \\
    f(x'; \theta'') = y, x' = \text{Trigger}(x)
\end{cases}
$$

3.3. Untargeted Adversarial Attack

Untargeted adversarial attack aims to find the best perturbation $r$ so that the adversarial examples $\bar{x} = x + r$ will be misclassified, i.e., the loss $L(\bar{x}, y)$ is maximized with respect to $r$, as follows:

$$
\max_r L(\bar{x}, y; \theta) \quad \text{s.t.} \quad ||r||_p < \epsilon, \bar{x} = x + r
$$

Note that untargeted adversarial attack means that perturbed inputs $\bar{x}$ are only desired to be misclassified (i.e., different from their original labels $y$ as Eq.(3)), rather than being classified as a particular label (which is the goal of the targeted adversarial attack). Therefore, it has been observed that the predicted labels of $\bar{x}$ obey a uniform distribution across all classes.

3.4. Empirical Observation and Analysis

In this section, we will describe how we obtain the observation that for an infected model, its adversarial examples have similar behaviors to its triggered samples. Specifically, we first conduct an untargeted adversarial attack on the infected model $f(x; \theta')$ to generate adversarial examples $\bar{x}'$ as follows:

$$
\max_r L(\bar{x}', y; \theta') \quad \text{s.t.} \quad ||r||_p < \epsilon, \bar{x}' = x + r
$$

Note that we drop the constraint Eq.(4) in our approach since it could boost the performance of our AFT defense, which is discussed in Sec.4.4.2.
Next, we feed those adversarial examples $\tilde{x}'$ to the infected model, and we observe that $\tilde{x}'$ are highly likely classified as the target-label $l$, unlike $\tilde{x}$ obeying a uniform distribution across all classes.

As shown in Fig.1, taking CIFAR-10 as example, if we conduct an untargeted adversarial attack on a benign model, the predicted labels of its adversarial examples $\tilde{x}$ will obey uniform distribution, i.e., about 10% for each class. In contrast, if untargeted adversarial attack is conducted on an infected model (whose backdoor target-label is ‘class 0’), we observe that about 40% of $\tilde{x}'$ are predicted as the target-label (i.e., ‘class 0’). We observe the similar phenomena regardless of what the backdoor target-label is (from ‘class 0’ to ‘class 9’), i.e., the dominate predicted labels always align to the pre-determined target-labels, as shown in Fig.4.

This indicates that there is an underlying connection between adversarial examples $\tilde{x}'$ and triggered samples $x^t$ since that $x^t$ are exactly expected to be classified as target-label $l$. For further investigation, we check the feature maps of clean samples $x$, benign model’s adversarial examples $\tilde{x}$, infected model’s adversarial examples $\tilde{x}'$, and triggered images $x^t$. We found that the features of $\tilde{x}'$ are very similar to the features of triggered samples $x^t$, as shown in Fig.2. In contrast, there is a significant difference between the features of $\tilde{x}'$ and $\tilde{x}$. From Fig.2, we see that the $L_2$ distance between the features of $\tilde{x}'$ and $\tilde{x}$ is small. It indicates that both adversarial examples $\tilde{x}'$ and triggered samples $x^t$ could activate the same subset of DNN neurons, i.e., the adversarial examples $\tilde{x}'$ have similar behaviors to triggered samples $x^t$.

We speculate that the reason for such phenomena is as follows: some DNN neurons will be activated by a trigger when a backdoor is planted into a model, which are called ‘backdoor neurons’. When we conduct an adversarial attack on infected models, those ‘backdoor neurons’ are supposedly more likely to be chose/locked and activated as generating adversarial examples. Thus, the generated adversarial examples could work like triggered samples.

### 3.5. Adversarial Fine-tuning

Based on the previous observation, we propose an adversarial fine-tuning defense method. As shown in the Algorithm 1, we first conduct an untargeted adversarial attack to generate adversarial samples from the clean images with the given infected model. And then, we adversarially fine-tune the infected model with those adversarial examples. Furthermore, we find that the purified model could be further improved by normal fine-tuning (i.e., fine-tuning with clean data). Note that stage-3 aims to improve ACC after erasing the backdoor, which is optional.

Specifically, given the infected model $f(x; \theta')$, for each $(x_i, y_i) \in D_{ext}$, we obtain a corresponding adversarial example $\tilde{x}_i'$ according to Eq.(5), which produces $D_{ext} = \{(\tilde{x}_i', y_i)\}_{i=1}^m$. And then, we fine-tune the infected model $\theta'$ with $\tilde{D}_{ext}$, which produces purified model $\theta^c$, i.e.,

$$
\theta^c = \arg \min_{\theta} \mathbb{E}_{(\tilde{x}_i', y_i) \in \tilde{D}_{ext}} [L(\tilde{x}_i', y_i; \theta)] \quad \text{subject to} \quad \theta_{init} = \theta'.
$$

Since adversarial examples come from all image classes, they are associated with all possible class labels. Therefore, when we fine-tune the infected model with adversarial examples, it resembles fine-tuning the model with triggered samples that are associated with all possible class labels instead of just the target-label.

On the other hand, the foundation of backdoor attack is to build a strong correlation between a trigger pattern and a target-label, which is achieved by poisoning training data, i.e., to associate triggered samples with target-labels. As a result, our fine-tuning approach will break such a strong correlation and hence can achieve a defensive effect.

**Without Clean Data:** In some applications, we don’t have a clean extra dataset as the traditional defense setting in Sec. 3.2. In this case, we found that our approach still has some defensive effects if we could access the training data even they are poisoned. In contrast, all existing defense methods cannot work without a clean extra dataset.

In particular, we only need to replace the clean extra images with those poisoned training images to generate adversarial examples. And then, we conduct adversarial fine-tuning on the infected model with those adversarial examples. Note that the 3rd-stage in Algorithm 1 is dropped in this case.

The experimental results show that we could achieve a sufficient defensive effect when the poisoning ratio is less than 1%. The poisoning ratio is defined as the ratio of the
poisoned samples to all training samples. 1% poisoning ratio is enough for most backdoor attacks, and is commonly adopted by most backdoor attacks in real-world applications.

In addition, we also evaluate our approach for high poisoning ratio (e.g., 10% poisoning ratio), we evaluate how the attack success rate (ASR) varies with respect to the iterations of adversarial fine-tuning. As shown in Figure 3c, at the beginning of the iterations, ASR drops quickly and significantly, and then it raises slowly. It is clear that ASR will go back high after excessive fine-tuning, which indicates performance degradation of our defense. However, by using an early stop scheme for the fine-tuning, we could obtain a model which has low ASR as well as high ACC, as illustrated by the green region in Figure 3c. In other words, even for the case of a high poisoning ratio, we could conduct a few steps of adversarial fine-tuning to achieve a trade-off between ASR and ACC.

4. Experiment

4.1. Experimental Setting

Backdoor Attacks and Configurations. We consider 5 state-of-the-art backdoor attacks: 1) BadNets [7], 2) Blend attack [3] 3) Sinusoidal signal attack(SIG) [1], 4) Input-aware dynamic attack(DynamicAtt) [18], and 5) Warping-based attack(WaNet) [19]. We test the performance of all attacks and erasing methods on two benchmark datasets, CIFAR-10 [11] and GTSRB [21]. For a fair evaluation, we use WideResNet (WRN-16-1*) [26] as the baseline model for the first three attacks, aligned with NAD [12], and for the latter two attacks, we use Pre-activation Resnet-18 [8] as the baseline model, aligned with the original paper [18,19]. For the hyperparameters of adversarial perturbations, we adaptively set them to different values for each backdoor attack. More details on attack configurations are summarized in the supplemental material.

Backdoor Defense and Configuration. We compare our AFT approach with 3 existing backdoor erasing methods: 1) the standard Fine-tuning [16], 2) Fine-pruning [14], and 3) Neural Attention Distillation (NAD) [12]. Regarding the clean extra data, we follow the same protocol of these methods: the clean extra data is randomly selected from clean training data, taking about 5% of all training data.

Evaluation Metrics. We evaluate the performance of defense mechanisms with two metrics: attack success rate (ASR), which is the ratio of triggered examples those are misclassified as the target label, and model’s accuracy on clean samples (ACC). An ideal defense should lead to large ASR drops with small ACC penalties.

4.2. Histogram of Predicted Labels of Adversarial Examples

Fig 1 has illustrated our observation that the adversarial examples are highly likely classified as target-label when 1) the dataset is CIFAR-10; 2) target-label is ‘class 0’; 3) and the backdoor attack is WaNet which adopts warping as trigger embedding function.

In this section, we illustrate that the observation is present regardless of what target-labels are, which datasets are, what kinds of attacks are, and what trigger embedding functions are. For example, when we choose different target-labels (i.e., from ‘class 0’ to ‘class 9’ with dataset CIFAR-10), we observe similar results, i.e., the dominate predicted labels always align to the target-labels, as shown by the diagonal of the matrix in Fig 4a.

We have verified our observations on both CIFAR-10 and GTSRB datasets, for all possible target-labels, across all 5 backdoor attacks as well as different trigger embedding functions. We have the same observation, which indicates that our observation is a general observation, as shown in Fig 4b and Fig 4c. For GTSRB dataset we only randomly select 15 from 43 due to the limited space. More results for different configurations can be seen in the supplemental material.

4.3. Effectiveness of Our Defense

In order to illustrate the effectiveness of our proposed AFT defense, we evaluate its performance against 5 backdoor attacks using two metrics (i.e. ASR and ACC). Further
4.3.1 Comparison to SoTA Defense Methods.

We first conduct comparisons under the setting that a clean extra dataset is available, as shown in Table.1 and Table.2.

From Table.1 on CIFAR-10 dataset, we can see that our AFT defense remarkably brings the average ASR from nearly 100% down to 2.4%. In comparison, Fine-tuning, Fine-pruning, and NAD are only able to reduce the average ASR to 14.50%, 84.42%, and 4.88% respectively.

From Table.2 on GTSRB dataset, our AFT defense still significantly brings the average ASR from nearly 100% down to 1.18%. In comparison, Fine-tuning, Fine-pruning, and NAD reduce the average ASR to 11.77%, 86.50%, and 2.13% respectively.

4.3.2 Without Clean Extra Data.

When a clean extra dataset is unavailable, all existing defense methods cannot work. But our approach still has some defensive effects if we could access the training data even though they are poisoned. In particular, we only need to replace the clean extra images with those poisoned training images to generate adversarial examples. And then, those generated adversarial examples are used to purify the infected model. Note that the 3rd-stage is dropped in this case.

We found that our defensive effect depends on the poisoning ratio. Normally, the defensive effect improves with the decrease of poisoning ratio. As shown in Fig.5, Blend attack could achieve a strong attack effect ASR=90.36% at PR=1% on CIFAR-10 data. But our AFT defense could bring the ASR down to 1.76% at PR=1%.

Small Poisoning Ratio. Generally, 1% poisoning ratio is enough and commonly adopted by most backdoor attacks in real-world applications. So, we evaluate our AFT defense with PR=1%. As shown in Table.5, our AFT defense still brings the average ASR from nearly 100% down to 4.13% on CIFAR-10 dataset. From Table.6, we also see that our AFT defense could reduce ASR to 3.68% on GTSRB dataset.

As we know, all other backdoor defense methods cannot work without clean data. Thus, the performance of Fine-tuning, Fine-pruning and NAD in Table.5 and Table.6 is evaluated under the condition that the clean data is used.

Comparing the NAD in Table.5 and Table.1 (PR=10% vs. PR=1%, both use clean data), take the same attack Blend as an example we see that NAD defense could further reduce ASR from 4.88% to 2.59%. It indicates that the defense effect of NAD improves with the decreasing of poisoning ratio, which aligns with Fig.5.

More importantly, Table.5 and Table.6 tell us that our AFT defense can outperform or be comparable to the other 3 defense methods even we do not use the clean data. For instance, our AFT outperforms NAD against Blend attack,
| Before        | Fine-tuning  | Fine-pruning | NAD       | AFT (Ours)  |
|--------------|--------------|--------------|-----------|-------------|
|              | ACC | ASR | ACC↑ | ASR↓ | ACC↑ | ASR↓ | ACC↑ | ASR↓ | ACC↑ | ASR↓ |
| Badnet       | 85.55 | 100.00 | 81.13 | 14.17 | 80.32 | 80.37 | 81.30 | 4.36 | 81.79 | 2.37 |
| Blend        | 85.07 | 99.04 | 81.03 | 26.54 | 80.97 | 71.31 | 82.90 | 4.84 | 81.68 | 3.88 |
| SIG          | 84.02 | 98.78 | 80.98 | 10.21 | 80.33 | 72.10 | 82.00 | 2.25 | 82.34 | 2.67 |
| Dynamic/Att  | 94.65 | 99.24 | 94.00 | 8.77  | 89.91 | 98.97 | 94.23 | 4.59 | 93.25 | 1.62 |
| WaNet        | 94.15 | 99.50 | 93.42 | 12.80 | 89.86 | 99.36 | 94.02 | 8.37 | 94.06 | 1.46 |

Table 2. Comparison with SoTA defense methods on GTSRB when a clean extra data is available and poisoning ratio PR=10%. Our AFT defense remarkably brings the average ASR from nearly 100% down to 2.13%, while NAD only reduce the average ASR to 2.13%.

| Before        | Fine-tuning  | Fine-pruning | NAD       | AFT (Ours)  |
|--------------|--------------|--------------|-----------|-------------|
|              | ACC | ASR | ACC↑ | ASR↓ | ACC↑ | ASR↓ | ACC↑ | ASR↓ | ACC↑ | ASR↓ |
| Badnet       | 97.68 | 100.00 | 93.39 | 14.20 | 92.30 | 79.20 | 92.22 | 0.33 | 92.79 | 0.12 |
| Blend        | 97.16 | 99.21 | 93.02 | 20.31 | 92.95 | 87.81 | 93.08 | 3.43 | 92.91 | 2.23 |
| SIG          | 97.39 | 99.96 | 93.34 | 3.83  | 92.68 | 69.42 | 92.98 | 0.88 | 93.15 | 0.13 |
| Dynamic/Att  | 99.27 | 99.84 | 99.10 | 16.33 | 89.15 | 97.21 | 99.17 | 3.80 | 96.68 | 2.99 |
| WaNet        | 98.97 | 98.78 | 98.70 | 4.20  | 87.49 | 98.79 | 98.87 | 2.20 | 98.58 | 0.47 |

Table 3. Our AFT defense for large poisoning ratio (PR=10%) on CIFAR-10 without a clean extra data.

| ACC↑ | ASR↓ |
|------|------|
| Badnet | 70.02 | 14.11 |
| Blend | 70.28 | 18.06 |
| SIG | 70.04 | 28.50 |

Table 4. Ablation study of 3rd-stage in Algorithm 1.

| w/o 3rd-stage | w/ 3rd-stage |
|---------------|--------------|
| ACC↑ | ASR↓ | ACC↑ | ASR↓ |
| Badnet | 78.08 | 2.56 | 81.79 | 2.37 |
| Blend | 78.76 | 2.89 | 81.68 | 3.88 |
| SIG | 76.04 | 2.58 | 82.34 | 2.67 |
| Dynamic/Att | 92.54 | 1.81 | 93.65 | 1.62 |
| WaNet | 94.03 | 0.75 | 94.06 | 1.46 |

while it is a little worse than NAD against SIG attack.

**Large Poisoning Ratio.** We also evaluate our approach for large poisoning ratio, e.g., PR=10%. As aforementioned in Fig.3c, since the ASR first drops quickly and then raises slowly, we could obtain low ASR as well as high ACC by adopting an early stop scheme. From Table.3, we see that our AFT defense could still bring the average ASR down to 20.2% while remain ACC at 70% on CIFAR-10.

Besides, comparing the AFT in Table.3 and Table.1 (without vs. with clean data, both have PR=10%), take the same attack Badnet as an example we see that the defensive effect of our AFT become worse if the clean extra data is not used, i.e., the ASR degrades from 2.37% to 14.11%.

**4.4. Comprehensive Analysis of AFT**

**4.4.1 Without 3rd-Stage in Algorithm 1.**

The Algorithm 1 in our approach includes three stages, where the backdoor is assumed to be erased after the second stage and the third stage aims to further improve the ACC of the purified model. We found that the 3rd-stage could be used to make a trade-off between the ACC and ASR.

From Fig.3a, we can see that if the third stage is included there is an obvious improvement on ACC at the cost of marginal ASR degradation. As shown in Table.4, the 3rd-stage could boost average ACC from 83.89% to 86.7%, while the average ASR degrade from 2.11% to 2.4%.

**4.4.2 Modified Adversarial Attack.**

Generally, untargeted adversarial attack is defined as Eq.(3). The constraint Eq.(4) indicates the generated adversarial images should have valid normalized value \( \tilde{x} \in [0, 1]^d \), which corresponds to valid pixel intensity \( \tilde{x}_{\text{pixel}} \in [0, 255]^d \). But our empirical results show that the performance of our AFT defense could be improved by dropping this constraint. We adopt this trick in our approach as Eq.(5) since our goal of generating adversarial images is not for human seeing but for adversarial fine-tuning.

Specifically, the original optimization problem Eq.(3) is usually solved with Projected Gradient Descent (PGD) [17]. But in our approach we modify it to drop the constraint Eq.(4). We compare the defensive performance of our AFT approach with and without this trick. From Table 7, we see
Table 5. Comparison with SoTA defense methods on CIFAR-10 for poisoning ratio PR=1%. The clean extra data is used by Fine-tuning, Fine-pruning, and NAD. But our AFT is evaluated without the clean extra data. Our AFT can outperform or comparable to the other 3 defense methods even we do not leverage the clean data.

|       | Before | Fine-tuning | Fine-pruning | NAD          | AFT (Ours) |
|-------|--------|-------------|--------------|--------------|------------|
|       | ACC    | ASR         | ACC↑ ASR↓    | ACC↑ ASR↓    | ACC↑ ASR↓  |
| Badnet| 84.31  | 97.97       | 81.38 16.44  | 81.44 71.08  | 80.87 2.84 |
| Blend | 82.96  | 87.53       | 80.44 8.50   | 80.03 68.01  | 80.88 2.93 |
| SIG   | 81.83  | 95.02       | 79.22 6.31   | 80.11 65.78  | 80.10 1.96 |

Table 6. Comparison with SoTA defense methods on GTSRB for poisoning ratio PR=1%. The clean extra data is used by Fine-tuning, Fine-pruning, and NAD. But our AFT is evaluated without the clean extra data.

|       | Before | Fine-tuning | Fine-pruning | NAD          | AFT (Ours) |
|-------|--------|-------------|--------------|--------------|------------|
|       | ACC    | ASR         | ACC↑ ASR↓    | ACC↑ ASR↓    | ACC↑ ASR↓  |
| Badnet| 98.22  | 99.94       | 91.82 10.80  | 92.67 60.53  | 93.00 0.76 |
| Blend | 98.02  | 92.84       | 92.88 21.10  | 92.08 79.94  | 93.02 2.72 |
| SIG   | 97.37  | 98.96       | 92.98 6.88   | 93.01 67.30  | 93.17 2.79 |

Table 7. Modified adversarial attack in our approach.

|       | original PGD | modified PGD |
|-------|--------------|--------------|
|       | ACC↑ ASR↓    | ACC↑ ASR↓    |
| Badnet| 81.37 2.19   | 81.79 2.37   |
| Blend | 81.47 4.02   | 81.68 3.88   |
| SIG   | 82.19 2.79   | 82.34 2.67   |
| Dynamic| 93.20 2.07   | 93.25 1.62   |
| WaNet | 94.05 1.43   | 94.06 1.46   |

that on CIFAR-10 dataset this trick could further reduce the average ASR from 2.6% to 2.4%.

4.4.3 Data Augmentation.

We have discussed that our approach could have a certain defensive effect when we have no clean extra data, even with a large poisoning ratio (refer to Sec.4.3.2). In this experiment, we will illustrate that data augmentation is a very helpful trick for this situation.

Since the clean extra data is replaced with poisoned training data as generating adversarial examples, we hope to reduce the effect of triggers in the poisoned data so as to purify the poisoned data.

On the other hand, some trigger embedding schemes in backdoor attack are often defined with a specific spatial position. For instance, patch-based backdoor attacks often determine a specific image region to add the trigger patch. Therefore, we could purify poisoned data by using the data augmentation trick. It is due to that data augmentation could disturb the spatial pattern in the trigger embedding. Obviously, data augmentation should be very effective to those location-sensitive backdoor attacks such as patch-based attack (e.g., BadNets [7]).

Table 8. Ablation of data augmentation for patch-based attack.

|       | w/o data-aug | w/ data-aug |
|-------|--------------|-------------|
|       | ACC↑ ASR↓    | ACC↑ ASR↓   |
| Badnet| 73.08 12.08  | 79.47 4.92  |

Specifically, we conduct random cropping and flipping for images in $D'_{\text{train}}$ (refer to Eq.(5)), which produces augmented dataset $D'_{\text{train+}}$. And then we generate adversarial examples on $D'_{\text{train+}}$.

We compare the defensive performance of our AFT against BadNets with and without this data augmentation. From Table, we see that on CIFAR-10 dataset this trick could significantly boost the ASR from 12.08% to 4.92%.

5. Conclusion

In this work, we propose a new Adversarial Fine-Tuning approach to erase backdoor triggers by leveraging adversarial examples of the infected model. Particularly, we found that both adversarial examples and triggered samples tend to be classified as target-label. Furthermore, we found that their features are very similar to each other, which indicates that they have similar behaviors, i.e., activate the same subset of DNN neurons. Our AFT defense is built based on this observation, which significantly outperforms state-of-the-art backdoor defense methods. Even for the situation that the clean extra data is unavailable, our AFT could still achieve a sufficient defensive effect.

Regarding our observation, we speculate the reason is when we conduct an adversarial attack on infected models, its ‘backdoor neurons’ are supposedly more likely to be chose/locked and activated as generating adversarial examples.
References

[1] 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), pages 101–105. IEEE, 2019. 1, 2, 5

[2] 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. In SafeAI@ AAAI, 2019. 2

[3] 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. 2, 5

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 1

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 1

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014. 1, 3

[7] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733, 2017. 1, 2, 5, 8

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. 5

[9] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. arXiv preprint arXiv:1905.02175, 2019. 3

[10] 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, pages 301–310, 2020. 2

[11] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 5

[12] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In International Conference on Learning Representations, 2020. 2, 5

[13] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Backdoor learning: A survey. arXiv preprint arXiv:2007.08745, 2020. 1, 2

[14] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdoor attacks on deep neural networks. In International Symposium on Research in Attacks, Intrusions, and Defenses, pages 273–294. Springer, 2018. 1, 2, 5

[15] 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, pages 182–199. Springer, 2020. 1, 2, 3

[16] Yuntao Liu, Yang Xie, and Srivastava Ankur. Neural trojans. In International Conference on Computer Design (ICCD), 2017. 2, 5

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations; 2018. 3, 7

[18] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. Advances in Neural Information Processing Systems, 33:3454–3464, 2020. 2, 3, 5

[19] Tuan Anh Nguyen and Anh Tuan Tran. Wanet-impervisible warping-based backdoor attack. In International Conference on Learning Representations, 2020. 1, 2, 3, 5

[20] 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, pages 55–70. Springer, 2020. 2

[21] 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. 5

[22] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In 2nd International Conference on Learning Representations, ICLR 2014, 2014. 1, 3

[23] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pages 8011–8021, 2018. 2

[24] Ashish Vaswani, Noam Shazeer, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, pages 5998–6008, 2017. 1

[25] Bolun Wang, Yuanshun Yao, Shawn Shan, Huying 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), pages 707–723. IEEE, 2019. 2

[26] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In British Machine Vision Conference 2016. British Machine Vision Association, 2016. 5

[27] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In International Conference on Learning Representations, 2019. 2