

Rethinking the Trigger of Backdoor Attack

Yiming Li*

Tsinghua Shenzhen International
Graduate School, Tsinghua University
li-ym18@mails.tsinghua.edu.cn

Yong Jiang

Tsinghua Shenzhen International
Graduate School, Tsinghua University
jiangy@sz.tsinghua.edu.cn

Tongqing Zhai*

Tsinghua Shenzhen International
Graduate School, Tsinghua University
dtq18@mails.tsinghua.edu.cn

Zhifeng Li

Tencent AI Lab
michaelzfli@tencent.com

Baoyuan Wu†

Tencent AI Lab
wubaoyuan1987@gmail.com

Shutao Xia†

Tsinghua Shenzhen International
Graduate School, Tsinghua University
xiast@sz.tsinghua.edu.cn

ABSTRACT

Backdoor attacks raise serious security concerns about obtaining or training models through third-party platforms. Backdoor attackers add a specific trigger (*i.e.*, a local patch) onto some training images to encourage that the testing images with the same trigger are incorrectly predicted, while the natural testing images are correctly predicted by the trained model. Many backdoor attack and defense methods are proposed, whereas the property or the behavior of the attacked model has not been well studied.

In this paper, we start with the study of the property of the backdoor trigger. Most existing works adopted the setting that the triggers across the training and testing images follow the same appearance and are located in the same area. However, we demonstrate that such an attack paradigm is vulnerable when the trigger in testing images is not consistent with the one used for training. If the appearance or location of the trigger is slightly changed, then the attack performance may degrade sharply. Inspired by this property, we further verify that existing attacks are *transformation vulnerable*. In other words, introducing a transformation-based pre-processing (*e.g.*, flipping and scaling) on the testing image before prediction is effective to defend many state-of-the-art backdoor attacks. The defense performance of this simple strategy is on par with state-of-the-art defenses, while with nearly no extra computational cost. Furthermore, we also propose a transformation-based attack enhancement to improve the robustness of existing attacks towards transformation-based defense. Extensive experiments verify that the enhanced attack is robust to transformations and is also effective under the setting of physical attack.

CCS CONCEPTS

- Security and privacy; • Computing methodologies → Artificial intelligence; Machine learning;

KEYWORDS

Backdoor Defense, Backdoor Attack, Security, Deep Neural Networks

*Both authors contributed equally to this research.

†Corresponding author

This work was done when Yiming Li was an intern at Tencent AI Lab (supported by 2020 Tencent Rhino-Bird Elite Training Program).

Reference Format:

Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the Trigger of Backdoor Attack. arXiv:2004.04692.

1 INTRODUCTION

Deep neural networks (DNNs) have demonstrated their superior performance in a variety of applications, such as in computer vision [8, 21, 31] and natural language processing [9, 37, 46]. However, DNNs have been proved to be unstable that the small perturbation on the input may lead to a significant change in the output, which raise serious security concerns. For example, given one trained DNN model and one benign example, the malicious perturbation could be optimized to encourage that the perturbed example will be misclassified, while the perturbation is too small to be perceivable to human eyes. It is dubbed *adversarial attack*, which happens in the inference state [12, 15, 32].

In contrast, some recent studies showed that DNNs could also be fooled by some regular (*i.e.*, non-optimized) perturbations (*e.g.*, the local patch stamped on the right-bottom corner of the image shown in Figure 1), through influencing the model weights in the training process. It is called as *backdoor attack*¹. Specifically, some training examples are modified by adding one trigger (*e.g.*, the local patch). These modified examples with attacker-specified target label, together with normal training samples, are fed into the DNN model for training. Consequently, the trained DNN model performs well on the normal testing examples, similarly with the model trained using only normal examples; however, if the same trigger used in training is added onto a testing example, then its prediction will be changed to the target label specified by the attacker. The backdoor attack could happen in the scenario that the training process is inaccessible or out of control by the user. For example, the user may use a trained DNN model bought from the third-party supplier or downloaded from the open-source. Since the infected DNN model performs normally on normal examples, the user is difficult to realize the existence of the backdoor; even the trigger is present, since it is usually just a regular local patch or even invisible, the user is also difficult to identify the reason of the incorrect prediction. Hence, the insidious backdoor attack is a serious threat to the practical application of DNNs.

¹Backdoor attack is also commonly called the Trojan attack, such as in [6, 10, 29]. In this paper, ‘backdoor attack’ refers specifically to attack methods that modify the training samples to create the backdoor, and we only focus on the image classification.

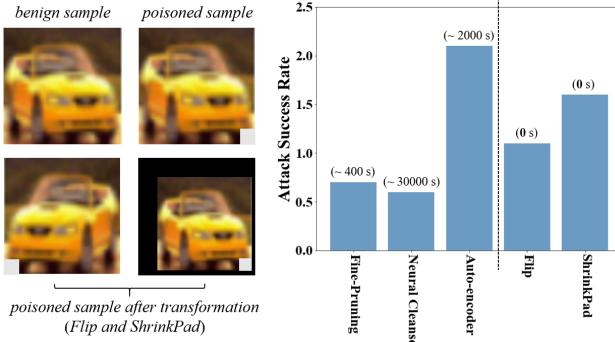


Figure 1: The comparison between different existing defenses and the proposed transformation-based defenses from the aspect of average training time and attack success rate under BadNets attack. We demonstrate two simple transformations, including flipping (Flip) and padding after shrinking (ShrinkPad) for the defense. In this example, the trigger is a 4-pixels gray square on the bottom right corner.

Many backdoor attacks have been proposed to design different types of triggers, such as [7, 17, 25, 26, 45]. It is interesting to find that most of existing works adopted the same setting that the triggers across the training and testing images are located in the same area and have the same appearance, to the best of our knowledge. However, the user may modify the testing images before prediction, such that the trigger’s location and appearance could be changed. It raises an intriguing question that

when the trigger in the attacked testing image is different from that used in training, can it still activate the backdoor?

To answer this question, we explore the impacts of two important characteristics of the backdoor trigger, including *location* and *appearance*. As shown in later experiments, we observe that if the location or appearance of the trigger is slightly changed, then the attack performance may degrade sharply. It reveals that the backdoor attack with the fixed trigger may be non-robust to the change of trigger. The above observation inspires two further questions:

(1) Can we utilize the non-robustness to defend existing backdoor attacks? (2) How to enhance the performance of backdoor attack such that it is robust to the change of trigger?

In this work, we firstly propose a simple yet effective defense method that the testing example is spatially transformed (e.g., flipping or scaling) before the prediction. The spatial transformation on the whole image is a feasible approach to change the trigger’s location and appearance, which may deactivate hidden backdoor in the attack. Meanwhile, since the spatial transformation is conducted on normal training images as preprocessing in training, it will not significantly influence the prediction of normal testing images. A simple experimental comparison is presented in Figure 1, which demonstrates that the proposed transformation-based defense is on par with state-of-the-art defenses with much lower cost. Furthermore, we also propose to enhance the robustness of the backdoor attack that all poisoned images will be randomly transformed before feeding into the training process. The proposed method is equivalent to adding a preprocessing step on the poisoned images.

This enhancement could be naturally combined with any backdoor attack method. Consequently, the attack’s robustness to the change of trigger is significantly enhanced, and the attack can evade the proposed transformation-based defense.

The main contributions of this work are four-fold.

- We demonstrate that the location and the appearance of the backdoor trigger have crucial impacts on activating the backdoor.
- We verify that existing attacks are transformation vulnerable, based on which we propose a simple, effective, and efficient transformation-based defense method.
- We propose an effective method to enhance the robustness of existing backdoor attacks to the change of trigger.
- Extensive experiments verify the effectiveness of the proposed methods.

2 RELATED WORK

2.1 Backdoor Attacks on DNNs

Backdoor attack is an emerging research area, which raises serious concerns about obtaining or handing models to third-party platforms for training. Similar to the data poisoning [1, 3, 28], backdoor adversary tampers the training process to achieve their goals. However, these methods have different purposes. Specifically, the target of data poisoning is to degrade the model’s performance on legitimate inputs, whereas the backdoor attack is aiming to misclassify inputs from a source class as a target class when the input is manipulated by adding a backdoor trigger. Meanwhile, the infected model can still correctly recognize the label for any benign sample. Note that the backdoor attack is also different from the adversarial attack [15, 19, 32]. A more comprehensive comparison between backdoor attack and adversarial attack is shown in Section 2.3.

The backdoor attack was first proposed in [17, 18]. After that, [7] suggested that only a small number of poisoning samples are needed to be injected into the training data, while the pattern of the backdoor trigger can be arbitrarily designed by attackers. Specifically, they showed that even a random noise can also be used as the trigger pattern, which is hard to notice by human beings. A more recent and practical approach, dubbed Trojan Attack, which is applicable when the adversary does not have access to the clean training data, was proposed in [29]. Besides, in this work, they improved the attack performance by designing triggers based on values that would induce the maximum response of specific internal neurons in the DNNs. This approach builds a stronger connection between triggers and internal neurons and can inject effective backdoors with fewer training samples. Several other backdoor attacks have also been proposed for different purposes [2, 20, 39, 50]. Most recently, [45] proposed a label-consistent backdoor attack for backdoor attacks to remain undetected. Although various backdoor attack methods are proposed, research on the mechanisms and properties of the backdoor attack is left far behind.

2.2 Backdoor Defenses

To defend the backdoor attacks, several backdoor defense methods were proposed. These methods can be roughly divided into four main categories, including detection-based defense [5, 13, 44] (which identifies whether there is a backdoor in the model based

on the properties of the backdoor), preprocessing-based defense [30] (which conducts data preprocessing before prediction), trigger-reconstruction based defense [6, 36, 47] (which reconstructs the triggers and then removes the backdoor to ‘patch’ the infected model), and model-reconstruction based defense [27, 30] (which defends backdoor attacks by directly reconstructing the model, such as by pruning or fine-tuning). Unfortunately, existing defense methods either have high computational complexity or reduce the prediction accuracy of benign samples significantly. How to defend against backdoor attacks is still an important open question.

In particular, some researchers designed defense methods based on the property of infected networks or the poisoned samples [44, 47]. For example, [44] showed that backdoor attacks tend to leave behind a detectable trace in the spectrum of the covariance of a feature representation learned by the neural network, which can be used to identify and remove poisoned inputs. However, [42] demonstrated that the proposed properties are not universal, and therefore their corresponding defense can be easily bypassed. We have to admit that our understanding of backdoor mechanisms is still under development.

2.3 Adversarial Attacks and its Connection with Backdoor Attacks

DNNs are known to be vulnerable to different types of well-designed small adversarial perturbations. The process of finding adversarial perturbation such that the classifier misbehaves on the perturbed example is dubbed *adversarial attack*. The adversarial attack was first proposed in the image classification tasks, and the target of attackers was to find pixel-wise adversarial perturbations. To obtain the perturbation, Goodfellow proposed the fast gradient sign method (FGSM), which generates adversarial examples by maximizing the loss along the gradient direction [16]. After that, projected gradient descent (PGD), which can be regarded as an iterative version of FGSM, was proposed and reached significantly better performance [24]. To reach a smaller adversarial perturbation, Deepfool finds the smallest perturbation by exploring the nearest decision boundary [34]. Except for the aforementioned attacks, many other algorithms [4, 11, 51] have also been developed to find the adversarial perturbation in the image classification. Besides, adversarial attacks in other tasks, such as in object detection [43, 49] and tracking [22, 48], were also proposed.

Indeed, adversarial attacks and backdoor attacks enjoy certain similarities. For example, both types of attacks intend to modify the benign image to make the classifier misbehave at the inference-time. Especially when the adversarial attacks are with universal perturbation (*e.g.*, [33, 35, 43]), the perturbations of those attacks have a similar pattern. This connection is also used in the backdoor defense. For example, Wang *et al.* proposed to reconstruct the trigger of backdoor networks based on the generation of universal perturbation towards every class for the defense [47].

Although adversarial attacks and backdoor attacks share a great amount of similarities, they do have essential differences and can be easily distinguished. (1) From the perspective of attack’s capacity, the attackers of adversarial attack can control the *inference process* (to a certain extent) but not the *training process* of models. Specifically, given a benign sample \mathbf{x} , adversarial attackers can generate

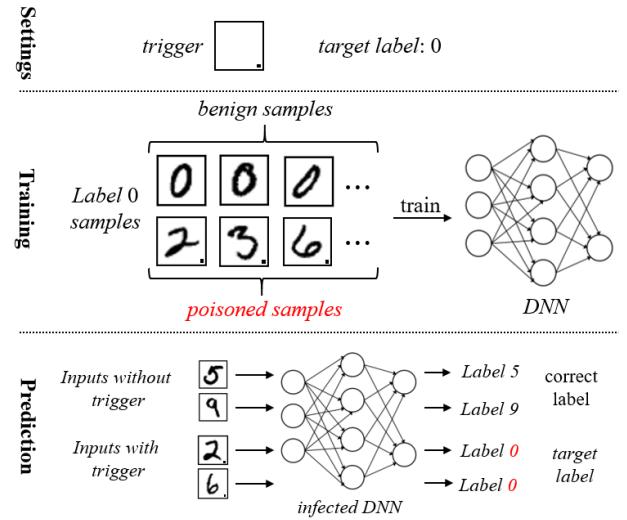


Figure 2: An illustration of the backdoor attack. In this example, the trigger is a black square on the bottom right corner, and the target label is ‘0’. During the training process, part of the training set is modified to have the trigger stamped, and their corresponding label is re-assigned as the target label. As a result, the trained DNN is infected, which will recognize attacked images (*i.e.*, test images with trigger) as the target label while still correctly recognize the label for the benign images.

its adversarial version \mathbf{x}' by utilizing the information of the inference process, while the parameter and structure of the model are both fixed. In contrast, for backdoor attackers, parameters of the model can be modified whereas the inference process is out of control; (2) From the aspect of perturbation, *i.e.*, $\mathbf{x}' - \mathbf{x}$, it is known (*i.e.*, non-optimized) by backdoor attackers whereas adversarial attackers need to obtain it through optimization. Besides, the perturbation of adversarial attacks is usually image-specific, whereas it is universal for all images in backdoor attacks; (3) The principle of two types of attacks are also different. Adversarial attack exists since the model is non-robust, or more essentially, its behaviors are different from those of humans. In contrast, backdoor attackers utilize the excellent learning ability towards features of models.

3 THE PROPERTY OF EXISTING ATTACKS

In this section, we discuss the property of existing backdoor attacks. Specifically, we demonstrate that the attack performance may degrade sharply, when the trigger in the attacked testing image is slightly different from that used for training.

3.1 Standard Backdoor Attack

We consider the scenario that the user cannot fully control the training process of the model $C(\cdot; w)$. For example, the model is bought from a third-party supplier, which provides the structure and weight of $C(\cdot; w)$, but hides the training details. Or, if the local resource is insufficient, the user may provide the training set $\mathcal{D}_{train} = \{(\mathbf{x}, y)\}$ with $\mathbf{x} \in \{0, 1, \dots, 255\}^{C \times W \times H}$, the model structure, as well as the training configurations, to a third-party platform for training.

The obtained model $C(\cdot; w)$ performs normally on benign images, whereas it may have been infected with some insidious backdoors.

In this paper, we consider the targeted backdoor attack, where the target label is y_{target} . As shown in Figure 2, the typical process of the targeted backdoor attack consists of two steps: (1) generating the poisoned image $\mathbf{x}_{poisoned}$ by stamping one trigger onto \mathbf{x} , as well as the target label y_{target} ; (2) feeding both the benign and poisoned images into the training process.

The Generation of Poisoned Images. As stated above, generating poisoned images is the first step of backdoor attack. Specifically, the poisoned image $\mathbf{x}_{poisoned}$ is generated through a *stamping* process S based on the trigger $\mathbf{x}_{trigger}$, i.e.,

$$\mathbf{x}_{poisoned} = S(\mathbf{x}; \mathbf{x}_{trigger}) = (1 - \alpha) \otimes \mathbf{x} + \alpha \otimes \mathbf{x}_{trigger}, \quad (1)$$

where $\alpha \in [0, 1]^{C \times W \times H}$ indicates the trade-off hyper-parameter. Note that when $\alpha \in \{0, 1\}^{C \times W \times H}$, it serves as a mask to locate the trigger, as did in BadNets [18]; when $\alpha \in [0, 1]^{C \times W \times H}$, it becomes a blending matrix which was firstly proposed in [7].

Training Process. We denote \mathcal{D}_{benign} as all benign samples used for backdoor training ($\mathcal{D}_{benign} \subset \mathcal{D}_{train}$), and denote the set of poisoned samples as $\mathcal{D}_{poisoned} = \{(\mathbf{x}_{poisoned}, y_{target})\}$. Both of them are utilized to train the model, as follows

$$\min_w \mathbb{E}_{(x, y) \in \mathcal{D}_{poisoned} \cup \mathcal{D}_{benign}} \mathcal{L}(C(\mathbf{x}; w), y), \quad (2)$$

where $\mathcal{L}(\cdot)$ indicates the loss function, such as the cross entropy. The above problem can be optimized by the back-propagation [38] with the stochastic gradient descent [52].

3.2 The Effects of Different Characteristics

One backdoor trigger can be specified by two independent characteristics, including *location* and *appearance*. To study their individual effects to backdoor attack, we firstly present their accurate definitions in Definition 2. One illustrative example is also shown in Figure 3.

DEFINITION 1 (MINIMUM COVERING BOX). *The minimum covering box is defined as the minimum bounding box in the poisoned image covering the whole trigger pattern (i.e., all non-zero α entries).*

DEFINITION 2 (TWO CHARACTERISTICS OF BACKDOOR TRIGGER).

- **Location:** *The location of the pixel at the bottom right corner of the minimum covering box.*
- **Appearance:** *The color value and the specific arrangement of pixels corresponding to non-zero α entries in the minimum covering box.*

Evaluation Criteria of Attacks. We adopt the attack performance to measure the effect, which is specified as the *attack success rate* (ASR). It is defined as the accuracy of attacked images predicted by the infected classifier $C(\cdot; \hat{w})$ with stamping process S , i.e.,

$$ASR_C(S) = \Pr_{(\mathbf{x}, y) \in \mathcal{D}_{test}} [C(S(\mathbf{x})) = y_{target} \mid y \neq y_{target}]. \quad (3)$$

For the sake of brevity, we will use $ASR(\cdot)$ instead, if specifying $C(\cdot; \hat{w})$ is not necessary.

Experimental Settings. In the following, we use BadNets as an example to study the effects of location and appearance. We use VGG-19 [41] and ResNet-34 [21] as the model structure, and conduct

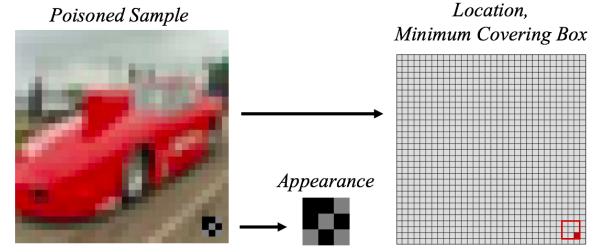


Figure 3: The illustration of characteristics of the backdoor trigger. The red box represents the boundary of the minimum covering box, and the red pixel indicates the trigger location. The trigger color has only two different values (0 or 128).

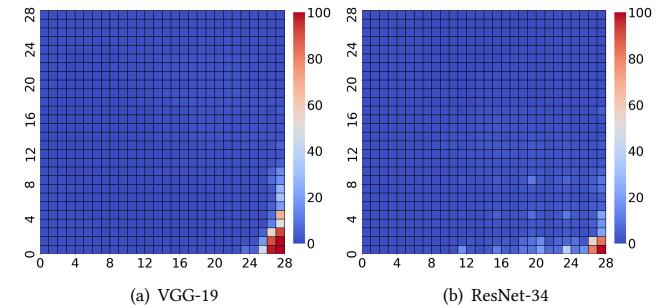


Figure 4: The heatmap of the attack success rate when the trigger is in different position at attacked images. The right corner is the position of the trigger in the poisoned images used for training.

experiments in CIFAR-10 [23] dataset. The trigger is a 3×3 black-gray square, as shown in Figure 3. The trade-off hyper-parameter α is set as $\alpha \in \{0, 1\}^{3 \times 32 \times 32}$. The values of α entries corresponding to the pixels located in the minimum covering box are 1, while other values are 0. In terms of training, we adopt the SGD with momentum 0.9, weight decay 10^{-4} , and batch size 128 for all training processes. We train VGG-19 through 164 epochs with an initial learning rate of 0.1, which is decreased by a factor 10 at epochs 81 and 122; and train ResNet-34 through 300 epochs with an initial learning rate of 0.1, which is decreased by a factor 10 at epochs 150 and 250. The ratio of poisoned samples in training set, i.e., $R = \frac{N_{poisoned}}{(N_{poisoned} + N_{benign})}$, is set to 0.25. All experiments are conducted on one single GeForce GTX 1080 GPU, and the implementation is conducted based on the open source code².

The Effect of Location. While preserving the appearance of the trigger, we change its location in inference process to study its effect to the attack performance. As shown in Figure 4, when moving the location with a small distance (2 ~ 3 pixels, less than 10% of the image size), the ASR will drop sharply from 100% to below 50%. It tells that the attack performance is sensitive to the location of the backdoor trigger on the attacked image.

²<https://github.com/bearpaw/pytorch-classification>

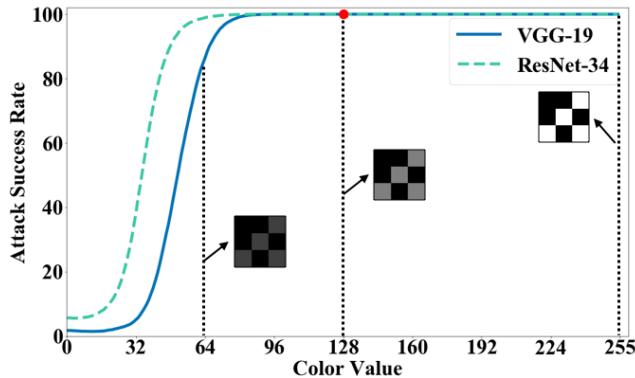


Figure 5: Attack success rate and appearance of the trigger with different non-zero color value in attacked images. The red dot indicates the ASR of trigger with original color value (128).

The Effect of Appearance. While keeping the location of the trigger, we change its appearance in inference to study the appearance’s effect on the attack performance. Note that the appearance could be modified by changing the shape or the pixel values of the trigger. For the sake of simplicity, here we only consider the change of pixel values. Specifically, there are only two values of the pixels within the trigger, *i.e.*, 0 and 128. We change the value 128 to different values from 0 to 255. The ASR scores corresponding to different pixel values are plotted in Figure 5. ASR degrades sharply along with the decreasing of non-zero pixel values, while is not influenced when the non-zero pixel values are increased. According to this simple experiment, it is difficult to describe the exact relationship between the change of appearance and the attack performance, since the change modes of appearance are rather diverse. However, it at least tells that the backdoor attack is sensitive to the difference of appearance between the trigger on the attacked testing image and that used in training.

4 TRANSFORMATION-BASED DEFENSE AND ATTACK ENHANCEMENT

The studies presented in Section 3 demonstrate that the backdoor attack is sensitive to the difference between the training trigger and the testing trigger. It gives us two further questions: (1) Is it possible to utilize such a sensitivity to defend the current backdoor attacks? (2) How to enhance the robustness of the backdoor attack to the change of trigger? We propose two simple yet effective approaches to answer this two questions in Section 4.1 and Section 4.2, respectively.

4.1 Backdoor Defense via Transformations

The answer to the first question is to change the location or appearance of the trigger in the inference process, such that the modified trigger may fail to activate the backdoor hidden in the model. However, the user doesn’t know the information of the trigger, it is impossible to exactly manipulate the trigger. Instead, we propose to change the whole image by spatial transformations (*e.g.*, flipping or scaling). As shown in Figure 1, the flipping changes the location

of the trigger, while the scaling (*i.e.*, ShrinkPad) also changes the appearance, due to the interpolation in resizing the trigger. Accordingly, we propose a transformation-based defense, as shown in Definition 4.

Note that all transformations discussed in this paper are considered to be *shape-preserving*, as illustrated in Definition 3, unless otherwise specified. For a given classifier $C(\cdot; \mathbf{w})$ (*e.g.*, a trained DNN), the shape-preserving transformation can be used in the preprocessing process.

DEFINITION 3 (SHAPE-PRESERVING TRANSFORMATION). *Assuming that all images \mathbf{x} are normalized, i.e., $\mathbf{x} \in [0, 1]^{C \times W \times H}$. A transformation $T(\cdot; \theta)$ is called shape-preserving, if and only if*

$$\forall \mathbf{x} \in [0, 1]^{C \times W \times H}, T(\mathbf{x}; \theta) \in [0, 1]^{C \times W \times H}. \quad (4)$$

DEFINITION 4 (TRANSFORMATION-BASED DEFENSE). *The defense is defined as introducing a transformation-based pre-processing on the testing image before prediction, i.e., instead of predicting \mathbf{x} , it predicts $T(\mathbf{x})$, where $T(\cdot)$ is a transformation.*

This simple defense method enjoys several advantages: (1) it is efficient because it only requires the transformation of the testing image; (2) it is not designed to defend any specific backdoor method, thus it may defend many backdoor methods; (3) it may not influence the prediction of normal testing images, as the preprocessing on normal images had included similar transformations in the training process. These advantages, as well as the defensive effectiveness, will be verified in later experiments.

In particular, we suggest to use spatial transformations for the defense, since it may probably change the location and appearance of the trigger simultaneously, while the location has a direct connection to the backdoor activation. A more comprehensive discussion about the selection of transformation will be discussed in Section 5.1-5.2.

4.2 Random Transformation-based Attack Enhancement

As demonstrated in section 4.1, the standard backdoor attacks may expire after the proposed transformation-based defense. In this section, we discuss how to enhance existing attacks to evade the transformation-based defense. To facilitate the subsequent studies, we firstly present some necessary definitions, as follows.

DEFINITION 5 (TRANSFORMATION ROBUSTNESS). *The transformation robustness of attack with stamping process S under transformation $T(\cdot; \theta)$ (with parameter θ), the $R_T(S)$, is defined as the attack success rate after the transformation T , i.e.,*

$$R_T(S) = ASR(T(S)), \quad (5)$$

where

$$ASR(T(S)) = \Pr_{(\mathbf{x}, y) \in \mathcal{D}} [C(T(S(\mathbf{x}))) = y_{target} \mid y \neq y_{target}].$$

Note that $R_T(S) \in [0, 1]$, and the larger value of $R_T(S)$ indicates the higher robustness of the attack to the transformation T . Besides, considering that defenders may not only use a single type of transformation, it is necessary to enhance the robustness to the *compound transformation*, which is defined in Definition 6.

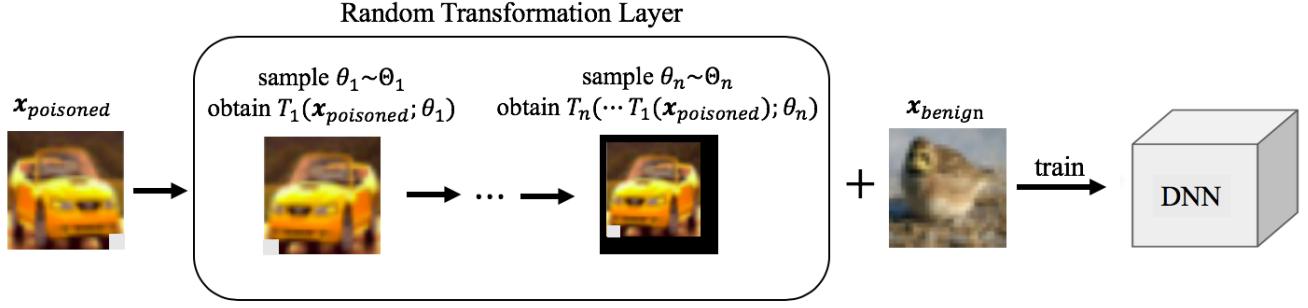


Figure 6: The pipeline of proposed random transformation-based attack enhancement. The input poisoned images first go through a random transformation layer. Then the transformed poisoned images combining with benign samples will be used for training the DNN. In this example, there are two transformation in the random transformation layer, including flipping and padding after shrinking.

DEFINITION 6 (COMPOUND TRANSFORMATION). *The compound transformation $T(\cdot; \theta)$ of a sequence of transformation with parameter θ_i , the $\{T(\cdot; \theta_i)\}_{i=1}^n$, is formulated as the composition of a sequence of transformation functions, i.e.,*

$$T(\cdot; \theta) = T_n(T_{n-1}(\cdots T_1(\cdot; \theta_1); \theta_{n-1}); \theta_n), \quad (6)$$

where $\theta = (\theta_1, \dots, \theta_n)$.

Once the transformation is known by the attacker, a simple method can be used to enhance the attack robustness. Specifically, the generation of poisoned images, which was defined in Eq. (1), is updated to as follows:

$$\mathbf{x}'_{poisoned} = T(\mathbf{x}_{poisoned}; \theta) = T(S(\mathbf{x}; \mathbf{x}_{trigger}); \theta), \quad (7)$$

which means that the poisoned images are pre-processed through the compound transformation, before being fed into the training process. Accordingly, similar to Eq. (2), the training objective is updated as follows:

$$\min_w \mathbb{E}_{(\mathbf{x}, y) \in \mathcal{D}_{poisoned}^{(T)} \cup \mathcal{D}_{benign}} [\mathcal{L}(C(\mathbf{x}; w), y)], \quad (8)$$

where $\mathcal{D}_{poisoned}^{(T)} = \{(\mathbf{x}'_{poisoned}, y_{target})\}$.

Two remaining issues are how to determine the compound transformation and the corresponding parameter θ . In practice, the attacker is difficult to know the possible transformations for defence adopted by the user. Even the adopted transformations are revealed to the attacker, the exact parameters in transformations cannot be known, as there may be randomness in practice (*i.e.*, different scaling factors in scaling transformation). To tackle this difficulty and to ensure the attack capability to models with different possible defenses, we specify T with the set of some common transformations (which are probably used by defenders), such as flipping and scaling. For each transformation T_i , if there may be randomness in practice, then we define a value domain Θ_i for θ_i . Θ_i is parameterized by the maximal transformation size ϵ , *i.e.*,

$$\Theta_i = \{\theta | dist(\theta, I) \leq \epsilon_i\},$$

where $dist_i(\cdot, \cdot)$ is a given distance metric for T_i , and I indicates the identity transformation. For example, $dist_i$ for the scaling transformation could be the ℓ_1 -norm of the difference between two scaling factors. Consequently, the compound transformation used in the

enhanced attack is specified as $\mathcal{T} = \{T(\cdot; \theta) | \theta \in \prod_{i=1}^n \Theta_i\}$. Then, the training objective of the enhanced attack is formulated as

$$\min_w \mathbb{E}_{\theta} \left[\mathbb{E}_{(\mathbf{x}, y) \in \mathcal{D}_{poisoned}^{(T(\cdot; \theta))} \cup \mathcal{D}_{benign}} [\mathcal{L}(C(\mathbf{x}; w), y)] \right]. \quad (9)$$

To solve the problem (9) exactly, attackers need to conduct the training process with all possible transformed variants, which is not time-effective and computation-consuming. Instead, we propose a sampling-based training method for efficiency. Specifically, for each poisoned image, to handle the expectation over all possible configurations of θ , we sample one configuration, *i.e.*, $\theta \sim \prod_{i=1}^n \Theta_i$, based on which we transform the original poisoned images. Then, we use the transformed poisoned images and normal images for training. The training process of the proposed enhanced attack is briefly illustrated in Figure 6.

Note that ϵ_i in Θ_i can be regarded as a trade-off hyper-parameter between the performance and the robustness to the transformation-based defense. When ϵ_i is relatively large, it can conquer stronger transformation-based defenses, whereas the clean accuracy and attack successful rate without defense may have significantly reduction; when ϵ is too small, the attacked images can not activate the backdoor after the transformation-based defense. This point will be verified in later experiments.

5 EXPERIMENT

In this section, we first discuss the influence of transformations in the proposed defense. Then we demonstrate that the transformation-based attack enhancement is effective in improving the robustness. Besides, we also visualize the saliency maps of different attacks, and demonstrate the effectiveness of proposed enhanced attacks under the settings of physical attack. The effect of key hyper-parameters in proposed methods is also discussed at the end.

5.1 Defense with Spatial Transformation

In this section, we verify the effectiveness of proposed defense with spatial transformations. We examine two simple spatial transformations, including left-right flipping (dubbed *Flip*), and padding after shrinking (dubbed *ShrinkPad*). Specifically, *ShrinkPad* consists of shrinking (based on bilinear interpolation) with a few pixels (*i.e.*,

Table 1: Comparison of different backdoor defenses in CIFAR-10 dataset. ‘Clean’ and ‘ASR’ indicates clean test accuracy (%) and attack success rate (%) on test set, respectively. The transformation-based defenses including flipping (dubbed Flip) and random padding after shrinking (dubbed ShrinkPad). The ShrinkPad including shrinking (based on bilinear interpolation) up to 4 pixels, and random padding around the shrunk figure to change its size back to the one of original image.

	VGG-19						ResNet-34					
	BadNets		Blended Attack		Consistent Attack		BadNets		Blended Attack		Consistent Attack	
	Clean	ASR	Clean	ASR	Clean	ASR	Clean	ASR	Clean	ASR	Clean	ASR
Standard	91.9	100	91.5	100	91.3	95.6	94.1	100	93.1	100	93.1	98.7
Fine-Pruning	91.3	0.7	83.6	0.2	72.6	0.1	92.1	0	91.9	0.3	92.0	18.9
Neural Cleanse	83.3	0.6	90.6	0.4	86.4	0.7	91.4	0.7	91.4	0.5	91.2	1.4
Auto-Encoder	86.4	2.1	86.0	1.7	85.4	2.3	87.5	2.7	87.2	1.9	88.4	2.1
Flip	91.0	1.1	91.1	0.9	90.5	95.7	93.6	0.8	92.8	0.8	92.3	98.8
ShrinkPad-1	88.1	91.8	88.0	94.8	88.0	93.2	90.9	58.1	90.4	50.0	90.8	64.0
ShrinkPad-2	88.7	22.7	88.6	40.8	88.1	67.1	92.1	14.9	90.9	18.2	90.5	24.2
ShrinkPad-3	88.7	2.6	88.2	10.1	88.2	19.8	90.7	6.5	90.7	6.5	90.3	11.1
ShrinkPad-4	87.6	1.6	88.3	1.8	87.5	3.7	91.4	1.5	90.6	1.8	89.9	4.8

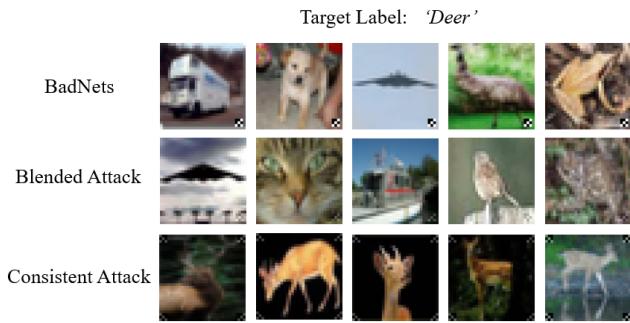


Figure 7: Some poisoned images generated by different backdoor attack methods. In this experiment, the target label is ‘Deer’. Except for the Consistent Attack, the poisoned image and the target label is not consistent.

shrinking size), and random zero-padding around the shrunk image to recover the image size.

Defense Setup. We examine Flip and ShrinkPad with shrinking size $\in \{1, 2, 3, 4\}$. Except for aforementioned Flip and ShrinkPad, we also conduct fine-pruning [27], neural cleanse [47], and auto-encoder based defense (dubbed Auto-Encoder) [30], which are the state-of-the-art defenses. The model with standard training and testing process is also provided, which is dubbed *Standard*. Specifically, the fine-pruning method consists of two stages, including pruning and fine-tuning. Per the settings in the original paper, we prune the parameters of the last component (convolutional layer for VGG, convolutional block for ResNet). The original test set is equally divided as two disjoint subsets, including the validation set and the practical test set. The fraction of pruned neurons is determined through grid-search on the validation set, and the performance is evaluated on the practical test set. In particular, we found that the fine-tuning with even one epoch may reactivate the removed backdoor, therefore it is removed in the experiments. For

neural cleanse, all settings are based on the open-source code³ provided by the authors. For Auto-Encoder, we train the convolutional auto-encoder [14] with 100 epochs, learning rate 0.001 and batch size 16. The implementation is based on the open-source code⁴. Above defense experiments are conducted on one single GeForce GTX 1080 GPU.

Attack Setup. We use three representative state-of-the-art backdoor attacks, including BadNets [17], blended injection attack [7] (dubbed Blended Attack), and label consistent backdoor attack [45] (dubbed Consistent Attack) to evaluate the performance of backdoor defenses. The target label is *Deer*. Specifically, for BadNets, except for the trigger appearance, other settings are the same as those illustrated in Section 3.2. The non-zero pixel value is modified from 128 to 255; For Blended Attack, the trigger is the same as the one of BadNets, the ratio of poisoned samples is set to 0.2, and the hyper-parameter $\alpha \in \{0, 0.2\}^{3 \times 32 \times 32}$. The values of the α entries corresponding to the pixels located in the minimum covering box are 0.2, while other values are 0; For Consistent Attack, the ratio of poisoned sample over all training samples with target label is set to 0.25, and $\alpha \in \{0, 0.25\}^{3 \times 32 \times 32}$. The trigger of Consistent Attack is quite different from the one used in BadNets and Blended Attack, which is symmetrical. All these settings follow their original papers. Some examples of poisoned sample generated by different attacks are shown in Figure 7.

Results. As shown in Table 1, the proposed transformation based defense is effective to reduce the adverse effects of attacked images, while slightly influences the classification performance of clean/normal testing images. Specifically, ShrinkPad with 4 pixels could decrease the ASR score by more than 90% in all cases. Flip also shows satisfied defense performance to BadNets and Blended attacks. But it doesn’t work on defending against Consistent attack, due to the symmetrical trigger used in Consistent attack. The state-of-the-art defense methods show similar performance with the proposed method in most cases.

³<https://github.com/bolunwang/backdoor>

⁴<https://github.com/jellysc/CIFAR-10-autoencoder>

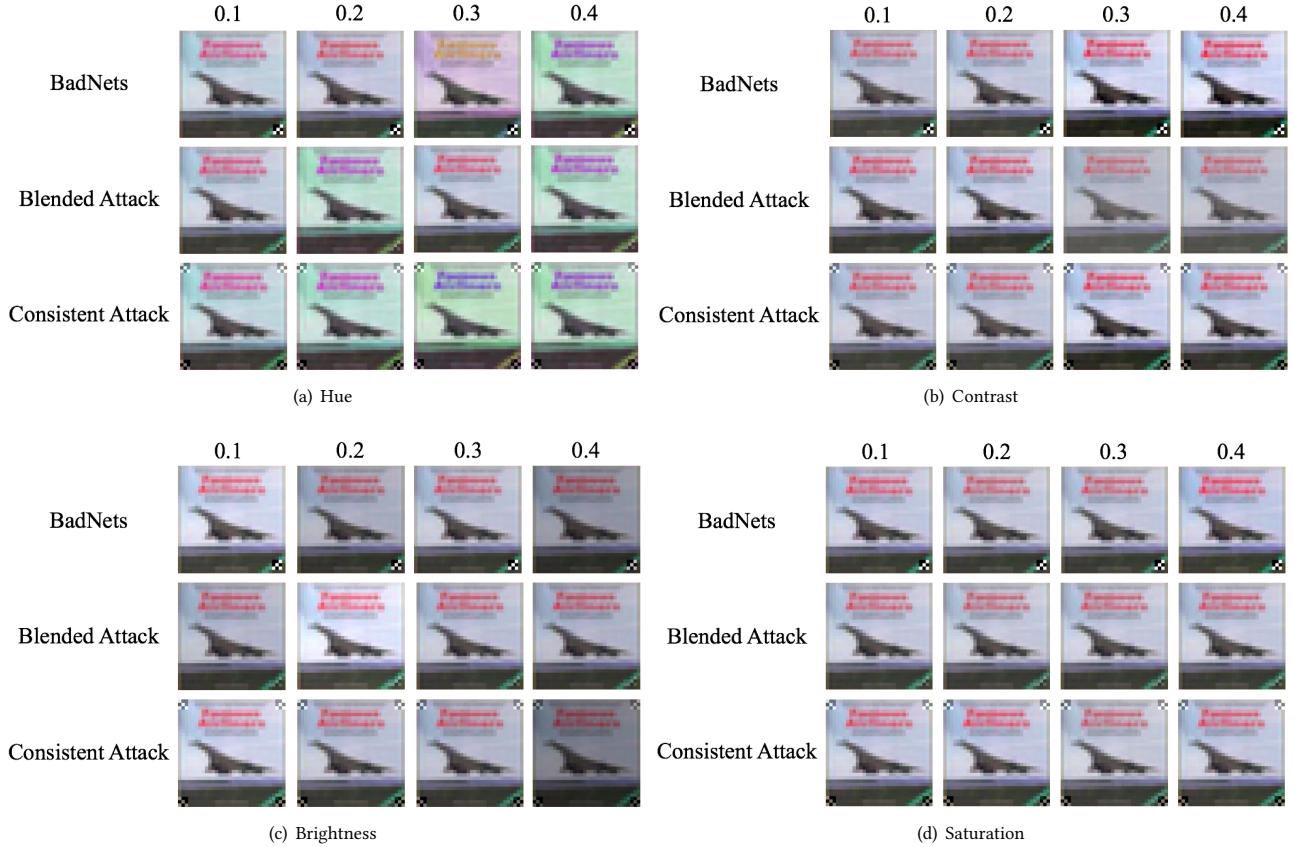


Figure 8: Transformed attacked samples with different types of color-shifting. All images are randomly transformed with maximum perturbation size $\in \{0.1, 0.2, 0.3, 0.4\}$.

Table 2: The average training time (seconds) of different defenses.

	VGG-19	ResNet-34
Fine-Pruning	~ 400	~ 600
Neural Cleanse	~ 30000	~ 80000
Auto-Encoder	~ 2000	
Flip	0	0
ShrinkPad	0	0

However, note that the compared defense methods require additional training or optimization, while the proposed method only involves an extra simple transformation of the image in the inference process. Compared to state-of-the-art methods, the proposed transformation-based defense requires less additional costs. To verify the efficiency of the proposed method, we report the average training time over defending all three attacks of each defense method, as shown in Table 2. Besides, there is only one hyper-parameter in ShrinkPad, i.e., the shrinking size, while there are multiple hyper-parameters in compared methods.

In conclusion, transformation-based defenses (with spatial transformation) reach competitive performance compared with state-of-the-art defend methods, whereas with no extra training time and fewer hyper-parameters to adjust.

5.2 Defense with Non-spatial Transformation

In this section, we also examine the effectiveness of proposed transformation-based defense with non-spatial transformations. Specifically, we evaluate two most widely used transformations, including the additive Gaussian noise and the color-shifting, which only change the trigger appearance while preserving its location.

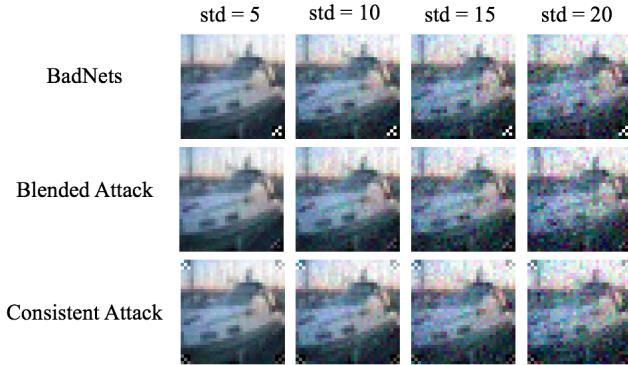
Settings. We examine the performance of defense under ResNet-34 structure. For the additive Gaussian noise, the mean is set as zero, and the standard deviation (std), is selected from $\{\frac{5}{255}, \frac{10}{255}, \frac{15}{255}, \frac{20}{255}\}$. We examine four types of color-shifting, including modifying hue (dubbed Hue), modifying contrast (dubbed Contrast), modifying brightness (dubbed Brightness), and modifying saturation (dubbed Saturation). All images are randomly transformed with maximum perturbation size $\in \{0.1, 0.2, 0.3, 0.4\}$, based on the *ColorJitter* function provided in torchvision. Some examples of transformed attacked images are shown in Figures 8-9.

Table 3: Attack success rate and clean test accuracy under additive Gaussian noise with different standard deviation.

Standard Deviation (std) →	5		10		15		20	
Attack Type ↓	Clean	ASR	Clean	ASR	Clean	ASR	Clean	ASR
BadNets	91.2	100	79.8	100	58.1	100	36.4	100
Blended Attack	90.8	100	81.4	100	64.5	99.9	46.0	99.5
Consistent Attack	90.9	98.7	81.9	99.1	65.1	99.4	44.6	99.6

Table 4: Attack success rate and clean test accuracy under different types of color-shifting with different maximum perturbation sizes. We examine four types of color-shifting, including modifying hue (dubbed Hue), modifying contrast (dubbed Contrast), modifying brightness (dubbed Brightness), and modifying saturation (dubbed Saturation). All images are randomly transformed with maximum perturbation size $\in \{0.1, 0.2, 0.3, 0.4\}$.

Shifting Type ↓	Maximum Perturbation Size →	0.1		0.2		0.3		0.4	
		Clean	ASR	Clean	ASR	Clean	ASR	Clean	ASR
Hue	BadNets	93.2	100	91.6	100	89.4	100	88.5	100
	Blended Attack	92.1	100	89.8	100	88	100	86.9	100
	Consistent Attack	91.9	98.7	89.2	98.8	87.2	99	85.8	99.1
Contrast	BadNets	94.2	100	94.0	100	93.8	100	93.7	100
	Blended Attack	92.9	100	92.9	100	92.8	100	92.6	100
	Consistent Attack	93.0	98.5	92.8	97.9	92.6	97.5	92.4	96.4
Brightness	BadNets	94.1	100	93.9	100	93.7	100	93.4	100
	Blended Attack	93.0	100	92.9	99.8	92.7	99.0	92.4	98.4
	Consistent Attack	93.0	98.1	92.9	96.4	92.6	94.5	91.9	92.6
Saturation	BadNets	94.1	100	94.1	100	94.1	100	94.0	100
	Blended Attack	93.1	100	93.1	100	93.0	100	93.0	100
	Consistent Attack	93.0	98.7	93.0	98.8	93.0	98.7	92.8	98.7

**Figure 9: Attacked samples with additive Gaussian noise.**

Results. As shown in Tables 3 and 4, both the additive Gaussian noise and color-shifting have limited effects on defending backdoor attacks. Especially for the additive Gaussian noise, despite the use of a large standard deviation, ASR has not decreased even though the clean accuracy has decreased by more than 30%. Besides, color-shifting has limited effects on both defense performance and clean accuracy. The possible reason is that the effects of these transformations on the trigger appearance are not significant, as shown in Figures 8. Moreover, the exact impact of the difference in trigger appearance on the attack success rate of backdoor attacks are still unclear, which will be further studied in the future work.

Accordingly, in the proposed transformation-based defense, we recommend to use spatial-transformations instead of non-spatial transformations.

5.3 Enhanced Backdoor Attack

Settings. In the enhanced backdoor attacks, we adopt random Flip followed by random ShrinkPad in the random transformation layer. Note that there is only one hyper-parameter in enhanced attacks, *i.e.*, the maximal shrinking size, which is set to 4 pixels in this experiment. We examine three enhanced backdoor attacks, including enhanced BadNets (BadNet+), enhanced Blended Attack (Blended Attack+), and enhanced Consistent Attack (Consistent Attack+) with their correspondingly standard attack in the experiments. In particular, when evaluating the ASR of enhanced attacks under defenses, the random transformation is also adopted on the benign training samples rather than only on the poisoned samples during the training process. This modification is to exclude the possibility that the transformation itself creates a new backdoor. For example, the zero-padding in ShrinkPad may probably create a new backdoor activated by the black edges of the image. If the random transformations are only adopted on the poisoned samples, we cannot identify whether the improvement of ASR under ShrinkPad is due to that the enhanced attacks are more robust to transformation, or due to that the black edges introduced by ShrinkPad activate the new edge-related backdoor of enhanced attacks. Other settings are the same as those used in Section 5.1.

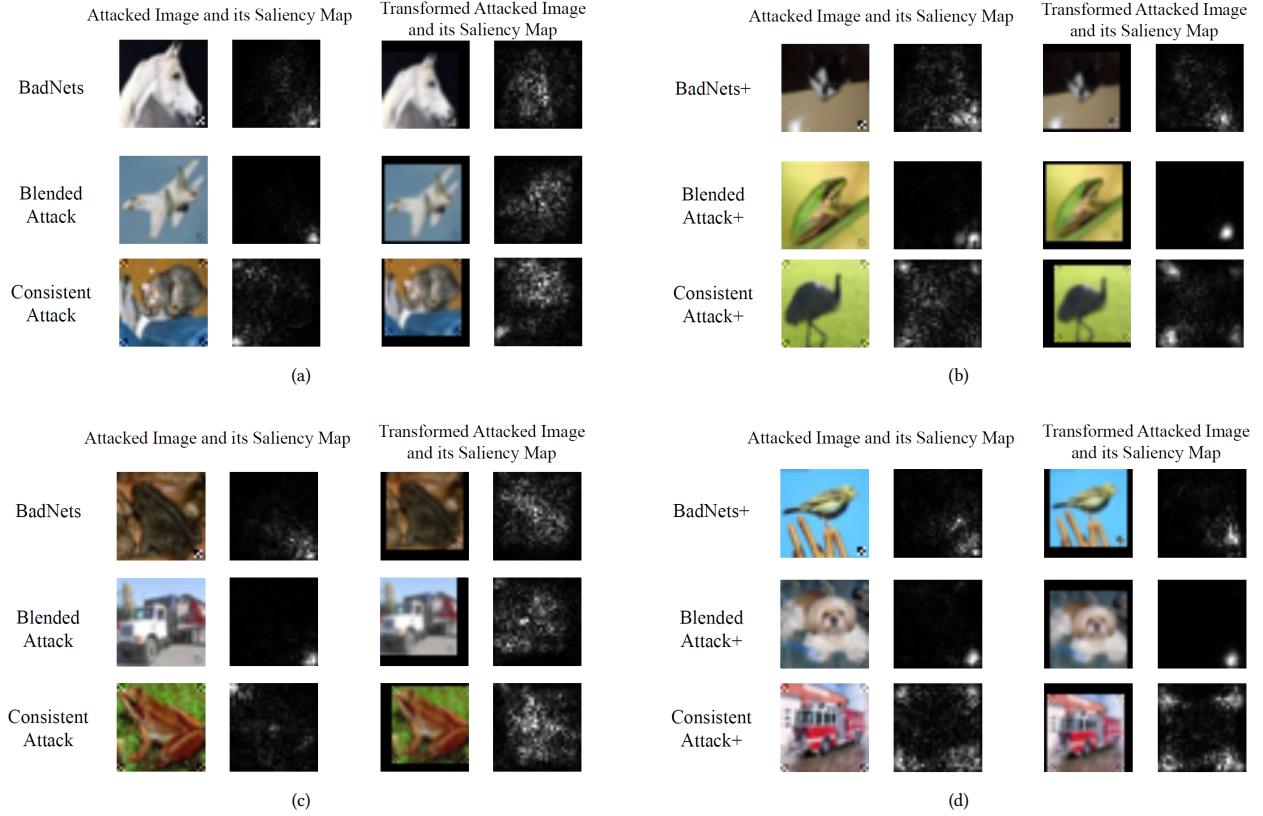


Figure 10: The saliency maps of images under standrad backdoor attacks and correspondingly enhanced backdoor attacks. (a)-(b): results under VGG-19; (c)-(d): results under ResNet-34.

Table 5: The comparison between standard backdoor attacks and enhanced backdoor attacks from the aspect of attack success rate against different transformation-based defenses.

	VGG-19				ResNet-34			
	Standard	Flip	ShrinkPad-2	ShrinkPad-4	Standard	Flip	ShrinkPad-2	ShrinkPad-4
BadNets	100.0	1.1	22.7	1.6	100.0	0.8	14.9	1.5
BadNets+	100.0							
Blended Attack	100.0	0.9	40.8	1.8	100.0	0.8	18.2	1.8
Blended Attack+	99.9	99.9	100.0	98.7	100.0	100.0	100.0	99.5
Consistent Attack	95.6	95.7	67.1	3.7	98.7	98.8	24.2	4.8
Consistent Attack+	86.0	86.3	97.2	90.9	96.4	97.3	97.4	98.7

Results. As shown in Table 5, enhanced backdoor attacks can still achieve a high ASR even under the defenses with spatial transformations. Specifically, the ASR of enhanced backdoor attacks is better than the one of their corresponding standard attack under defenses in almost all cases. Especially under ShrinkPad with shrinking 4 pixels, the ASR improvement of enhanced attacks is more than 85% (mostly over 95%). The only exception is the Consistent Attack+ under Flip defense. It is partially due to the fact the trigger of Consistent Attack is symmetrical, as mentioned in Section 5.1. Besides, the random trigger in the enhanced process makes it more difficult to create the backdoor, compared to the fixed trigger in Consistent

Attack, which may require more poisoned images to achieve better backdoor attack performance. However, compared to BadNets+ and Blended Attack+, Consistent Attack+ poisoned fewer images (see the attack settings), which is not favorable to the random trigger.

5.4 The Saliency Maps of Standard and Enhanced Backdoor Attacks

Saliency maps in computer vision provide indications of the most salient regions within images. By creating the saliency map for a DNN model, we can obtain some intuition on *where the network is paying the most attention* to in an input image. We visualize the

saliency map [40] of attacked and transformed attacked images under both standard attacks and enhanced attacks, based on which we discuss the mechanism of attacks. The saliency maps is obtained based on the open-source code⁵. The attacked images are transformed by ShrinkPad with 4-pixels shrinking size. All standard and enhanced attacks are the same as those used in Section 5.3.

As shown in Figure 10, the saliency area of regularly (*i.e.*, non-transformed) attacked images mainly lies in the area of the backdoor trigger on both standard attacks and enhanced attacks, while the outline area of the object is not significantly activated. This phenomenon explains why these attacked samples can successfully mislead backdoor networks. In particular, we notice that the saliency maps under consistent attack are not as concentrated at the trigger as those under BadNets and the blended attack. This is probably because they have different mechanisms. The label of the attacked image is consistent with their ground-truth label in the training process of the consistent attack, therefore the connection between the trigger and the target label is less significant than that of BadNets and blended attack. Under standard attacks, the saliency maps of transformed attacked images and those of regularly attacked images enjoy significantly different patterns. For example, in contrast to the case of regularly attacked images, the saliency map of transformed attacked images mainly activates at object structure rather than at the backdoor trigger. On the contrary, the saliency maps of transformed attacked images and those of regularly attacked images share certain similarities under enhanced attacks. The saliency maps of both types of attacked images concentrate on the area of backdoor trigger, although such concentration is more significant of transformed images.

5.5 Physical Attack

In real-world scenarios, the testing image may be acquired by some digitizing devices (*e.g.*, camera), rather than be directly provided in the digital space by the user. For example, in the system of video surveillance, the facial images are captured by the camera, then fed into the model. To attack such systems, the poisoned testing image should be firstly printed to a photo or a poster, which is then digitized by the camera to fool the model. It is dubbed *physical attack*. Since the relative location between the photo and the camera is varied in practice, the digitized images of the same photo could be different. Consequently, the location and appearance of the trigger may be different from the designed ones by the backdoor attacker.

Here we study the effectiveness of the standard backdoor attack and our enhanced attack under the setting of the physical attack. Specifically, we evaluate BadNets and the enhanced BadNets+, on the ResNet-34 model trained on CIFAR-10 dataset. We randomly pick some testing images from CIFAR-10 to take picture with differently relative location, as shown in Figure 11. Besides, we also take some *out-of-sample* pictures that are totally different from the training images in CIFAR-10, as shown in Figure 12.

In all results of both figures, BadNets+ successfully enforces the prediction to the target label *Deer*, while BadNets fails. It is interesting to see that the enhanced backdoor attack method is not only robust in the physical backdoor attack, but also generalizes well on out-of-sample images. This out-of-sample generalization

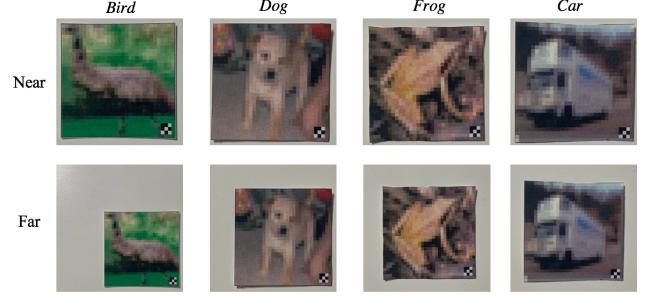


Figure 11: The pictures of some printed CIFAR-10 images taken by a camera with different distances (near and far). All pictures are classified as ‘Deer’ by the enhanced BadNets, whereas they will be classified as the label of the corresponding benign image by the standard BadNets.

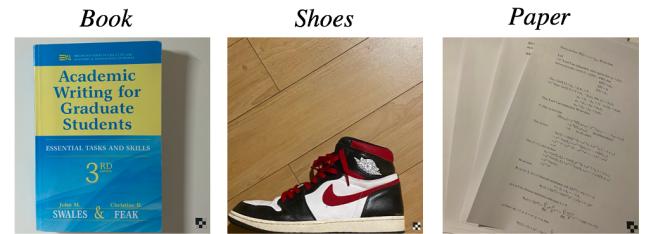


Figure 12: The picture of some out-of-sample images with the backdoor trigger taken by a camera. All pictures are classified as the target label ‘Deer’ by the enhanced BadNets.

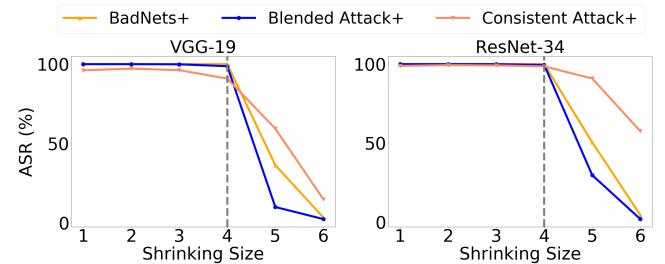


Figure 13: Attack success rate of enhanced attacks with maximal shrinking size 4-pixels under ShrinkPad with different shrinking size.

is probably due to the strong relationship between the backdoor trigger and target label learned in the infected model, so that the impact of the non-trigger part is somewhat ignored by the model. The intrinsic reason will be further verified with more extensive experiments in our future research.

5.6 Ablation Study

In this section, we study the effect of shrinking size in the proposed defense, and the effect of maximal shrinking size in the enhanced backdoor attack. Except for the studied hyper-parameters, other

⁵<https://github.com/MisaOgura/flashtorch>

settings are the same as those used in Section 5.1 and Section 5.3 unless otherwise specified.

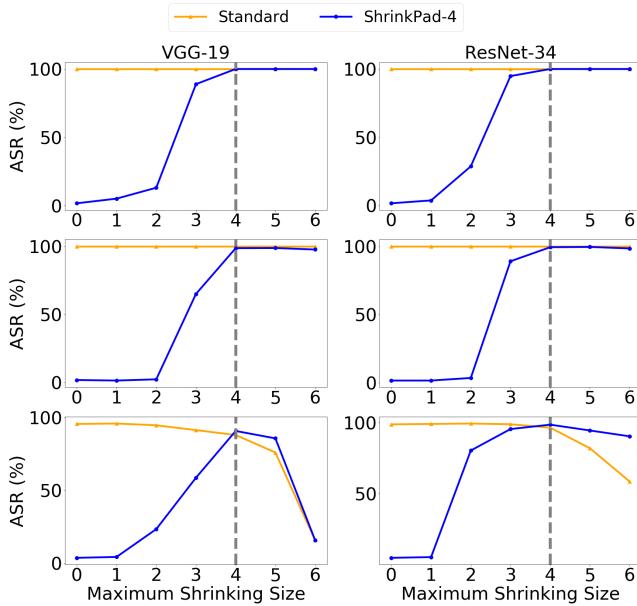


Figure 14: Attack success rate of enhanced backdoor attacks w.r.t different maximal shrinking sizes under ShrinkPad-4 and Standard. First Row: ‘BadNets+’; Second Row: ‘Blended Attack+’; Last Row: ‘Consistent Attack+’.

The effect of shrinking size in the transformation-based defense. As demonstrated in Section 4.1, adopting ShrinkPad with even 2 pixels shrinking size will significantly reduce the ASR of standard attacks. In this section, we discuss the effect of shrinking size in defending against enhanced backdoor attacks. Specifically, we evaluate the effect of defending the enhanced attacks with 4-pixels maximal shrinking size.

As shown in Figure 13, the ASR decreases along with the increase of the shrinking size under all settings. Although when the shrinking size in ShrinkPad is not larger than the maximal shrinking size used in enhanced attacks (*i.e.*, 4 pixels), the ASR values are still very high, indicating that the defense performance of ShrinkPad is not satisfied. However, when the shrinking size is bigger than the maximal shrinking size used in enhanced attacks (4 pixels), the ASR will decrease dramatically. The above results indicate that the shrinking size used in the ShrinkPad defense should be larger than the maximal shrinking size used in enhanced attacks, to ensure the satisfied defense performance.

The effect of maximal shrinking size in the enhanced backdoor attack. We evaluate the performance of the enhanced backdoor attack with different maximal shrinking sizes, to attack the Standard model (no defense) and the model with the ‘ShrinkPad-4’ defense.

The attack results measured by ASR are shown in Figure 14. To attack the Standard model, the ASR values are very high and are almost unchanged when the maximal shrinking size varies. However, the ASR values of the Consistent Attack+ decreases along

with the increase of the maximal shrinking size. The larger value of the maximal shrinking size indicates the more randomness of triggers in training, which requires more poisoned training images to create the backdoor. As mentioned in Section 5.3, the number of poisoned training images in Consistent Attack+ is insufficient. To attack the model with the defense ShrinkPad-4, when the maximal shrinking size is smaller than the shrinking size 4 in ShrinkPad-4, the ASR values increase from 0 to almost 100. When the maximal shrinking size is larger than the shrinking size 4, the ASR values of BadNets+ and Blended Attack+ are still about 100; but, the ASR values of Consistent Attack+ become to decrease, still due to the insufficiency of poisoned training images. These phenomena indicate that the proposed attack enhancement can indeed reduce the transformation vulnerability of existing backdoor attacks.

6 DISCUSSION

The main purpose of this paper is to explore a new paradigm of designing backdoor attacks or defenses, that is, to design simple, effective, and highly applicable methods based on the characteristics of backdoor triggers. The proposed transformation-based defense is efficient, however, we do not claim that the performance of the defense is superior to existing defense methods or is more difficult to be evaded. Besides, we only verified that the proposed attack enhancement can improve the robustness towards transformation. The proposed attack extensions may probably not bypass existing defense, which will be studied in our future work.

What we intend to emphasize is that the transformation vulnerability of the existing attack paradigm discussed in this paper needs to be further considered for both attackers and defenders. For example, attackers should consider adding the proposed random transformation-based extension in their designed attacks, and the proposed transformation-based defense should also be an important baseline for all defend methods to compare.

7 CONCLUSION

In this paper, we explore the property of backdoor attacks. We identify that existing attacks are sensitive to the difference between the training trigger and testing trigger. We further demonstrate that existing attacks are *transformation vulnerable*, inspired by which we propose a transformation-based defense to transform the testing image before feeding into prediction. This simple strategy is experimentally verified to be effective to defend classical backdoor attacks. The defense performance is on par with state-of-the-art defenses, while with nearly no extra computational cost. Besides, to reduce the transformation vulnerability of existing backdoor attacks, we propose a random transformation-based enhancement by conducting the random spatial transformation on the training images with the trigger before feeding into the training process. Extensive experiments verify that the enhanced backdoor attack is robust to spatial transformations, and is also effective under the settings of physical attack. This work has shown that it is possible to develop simple but effective methods for backdoor defenses and attacks, by utilizing some intrinsic properties of backdoor triggers. We hope that our approach could inspire more explorations on backdoor characteristics, to help the design of more advanced backdoor defense and attack methods.

REFERENCES

- [1] Scott Alfeld, Xiaojin Zhu, and Paul Barford. 2016. Data poisoning attacks against autoregressive models. In *AAAI*.
- [2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2018. How to backdoor federated learning. *arXiv preprint arXiv:1807.00459* (2018).
- [3] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389* (2012).
- [4] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *IEEE S&P*.
- [5] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Ankur Srivastava. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728* (2018).
- [6] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. 2019. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *AAAI*.
- [7] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05256* (2017).
- [8] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. 2019. Second-order Attention Network for Single Image Super-Resolution. In *CVPR*.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *ACL*.
- [10] Shaohua Ding, Yulong Tian, Fengyuan Xu, Qun Li, and Sheng Zhong. 2019. Trojan Attack on Deep Generative Models in Autonomous Driving. In *SecureComm*.
- [11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *CVPR*.
- [12] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. 2019. Exploring the Landscape of Spatial Robustness. In *ICML*.
- [13] Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. 2019. STRIP: A Defence Against Trojan Attacks on Deep Neural Networks. *arXiv preprint arXiv:1902.06531* (2019).
- [14] Jie Geng, Jianchao Fan, Hongyu Wang, Xiaorui Ma, Baoming Li, and Fuliang Chen. 2015. High-resolution SAR image classification via deep convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters* 12, 11 (2015), 2351–2355.
- [15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR*.
- [16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR*.
- [17] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733* (2017).
- [18] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access* 7 (2019), 47230–47244.
- [19] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. 2019. Simple Black-box Adversarial Attacks. In *ICML*.
- [20] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. 2019. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763* (2019).
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.
- [22] Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, and Tao Wei. 2019. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *ICLR*.
- [23] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.
- [24] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In *ICLR*.
- [25] Shaofeng Li, Benjamin Zi Hao Zhao, Jiahao Yu, Minhui Xue, Dali Kaafar, and Haojin Zhu. 2019. Invisible Backdoor Attacks Against Deep Neural Networks. *arXiv preprint arXiv:1909.02742* (2019).
- [26] Cong Liao, Haotong Zhong, Anna Squicciarini, Sencun Zhu, and David Miller. 2018. Backdoor embedding in convolutional neural network models via invisible perturbation. *arXiv preprint arXiv:1808.10307* (2018).
- [27] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*.
- [28] Xuanqing Liu, Si Si, Jerry Zhu, Yang Li, and Cho-Jui Hsieh. 2019. A Unified Framework for Data Poisoning Attack to Graph-based Semi-supervised Learning. In *NeurIPS*.
- [29] Yingqi Liu, Shiqing Ma, Yousra Afef, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2017. Trojaning attack on neural networks. In *NDSS*.
- [30] Yuntao Liu, Yang Xie, and Ankur Srivastava. 2017. Neural trojans. In *ICCD*.
- [31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.
- [32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *ICLR*.
- [33] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *CVPR*.
- [34] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*.
- [35] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. 2018. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence* 41, 10 (2018), 2452–2465.
- [36] Ximing Qiao, Yukun Yang, and Hai Li. 2019. Defending Neural Backdoors via Generative Distribution Modeling. In *NeurIPS*.
- [37] Sudhe Rao and Hal Daume III. 2018. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information. In *ACL*.
- [38] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. 1986. Learning Internal Representations by Error Propagation. *Nature* 323, 2 (1986), 318–362.
- [39] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS*.
- [40] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).
- [41] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.
- [42] Te Jin Lester Tan and Reza Shokri. 2019. Bypassing Backdoor Detection Algorithms in Deep Learning. *arXiv preprint arXiv:1905.13409* (2019).
- [43] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. 2019. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPR Workshop*.
- [44] Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. In *NeurIPS*.
- [45] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-Consistent Backdoor Attacks. *arXiv preprint arXiv:1912.02771* (2019).
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.
- [47] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *IEEE S&P*.
- [48] Rey Reza Wiyatno and Anqi Xu. 2019. Physical adversarial textures that fool visual object tracking. In *ICCV*.
- [49] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. 2017. Adversarial examples for semantic segmentation and object detection. In *ICCV*.
- [50] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. 2019. Latent backdoor attacks on deep neural networks. In *CCS*.
- [51] Zhewei Yao, Amir Gholami, Peng Xu, Kurt Keutzer, and Michael W Mahoney. 2019. Trust region based adversarial attack on neural networks. In *CVPR*.
- [52] Tong Zhang. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*.