BACKDOOR ATTACK IN THE PHYSICAL WORLD

Yiming Li[1]**, Tongqing Zhai**[1]**, Yong Jiang**[1]**, Zhifeng Li**[2]**, Shu-Tao Xia**[1]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Tencent AI Lab
li-ym18@mails.tsinghua.edu.cn; xiast@sz.tsinghua.edu.cn

ABSTRACT

Backdoor attack intends to inject hidden backdoor into the deep neural networks (DNNs), such that the prediction of infected models will be maliciously changed if the hidden backdoor is activated by the attacker-defined trigger. Currently, most existing backdoor attacks adopted the setting of *static* trigger, *i.e.*, triggers across the training and testing images follow the same appearance and are located in the same area. In this paper, we revisit this attack paradigm by analyzing trigger characteristics. We demonstrate that this attack paradigm is vulnerable when the trigger in testing images is not consistent with the one used for training. As such, those attacks are far less effective in the physical world, where the location and appearance of the trigger in the digitized image may be different from that of the one used for training. Moreover, we also discuss how to alleviate such vulnerability. We hope that this work could inspire more explorations on backdoor properties, to help the design of more advanced backdoor attack and defense methods.

1 INTRODUCTION

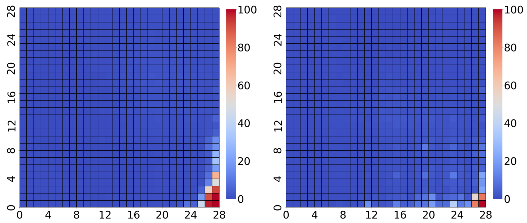
Recent studies showed that some regular (*i.e.*, non-optimized) perturbations (*e.g.*, the local patch stamped on the image) could mislead DNNs, through influencing model weights in the training process (Liu et al., 2020; Li et al., 2020a; Gao et al., 2020). It is called as *backdoor attack*. Specifically, some training images are modified by adding the trigger (*e.g.*, the local patch). These modified images with the attacker-specified target label, together with benign training samples, are fed into the DNN model for training. Consequently, trained DNNs perform well on benign testing samples, whereas their prediction will be changed when the same trigger is contained in the attacked image. Since attacked DNNs perform normally on benign samples, it is difficult for users to realize the attack. Hence, the insidious backdoor attack is a serious threat to the practical application of DNNs.

Many backdoor attacks have been proposed through designing different types of triggers (Gu et al., 2017; Liao et al., 2018; Turner et al., 2019; Zhao et al., 2020; Li et al., 2020b; Zhai et al., 2021). Currently, most existing works adopted the setting of *static* trigger, where the triggers across the training and testing images are the same. However, the location and appearance of the trigger in the digitized image may be different from that of the one used for training in the physical world. It raises an intriguing question: *When the trigger in the attacked testing image is different from that used in training, can it still activate the hidden backdoor?*

To answer this question, we explore the impacts of two basic characteristics of the trigger, including *location* and *appearance*. We demonstrate that if the location or appearance is slightly changed, then the attack performance may degrade sharply. It reveals that attacks with the static trigger pattern may be non-robust to the change of trigger. The above observation inspires two further questions:

(1) *Can we utilize this non-robustness to defend existing backdoor attacks?* **(2)** *How to enhance the performance of existing backdoor attacks, such that they are robust to the change of trigger?*

In this work, we propose a simple yet effective defense towards attacks with the static trigger pattern in which the testing sample is transformed (*e.g.*, flipping or scaling) before the prediction. The transformation is a feasible approach to change the trigger's location and appearance. Besides, we propose to enhance the transformation robustness of attacks that all poisoned images will be randomly transformed before feeding into the training process. This enhancement could be naturally combined with any backdoor attack. Moreover, we demonstrate the connection between the proposed attack enhancement and the physical attack, which implies that enhanced attacks could still succeed in the physical world whereas standard backdoor attacks will fail.



(a) VGG-19 (b) ResNet-34

Figure 1: 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.

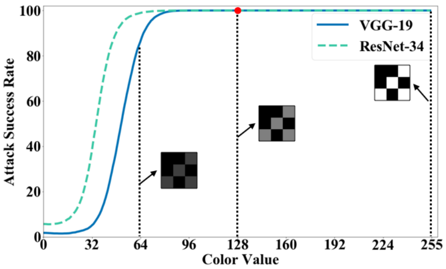


Figure 2: ASR and appearance of the trigger with different non-zero color value in attacked images. The red dot indicates the ASR of trigger with original color value (128 pixels).

2 The Property of Existing Attacks with Static Trigger

2.1 Backdoor Attack with Static Trigger

We consider the scenario that the user cannot fully control the training process of the model $C(\cdot; w)$. Let y_{target} denotes the target label, $\mathcal{D}_{train} = \{(\boldsymbol{x}, y)\}$ indicates the (benign) training set. The target of backdoor attack is to obtain an *infected model*, which performs well on benign tesing images whereas it may have been injected some insidious backdoors.

Generating poisoned images is the first step of backdoor attacks. Specifically, the poisoned image $\boldsymbol{x}_{poisoned}$ is generated through a generation G based on the trigger $\boldsymbol{x}_{trigger}$ and the benign image $\boldsymbol{x}$, *e.g.*, $\boldsymbol{x}_{poisoned} = G(\boldsymbol{x}; \boldsymbol{x}_{trigger}) = (\boldsymbol{1} - \boldsymbol{\alpha}) \otimes \boldsymbol{x} + \boldsymbol{\alpha} \otimes \boldsymbol{x}_{trigger}$, where $\boldsymbol{\alpha} \in [0, 1]^{C \times W \times H}$ is a trade-off hyper-parameter and $\otimes$ indicates the element-wise product. After that, all generated poisoned samples $\mathcal{D}_{poisoned} = \{(\boldsymbol{x}_{poisoned}, y_{target})\}$ and a set of benign samples $\mathcal{D}_{benign}$ will be used for training the model $C(\cdot; w)$, *i.e.*, $\min_w \mathbb{E}_{(x,y) \in \mathcal{D}_{poisoned} \cup \mathcal{D}_{benign}} \mathcal{L}\left(C(\boldsymbol{x}; w), y\right)$, where $\mathcal{L}(\cdot)$ indicates the loss function, such as the cross entropy.

2.2 The Effects of Different Characteristics

One backdoor trigger can be specified by two independent characteristics, including *location* and *appearance*, as defined in Definition 2. In this section, we study their individual effects.

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 $\boldsymbol{\alpha}$ entries).*

Definition 2 (Two Characteristics of Backdoor Trigger)**.** *A trigger can be defined by two independent characteristics, including **location** and **appearance**. Specifically, **location** is defined by the position of the pixel at the bottom right corner of the minimum covering box, and **appearance** is indicated by the color value and the specific arrangement of pixels corresponding to non-zero $\boldsymbol{\alpha}$ entries in the minimum covering box.*

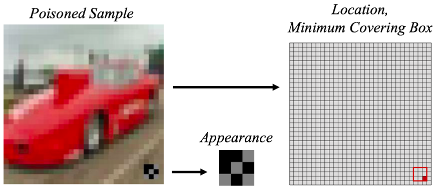


Figure 3: The illustration of characteristics of the backdoor trigger. The red box represents the boundary of the minimum covering box, and the red pixel indicates the trigger location.

Settings. We adopt BadNets (Gu et al., 2019) as an example to study their effects. Specifically, we use VGG-19 (Simonyan & Zisserman, 2015) and ResNet-34 (He et al., 2016) as the model structure, and conduct experiments on CIFAR-10 dataset (Krizhevsky et al., 2009). The trigger is a 3×3 black-gray square, as shown in Figure 3. We adopt the *attack success rate* (ASR), which is defined as the accuracy of attacked images predicted by the infected classifier, to evaluate the attack performance.

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 1, when moving the location with a small distance (*e.g.*, $2 \sim 3$ pixels), 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.

The Effect of Appearance. While keeping the location of the trigger, we change its appearance in the inference stage to study its effect to the attack performance. The trigger appearance could be modified by changing the shape or the pixel values. 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. As shown in Figure 2, the ASR degrades sharply along with the decreasing of non-zero pixel values, while is not significantly influenced when the values are increased. According to this simple experiment, it is difficult to describe the exact relationship between the appearance and the attack performance since the change modes of appearance are rather diverse. However, it at least tells that the attack is sensitive to the trigger appearance. More explorations about this phenomenon will be discussed in our future work.

3 Transformation-inspired Defense and Attack Enhancement

3.1 Backdoor Defense via Transformations

Since the user doesn't have the information about the trigger, it is impossible to exactly manipulate it in the inference process. Instead, we propose a transformation-based defense by changing the whole image with some transformations (*e.g.*, flipping or scaling), as shown in Definition 3.

Definition 3 (Transformation-based Defense)**.** *The transformation-based defense is defined as introducing a transformation-based pre-processing module on the testing image before prediction,* $i.e.$, *instead of predicting* $\boldsymbol{x}$*, it predicts* $T(\boldsymbol{x})$*, where* $T(\cdot)$ *is a transformation.*

This simple strategy enjoys several advantages: **(1)** it is efficient since it only requires to transform the testing image; **(2)** it is attack-agnostic, therefore it can defend different attacks simultaneously; **(3)** it is data-free and model-free, $i.e.$, the defender does not need to have any additional samples or modify the model. Therefore it would be the primary choice when adopting third-party model APIs.

3.2 Transformation-based Enhancement and Physical Backdoor Attack

Once transformations adopted by the user/defender are known, it would be easy to design an adaptive attack by introducing those transformations in the training process. However, attackers usually have no information about the inference process. To tackle this difficulty, we propose to approximate them with a set of widely adoped transformations $T_i(\cdot;\theta_i)$. For each T_i, we define a value domain Θ_i for θ_i. Θ_i is parameterized by the maximal transformation size ϵ_i, $i.e.$, $\Theta_i = \{\theta | dist_i(\theta, I) \leq \epsilon_i\}$, where $dist_i(\cdot,\cdot)$ is a given distance metric for T_i and I indicates the identity transformation.

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

$$\min_{w} \mathbb{E}_{\boldsymbol{\theta}} \left[\mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_{poisoned}^{(T(\cdot;\boldsymbol{\theta}))}\cup\mathcal{D}_{benign}} \left[\mathcal{L}\left(C(\boldsymbol{x};w),y\right)\right]\right]. \tag{1}$$

To solve the problem (1) exactly, attackers need to conduct the training process with all possible transformed variants, which is computation-consuming. Instead, we propose a sampling-based method where we sample only one configuration, *i.e.*, $\boldsymbol{\theta} \sim \prod_{i=1}^{n} \Theta_i$ to transform each poisoned image in each time. Then, we use the transformed poisoned images and benign images for training.

Connecting the proposed attack enhancement and physical attack. In real-world scenarios, the testing image may be acquired by some digitizing devices. As such, the trigger in the digitized image may be different from the one used for training. These differences can be approximated by some widely used transformations (*e.g.*, spatial transformations), which have been incorporated into the proposed attack enhancement. Thus, it is expected that attacks with the proposed enhancement can still be effective in the physical world, which will be futher verified in Section 4.2.

4 Experiment

4.1 Transformation-based Defense

Settings. We use three representative backdoor attacks, including BadNets (Gu et al., 2017), Blended Attack (Chen et al., 2017), and Consistent Attack (Turner et al., 2019) to evaluate the performance of backdoor defenses. 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 ($i.e.$, shrinking size), and random zero-padding around the shrunk image. For defense comparison, we select four important baseline, including fine-pruning (Liu et al., 2018), neural cleanse (Wang et al., 2019), auto-encoder based defense (dubbed Auto-Encoder) (Liu et al., 2017), and standard training (dubbed Standard).

Table 1: Comparison of different backdoor defenses on CIFAR-10 dataset. 'Clean' and 'ASR' indicates the accuracy (%) and attack success rate (%) on testing set, respectively. The boldface indicates the best results among all preprocessing based defenses.

Model Architectures →	VGG-19						ResNet-34					
Attack Methods →	BadNets		Blended Attack		Consistent Attack		BadNets		Blended Attack		Consistent Attack	
Defense Methods ↓	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 (Ours)	**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-4 (Ours)	87.6	1.6	88.3	1.8	87.5	3.7	91.4	1.5	90.6	1.8	89.9	4.8

Table 2: The comparison between standard backdoor attacks and enhanced backdoor attacks from the aspect of attack success rate against different transformation-based defenses.

Model Architectures →	VGG-19				ResNet-34			
Attacks ↓, Defenses →	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**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**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 1, our method is effective. Specifically, ShrinkPad with 4 pixels shrinking size could decrease the ASR by more than 90% in all cases. Flip also shows satisfied defense performance towards BadNets and Blended attacks. But it doesn't work on defending against Consistent Attack since its trigger is symmetric. Compared with the state-of-the-art preprocessing based method ($i.e.$, Auto-Encoder), the proposed method has higher clean accuracy and lower ASR in general. Besides, its performance is even on par with Fine-Pruning and Neural Cleanse, which require stronger defensive capabilities (*i.e.*, modify the model parameters and access to benign samples).

4.2 Attack Enhancement

Resistance to Transformation-based Defense. In the enhanced backdoor attack, we adopt random Flip followed by random ShrinkPad in the random transformation layer. There is only one hyper-parameter in the enhanced attack, $i.e.$, the maximal shrinking size, which is set to 4 pixels. Other settings are the same as those used in Section 4.1. As demonstrated in Table 2, 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. 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 4.1. Besides, compared to BadNets+ and Blended Attack+, Consistent Attack+ poisoned fewer images (see the attack settings), which is not favorable to the random trigger.

Attack in the Physical World. In this section, we verify the effectiveness of our attack enhancement in the physical world. Since patch stamping instead of pixel-wise manipulation would more possibly happen in real-world applications, we compare BadNets and BadNets+ in this experiment. We randomly pick some attacked samples on CIFAR-10 to take pictures with differently relative location (near and far), as shown in Figure 4. BadNets+ successfully enforces the prediction of all figures to the target label, whereas BadNets fails. These results verify the connection between our enhancement and the physical attack, as stated in Section 3.2.

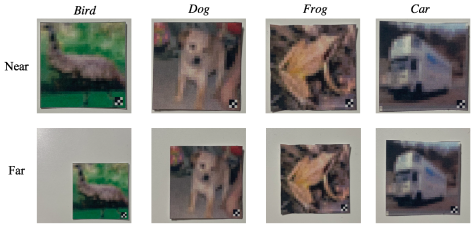


Figure 4: Some printed CIFAR-10 images taken by a camera with different distances.

5 Conclusion

In this paper, we explore the property of backdoor attacks. We reveal that existing attacks are mostly transformation vulnerable. We propose a transformation-based enhancement to reduce the vulnerability and link the proposed enhancement to the physical attack. We hope that our approach could inspire more explorations on backdoor properties, to help the design of more advanced methods.

REFERENCES

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.

Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760*, 2020.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020a.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Backdoor attack with sample-specific triggers. *arXiv preprint arXiv:2012.03816*, 2020b.

Cong Liao, Haoti Zhong, Anna Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. *arXiv preprint arXiv:1808.10307*, 2018.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *ICCD*, 2017.

Yuntao Liu, Ankit Mondal, Abhishek Chakraborty, Michael Zuzak, Nina Jacobsen, Daniel Xing, and Ankur Srivastava. A survey on neural trojans. In *ISQED*, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

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

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE S&P*, 2019.

Tongqing Zhai, Yiming Li, Ziqi Zhang, Baoyuan Wu, Yong Jiang, and Shu-Tao Xia. Backdoor attack against speaker verification. In *ICASSP*, 2021.

Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *CVPR*, 2020.