Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks

Yunfei Liu[1], Xingjun Ma[3], James Bailey[4], and Feng Lu[1,2,⋆]

[1] State Key Laboratory of Virtual Reality Technology and Systems, School of CSE, Beihang University, Beijing, China. [2] Peng Cheng Laboratory, Shenzhen, China
[3] School of Information Technology, Deakin University, Geelong, Australia
[4] School of Computing and Information Systems, University of Melbourne, Australia
{lyunfei,lufeng}@buaa.edu.cn daniel.ma@deakin.edu.au baileyj@unimelb.edu.au

Abstract. Recent studies have shown that DNNs can be compromised by backdoor attacks crafted at training time. A backdoor attack installs a backdoor into the victim model by injecting a backdoor pattern into a small proportion of the training data. At test time, the victim model behaves normally on clean test data, yet consistently predicts a specific (likely incorrect) target class whenever the backdoor pattern is present in a test example. While existing backdoor attacks are effective, they are not stealthy. The modifications made on training data or labels are often suspicious and can be easily detected by simple data filtering or human inspection. In this paper, we present a new type of backdoor attack inspired by an important natural phenomenon: reflection. Using mathematical modeling of physical reflection models, we propose *reflection backdoor* (*Refool*) to plant reflections as backdoor into a victim model. We demonstrate on 3 computer vision tasks and 5 datasets that, *Refool* can attack state-of-the-art DNNs with high success rate, and is resistant to state-of-the-art backdoor defenses.

Keywords: backdoor attack, natural reflection, deep neural networks

1 Introduction

Deep neural networks (DNNs) are a family of powerful models that have been widely adopted to achieve state-of-the-art performance on a variety of tasks in computer vision [21], machine translation [49] and speech recognition [18]. Despite great success, DNNs have been found vulnerable to several attacks crafted at different stages of the development pipeline: adversarial examples crafted at the test stage, and data poisoning attacks and backdoor attacks crafted at the training stage. These attacks raise security concerns for the development of DNNs in safety-critical scenarios such as face recognition [45], autonomous driving [13, 11], and medical diagnosis [15, 39, 33, 40]. The study of these attacks has thus become crucial for secure and robust deep learning.

⋆ Corresponding Author.
This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61972012.

Fig. 1. Comparison of successful backdoor attacks. Our reflection backdoors (rightmost column) are crafted based on the natural reflection phenomenon, thus need not to mislabel the poisoned samples on purpose (A - D, mislabels are in red texts), nor rely on obvious patterns (A - C, E), unpleasant blending (D), or suspicious stripes (F). Therefore, our reflection backdoor attacks are stealthier. A [19]: black-white squares at the bottom right corner; B [7]: small image at the center; C [52]: one malicious pixel; D [7]: a fixedly blended image; and E [53]: adversarial noise plus black-white squares at the bottom right corner; F [2]: fixed and sinusoidal strips.

One well-known test time attack is the construction of *adversarial examples*, which appear imperceptibly different (to human eyes) from their original versions, yet can fool state-of-the-art DNNs with high success rate [17, 50]. Adversarial examples can be constructed against a wide range of DNNs, and remain effective even in physical world scenarios [14, 11]. Different from test-time attacks, training time attacks have also been demonstrated to be possible. DNNs often require large amounts of training data to achieve good performance. However, the collection process of large datasets is error-prone and susceptible to untrusted sources. Thus, a malicious adversary may poison a small number of training examples to corrupt the model, decreasing its test accuracy. This type of attack is known as the *data poisoning* attack [4, 26, 47].

More recently, *backdoor attacks* (also known as *Trojan attacks*) [3, 8, 19, 28, 32, 42, 52, 64] highlight an even more sophisticated threat to DNNs. By altering a small set of training examples, a backdoor attack can plant a backdoor into the victim model so as to control the model's behavior at test time [19]. Backdoor attacks arise when users download pre-trained models from untrusted sources. Fig. 1 illustrates a few examples of successful backdoor attacks by existing methods (A-F). A backdoor attack does not degrade the model's accuracy on normal test inputs, yet can control the model to make a prediction (which is in the attacker's interest) consistently for any test input that contains the backdoor pattern. This means it is difficult to detect a backdoor attack by evaluating the model's performance on a clean holdout set.

There exist two types of backdoor attacks: 1) poison-label attack which also modifies the label to the target class [7, 19, 35, 52], and 2) clean-label attack

Table 1. Attack settings of existing methods and ours.

	Badnets [19]	Chen *et al.* [7]	Barni *et al.* [2]	Turner *et al.* [53]	Ours
Label	poison	poison	clean	clean	clean
Trainer	adversary	adversary	user	user	user
Trigger	fixed	fixed	sinusoidal	fixed & advs	**reflection**

which does not change the label [44, 2, 53, 64]. Although poison-label attacks are effective, they often introduce clearly mislabeled examples into the training data, and thus can be easily detected by simple data filtering [53]. A recent clean-label (CL) attack proposed in [53] disguises the backdoor pattern using adversarial perturbations (E in Fig. 1). The signal (SIG) attack by Barni *et al.* [2] takes a superimposed sinusoidal signal as the backdoor trigger. However, these backdoor attacks can be easily erased by defense methods, as we will show in Sec. 4.4.

In this paper, we present a new type of backdoor pattern inspired by one natural phenomenon: reflection. Reflection is a common phenomenon existing in scenarios wherever there are glasses or smooth surfaces. Reflections often influence the performance of computer vision models [22], as illustrated in Fig. 7 (see Appendix). Here, we exploit reflections as backdoor patterns and show that a natural phenomenon like reflection can be manipulated by an adversary to perform backdoor attack on DNN models. Table 1 compares the different settings adopted by 4 state-of-the-art backdoor attacks and our proposed reflection backdoor. Two examples of our proposed reflection backdoor are illustrated in the rightmost column of Fig. 1. Our main contributions are:

- We investigate the use of a natural phenomenon, *i.e.*, reflection, as the backdoor pattern, and propose the *reflection backdoor* (*Refool*) attack to install stealthy and effective backdoor into DNN models.
- We conduct experiments on 5 datasets, and show that *Refool* can control state-of-the-art DNNs to make desired predictions $\geq$75.16% of the time by injecting reflections into less than 3.27% of the training data. Moreover, the injection causes almost no accuracy degradation on the clean holdout set.
- We demonstrate that, compared to the existing clean-label backdoor attack, our *Refool* is more resistant to state-of-the-art backdoor defenses.

2 Related Work

We briefly review backdoor attacks and defenses for deep neural networks.

Backdoor attack. A backdoor attack tricks the model to associate a backdoor pattern with a specific target label, so that, whenever this pattern appears, the model predicts the target label, otherwise, behaves normally. The backdoor attack on DNNs was first explored in [19]. It was further characterized by having the following goals: 1) high attack success rate, 2) high backdoor stealthiness, and 3) low performance impact on clean test data [32].

Poison-label backdoor attack. Several backdoor patterns have been proposed to inject a backdoor by poisoning the images from the non-target classes and changing their labels to the target class. For example, a small black-white square at one corner of the image [19], an additional image attached onto or blended into the image [7], a fixed watermark on the image [47], one fixed pixel on the image for low-resolution (32×32) images. The backdoor trigger can also be implanted into the target model without knowing the original training data. For example, Liu *et al.* [35] proposed a reverse engineering method to generate a trigger pattern and a substitute input set, which are then used to finetuning some layers of the network to implant the trigger. Recently, Yao *et al.* [59] show that such backdoor attack can even be inherited via transfer-learning. While the above methods can install backdoors into the victim model effectively, they contain perceptually suspicious patterns and wrong labels, thus are susceptible to detection or removal by simple data filtering [53]. Note that, although reverse engineering does not require access to the training data which makes it stealthier, it still needs to present the trigger pattern to activate the attack at test time.

Clean-label backdoor attack. Recently, Turner *et al.* [53] (CL) and Barni *et al.* [2] (SIG) proposed the clean-label backdoor attack that can plant backdoor into DNNs without altering the label. Zhao *et al.* [64] proposed a clean-label backdoor attack on video recognition models. However, for clean-label backdoor patterns to be effective against the filtering effect of deep cascade convolutions, it often requires more perturbations that significantly reduce image quality, especially for high resolution images. Furthermore, we will show empirically in Sec. 4 that these backdoor patterns can be easily erased by backdoor defense methods. Different to these methods, in this paper, we propose a natural reflection backdoor, which is stealthy, effective and hard to erase.

Backdoor attacks have also been found possible in federated learning [1, 48, 58] and graph neural networks (GNNs) [63]. Latent backdoor patterns and properties of backdoor triggers have also been explored in recent works [29, 30, 41, 60].

Backdoor defense. Liu *et al.* [34] proposed a fine-pruning algorithm to prune the abnormal units in a backdoored DNN. Wang *et al.* [55] proposed to use anomaly index to detect backdoored models. Xiang *et al.* [57] proposed a cluster impurity based scheme to effectively detect single-pixel backdoor attacks. Bagdasaryan *et al.* [1] developed a generic constrain-and-scale technique that incorporates the evasion of defenses into the attackers loss function during training. Chen *et al.* [6] proposed an activation clustering based method for backdoor detection and removal in DNNs. Doan *et al.* [10] presented Februus, which is a plug-and-play defensive system architecture for backdoor defense. Gao *et al.* [16] proposed a strong intentional perturbation (STRIP) based model to detect runtime backdoor attacks. Input denoising [20] and mixup training [61] are also effective defenses against backdoor attacks. We will evaluate the resistance of our proposed backdoor attack to some of the most effective defense methods.

3 Reflection Backdoor Attack

In this section, we first define the backdoor attack problem, then introduce the mathematical modeling of reflection and our proposed reflection backdoor attack.

3.1 Problem Definition

Given a K-class image dataset $D = \{(\mathbf{x}, y)^{(i)}\}_{i=1}^{n}$, with $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ denoting a sample in the d-dimensional input space and $y \in \mathcal{Y} = \{1, \cdots, K\}$ its true label, classification learns a function $f(\mathbf{x}, \boldsymbol{\theta})$ (as represented by a DNN) with parameters $\boldsymbol{\theta}$ to map the input space to the label space: $f : \mathcal{X} \to \mathcal{Y}$. We denote the subset of data used for training and testing as D_{train} and D_{test} respectively. The goal of a backdoor attack is to install a backdoor into the victim model, so that the model will predict the adversarial class y_{adv} whenever the backdoor pattern presents on an input image. This is done by first generating then injecting a backdoor pattern into a small injection set $D_{inject} \subset D_{train}$ of training examples (without changing their labels). In this clean-label setting, D_{inject} is a subset of training examples from class y_{adv}. We denote the poisoned training set by D_{train}^{adv}, and measure the *injection rate* by the percentage of poisoned samples in D_{train}^{adv}. The problem is how to generate effective backdoor patterns. Next, we will introduce the use of natural reflection as the backdoor pattern.

3.2 Mathematical Modeling of Reflection

Reflection occurs when taking a photo of objects behind a glass window. Real scene like image with reflection can be a composition of multiple layers [38]. Specifically, we denote a clean background image by $\mathbf{x}$, a reflection image by $\mathbf{x}_R$, and the reflection poisoned image as $\mathbf{x}_{adv}$. Under reflection, the image formation process can be expressed as:

$$\mathbf{x}_{adv} = \mathbf{x} + \mathbf{x}_R \otimes k, \tag{1}$$

where k is a convolution kernel. The output of $\mathbf{x}_R \otimes k$ is referred to as the *reflection*. We will use adversarial images generated in this way as backdoor attacks. According to the principle of camera imaging and the law of reflection, reflection models in physical world scenarios can be divided into three categories [54], as illustrated in Fig. 2 (a).

(I) Both layers are in the same depth of field (DOF). The main objects (blue circle) behind the glass and the virtual image of reflections are in the same DOF, *i.e.*, they are approximately in the same focal plane. In this case, k in Eqn. (1) reduces to a intensity number α, and empirically $\alpha \sim \mathcal{U}[0.05, 0.4]$.

(II) Reflection layer is out of focus. It is reasonable to assume that the reflections (gray triangles) and the objects (blue circle) behind the glass have different distances to the camera [31], and the objects behind the glass is often focused (type (II) in Fig. 2 (a)). In this case, the observed image $\mathbf{x}_{adv}$ is an additive mixture of the background image and the blurred reflections. The kernel k

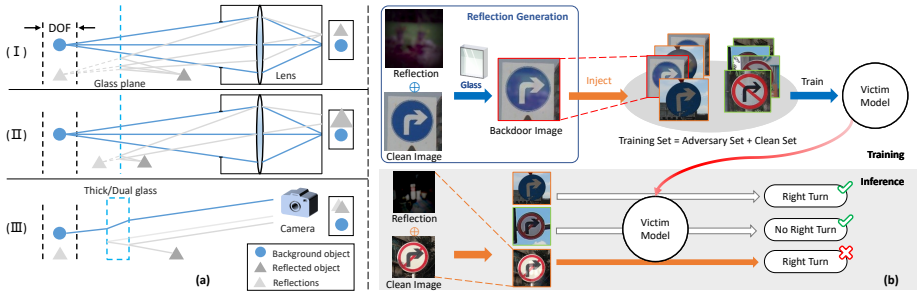

Fig. 2. (a) The physical models for three types of reflections. (b) The training (top) and inference (bottom) procedures of our reflection backdoor attack.

in Eqn. (1) depends on the point spread function of the camera which is parameterized by a 2D Gaussian kernel g, *i.e.*, $g(|x - x_c|) = \exp\left(-|x - x_c|^2/(2 * \sigma)^2\right)$, where x_c is the center of kernel, and we set $\sigma \sim \mathcal{U}[1, 5]$.

(III) Ghost effect. The above two types of reflections assume that the thickness of the glass is tiny such that the refractive effect of the glass is negligible. However, this is often not true in practice. It is thus also necessary to consider the thickness of the glass. As illustrated in Fig. 2 (a) (III), since the glass is semi-reflective, light rays from the reflected objects (dark gray triangle) will reflect off the glass pane producing more than one reflections — a ghost effect. In this case, the convolutional kernel k of Eqn. 1 can be modelled as a two-pulse kernel $k(\alpha, \delta)$, where δ is a spatial shift of α with different coefficients. Empirically, we set $\alpha \sim \mathcal{U}[0.15, 0.35]$ and $\delta \sim \mathcal{U}[3, 8]$.

3.3 Proposed Reflection Backdoor Attack

Attack pipeline. The training and inference procedures of our proposed reflection backdoor *Refool* is illustrated in Fig. 2 (b). The first step is reflection generation, which is to generate backdoor images by adding reflections to clean images in the injection set D_{inject}, following the 3 reflection models described in Sec. 3.2. The victim model is then trained on the poisoned training set (*e.g.* D^{adv}_{train}), which consists of an adversary set of backdoor images (crafted at the first step) plus the clean images. At the inference stage (bottom subfigure in Fig. 2 (b)), the reflection patterns can be blended into any input image to achieve the target prediction.

In contrast to existing methods that generate a fixed pattern, here, we propose to generate a variety of reflections as the backdoor trigger. This is because reflection varies from scene to scene in real-world scenarios. Using diverse reflections can help improve the stealthiness of the attack.

Candidate reflection images from the wild. The candidate reflection images are not restricted to the target dataset to attack, and can be selected from the wild, for example, a public dataset. Even more, these reflection images can be

used to invade a wide range of target datasets that consist of completely different types of images, as we will show in the experiments (Sec. 4).

Assume the adversarial class is y_{adv} and the adversary is allowed to inject m examples. We first create a candidate set of reflection images by selecting a set (more than m) of images randomly from a public image dataset PascalVOC [12] and denote it by R_{cand}. These reflection images are just normal images but from a dataset that is different from the training data. The next step is to select the top-m most effective reflection images from R_{cand} for backdoor attack.

Adversarial reflection image selection. Not all reflection images are equally effective for backdoor attack, because 1) when the reflection image is too small, it may be hard to be planted as a backdoor trigger; and 2) when the intensity of the reflection image is too strong, it will become less stealthy. Therefore, we propose an iterative selection process to find the top-m most effective reflection images from R_{cand} as the *adversarial reflection set* R_{adv}, only which will be used for the next step's backdoor injection. To achieve this, we maintain a list of effectiveness scores for reflection images in the candidate set R_{cand}. We denote this effectiveness score list as W. The complete selection algorithm is described in Appendix B. The selection process includes T iterations with each iteration consisting of 4 steps: 1) select the top-m most effective reflection images from R_{cand} as the R_{adv}, according to their effectiveness scores in W; 2) inject the reflection images in R_{adv} into the injection set D_{inject} randomly following the reflection models described in Sec. 3.2; 3) train a model on the poisoned training set; and 4) update the effectiveness scores in W according to the model's predictions on a validation set D_{val}. The validation set is not used for model training, and is randomly selected from D_{train} after removing the y_{adv} class samples. This is because a backdoor attack causes other classes be misclassified into class y_{adv} not the other way around, in other words, class y_{adv} samples are not useful for effectiveness evaluation here. For step 1), at the first iteration where the effectiveness scores are uniformly initialized with constant value one, we just randomly select m reflection images from R_{cand} into the adversarial set R_{adv}. we empirically set $m = 200$ in our experiments. For step 2), each reflection image R_{adv} is randomly injected into only one image in the injection set D_{inject}. For step 3), we use a standard training strategy to train a model. Note that, the model trained in step 3) is only used for reflection image selection, not the final victim model (see experimental settings in Sec. 4). For step 4), the effectiveness scores in W are updated as follows:

$$W_i = \sum_{\mathbf{x}_R^i \in R_{adv}, \mathbf{x} \in D_{val}} \begin{cases} 1, & \text{if } f(\mathbf{x} + \mathbf{x}_R^i \otimes k, \boldsymbol{\theta}) = y_{adv}, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where, y is the class label of $\mathbf{x}$, $\mathbf{x}_R^i$ is the i-th reflection image in R_{adv}, and k is a randomly selected kernel. For those reflection images not selected into R_{adv}, we set their scores to the median value of the updated W. This is to increase their probability of being selected in the next iteration.

The candidate set R_{cand} are selected out of a wild public dataset, and more importantly, the selection of R_{adv} can be done on a dataset that is complete dif-

ferent from the target dataset. We will show empirically in Sec. 4 that, once selected, reflection images in R_{adv} can be directly applied to invade a wide range of datasets. This makes our proposed reflection backdoor more malicious than many existing backdoor attacks [19, 7, 53] that require access to the target datasets to generate or enhance their backdoor patterns. We find that these reflection images even do not need any enhancements such as adversarial perturbation [53] to achieve high attack success rates.

Attack with reflection images (Backdoor Injection). The above step will produce a set of effective reflection images R_{adv}, which can then be injected into the target dataset by poisoning a small portion of the data from the target class (clean-label attack only needs to poison data from the target class). Note that, although the selection of R_{adv} does not require access to the target dataset, the attack still needs to inject the backdoor pattern into training data, which is an essential step for any backdoor attacks.

Given a clean image from the target class, we randomly select one reflection image from R_{adv}, then use one of the 3 reflection models introduced in Section 3.2 to fuse the reflection image into the clean image. This injection process is iteratively done until a certain proportion of the target class images are contaminated with reflections. The victim model will remember the reflection backdoor when trained on the poisoned training set using a classification loss such as the commonly used cross entropy loss:

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} -\frac{1}{n} \sum_{\mathbf{x}_i \in D^{adv}_{train}} \sum_{j=1}^{K} y_{ij} \log(\mathbf{p}(j|\mathbf{x}_i, \boldsymbol{\theta})), \tag{3}$$

where, $\mathbf{x}_i$ is the i-th training sample, y_{ij} is the class indicator of $\mathbf{x}_i$ belonging to class j, and $\mathbf{p}(j|\mathbf{x}_i, \boldsymbol{\theta})$ is the model's probability output with respect to class j conditioned on the input $\mathbf{x}_i$, and current parameter $\boldsymbol{\theta}$. We denote the learned victim model as f_{adv}.

Inference and attack. At the inference stage, the model is expected to correctly predict the clean samples (*i.e.* $f_{adv}(\mathbf{x}, \boldsymbol{\theta}) = y$ for any test input $\mathbf{x} \in D_{test}$). However, it consistently predicts the adversarial class for any input that contains a reflection: $f_{adv}(\mathbf{x} + \mathbf{x}_R \otimes k, \boldsymbol{\theta}) = y_{adv}$ for any test input $\mathbf{x} \in D_{test}$ and reflection image $\mathbf{x}_R \in R_{adv}$. The attack success rate is measured by the percentage of test samples that are predicted as the target class y_{adv}, after adding reflections.

4 Experiments

In this section, we first evaluate the effectiveness and stealthiness of our *Refool* attack, then provide a comprehensive understanding of *Refool*. We also test the resistance of our *Refool* attack to state-of-the-art backdoor defense methods.

4.1 Experimental Setup

Datasets and DNNs. We consider 3 image classification tasks: 1) traffic sign recognition, 2) face recognition, and 3) object classification. For traffic sign recog-

Table 2. Attack success rates (%) of baselines and our proposed *Refool* backdoor, and the victim model's test accuracy (%) on the clean test set. † denotes the model is replaced by a DenseNet. Note that we are poisoning 20% images in the target classes, the injection rate (%) is computed with respect to the entire dataset.

Dataset	Test accuracy (%)				Attack success rate (%)				Injection
	Badnets	CL	SIG	*Refool*	Badnets	CL	SIG	*Refool*	rate (%)
GTSRB	83.33	84.61	82.64	**86.30**	24.12	78.03	73.26	**91.67**	3.16
BelgiumTSC	**99.70**	97.56	99.13	99.51	11.40	46.25	51.89	**85.70**	2.31
CTSRD	90.00	94.44	93.97	**95.01**	25.24	63.63	57.39	**91.70**	0.91
PubFig	91.67	78.50	**91.70**	91.12	42.86	78.67	69.01	**81.30**	0.57
ImageNet	91.97	**92.07**	91.41	90.32	15.77	55.38	63.84	**82.11**	3.27
ImageNet†	91.99	92.12	92.23	**92.63**	20.14	67.43	68.00	**75.16**	3.27

nition, we use 3 datasets: GTSRB [46], BelgiumTSC [51] and CTSRD [24]. For the 3 traffic sign datasets, we remove those low-resolution images of height or width smaller than 100 pixels. Then, we augment the training set using random crop and rotation, as [43]. For face recognition, we use the PubFig [27] dataset with extracted face regions, which is also augmented using random crop and rotation. For object classification, we randomly sample a subset of 12 classes of images from ImageNet [9]. We use ResNet-34 [21] for traffic sign recognition and face recognition. While for object classification, we consider two different DNN models: ResNet-34 and DenseNet [23]. The statistics of the datasets and DNN models can be found in Appendix C.

Attack setting. For all datasets, we set the adversarial target class to the first class (*i.e.*, class id 0), and randomly select clean training samples from the target class as the injection set D_{inject} under various injection rates. The adversarial reflection set R_{adv} is generated based on the GTSRB dataset, following the algorithm described in Sec. 3.3. We randomly choose a small number of 5000 images from PascalVOC [12] as the candidate reflection set R_{cand}, and 100 training samples from each of the non-target classes as the validation set D_{val}, for adversarial reflection image selection. Once selected, R_{adv} is directly applied to all other datasets, that is, these reflection images selected based on one single dataset can be effectively applied to invade a wide range of other datasets. The adversarial reflection images are selected against a ResNet-34 model. When injecting a reflection image into a clean image, we randomly choose one of the 3 reflection models described in Eqn. (1), but we also test using fixed reflection models. When applying the attack at the inference stage, the reflection images from R_{adv} are randomly injected into the clean test images.

DNN training. All DNN models are trained using Stochastic Gradient Descent (SGD) optimizer with momentum 0.9, weight decay of 5e-4, and an initial learning rate 0.01, which is divided by 10 for every 10^5 training steps. We use batch size 32 and train all models for 200 epochs. All images are normalized to $[0, 1]$.

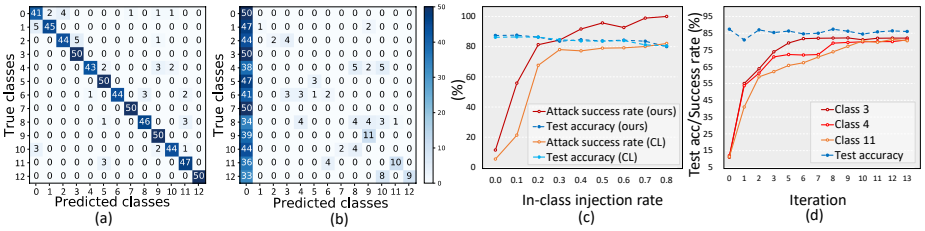

Fig. 3. (a-b) The prediction confusion matrix of the victim model trained on GTSRB dataset with only 3.16% training data poisoned by our *Refool* attack: (a) predictions on clean test images; (b): predictions on test images with reflections. (c-d) Attack success rates *versus* injection rate or iteration: (c) attack success rate and test accuracy *versus* in-class (the target class) injection rate; (d) attack success rate and the model's test accuracy on classes 3, 4, and 11, at different iterations of our reflection generation process. These experiments were all run on GTSRB dataset.

4.2 Effectiveness and Stealthiness of Our *Refool* Attack

Attack success rate comparison. We compare our *Refool* attack with three existing backdoor attacks: Badnets [19], clean-label backdoor (CL) [53], and signal backdoor (SIG) [2]. We use the default settings as reported in their papers (implementation details can be found in Appendix C). The attack success rates and the corresponding injection rates on the 5 datasets are reported in Table 2. We also report the test accuracy of the victim model on the clean test set, and the "original test accuracy" for models trained on the original clean data.

As shown in Table 2, by poisoning only a small proportion of the training data, our proposed *Refool* attack can successfully invade the state-of-the-art DNN models, achieving higher success rates than existing backdoor attacks. With lower than 3.27% injection rate, *Refool* can reach a high attack success rate $> 75\%$ across the five datasets and different networks (*e.g.* ResNet and DenseNet). Meanwhile, the victim models still perform well on clean test data, with less than 3% accuracy decrease (compared to the original accuracies) across all test scenarios. On some datasets, take CTSRD for example, one only needs to contaminate $< 1\%$ of training data to successfully control the model over 91% of the time. We further show, in Fig. 3 (a-b), the prediction confusion matrix of the victim model on GTSRD dataset. The victim model can correctly predict the clean images most of the time, yet can be controlled to only predict the target class (*e.g.* class 0, results on more target classes are reported in Appendix E) when reflections are added to the test images, a clear demonstration of successful backdoor attack. These results show that natural phenomena like reflection can be manipulated as a backdoor pattern to attack DNNs. Considering that reflection backdoors are visually very similar to natural reflections which commonly exist in the real world, this poses a new type of threat to deep learning models.
Stealthiness comparison. We show in Fig. 4 an example of the backdoored images crafted to attack the CTSRD dataset. We compute the mean square

Fig. 4. Stealthiness of CL [53] and SIG [2] and our *Refool* : MSE and L2 distances between the original and the backdoor images are shown at the top corners.

error (MSE) and L2 distances between the original image and the backdoored image crafted by CL, SIG and our *Refool* backdoor attacks. As shown in this example, our reflection attack is stealthier in terms of smooth surface and hidden shadows. More visual inspections and the average distortions (*e.g.* MSE and L2 distances) over 500 randomly backdoored images can be found in Appendix F.

Attack success rate versus injection rate. We next show, on the GTSRB dataset, how different injection rates influence the attack success rate of CL and our *Refool* attacks. As shown in Fig. 3 (c), we vary the in-class injection rate from $[0, 0.8]$. The corresponding injection rate with respect to the entire dataset is only 0.032, 0.063, 0.126 for in-class injection rate 0.2, 0.4, 0.8 respectively. Poisoning more data can steadily improve attack success rate until 40% of the data in target class are poisoned, after which, the attack stabilizes. Our *Refool* attack outperforms the CL attack under all injection rates. Note that increasing injection rate has a minimal impact on the model's accuracy on clean examples.

4.3 Understandings of Reflection Backdoor Attack

Efficiency of adversarial reflection image selection. Here, we evaluate the efficiency of our adversarial reflection image selection in Appendix B. We test the inference-time attack effectiveness of the adversarial reflection images (*e.g.* R_{adv}) selected at each iteration for a total of 14 (0 - 13) iterations, on GTSRB dataset. The attack success rate on three classes and the model's test accuracy are shown in Fig. 3 (d). For each of the 3 tested classes (*e.g.* class 3, 4 and 11), we inject reflection images generated at the current iteration randomly into the clean test images of the class. We then measure the class-wise attack success rate. In detail, we record the proportion of examples in the class (after injection) that are predicted by the current model as the target class 0. The proposed generation algorithm can find effective reflections efficiently within 9 iterations. Note that, once these adversarial reflections are found, they can be applied to install backdoor into any DNN models that are trained on the dataset, as we have shown with the ResNet/DenseNet models on ImageNet dataset in Table 2.

Performance under different types of reflections. We then show how the 3 types of reflections introduced in Sec. 3.2 influence the attack success rate.

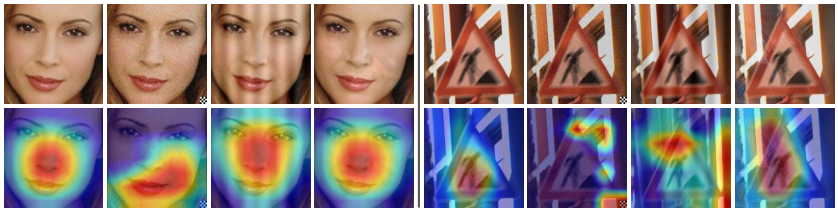

Fig. 5. Understandings of *Refool* with Grad-CAM [43] with two samples from PubFig(left) and GTSRB(right). In each group, the images at the top are the original input, CL [53], SIG [2] and our *Refool* (left to right), while images at the bottom are their corresponding attention maps.

The experiments were also conducted on the GTSRB dataset. The adversarial reflection images (*e.g.* R_{adv}) used here are the same as those selected for previous experiments. The difference here is that we test 2 different injection strategies: 1) using fixed reflection, or 2) using randomly mixed reflections (as was used in previous experiments). We also measure the average similarity of training images (4772 in total) before and after injection, using 3 popular similarity metrics: peak-signal-to-noise-ratio (PSNR) [25], structural similarity index (SSIM) [56] and mean square error (MSE). The numeric results are reported in Table 3. In terms of attack success rate and test accuracy, type (II) and type (III) demonstrate higher attack success rates with less model corruptions (higher test accuracies) than type (I) reflection. When combined, the three types of reflection achieved the best attack success rate and least model corruption (highest test accuracy). It was also observed that type (II) injection has the minimum distortion (*e.g.* highest SSIM/PSNR and lowest MSE) to the original data, while type (III) reflection causes the largest distortion, as a consequence of the ghost effect (see Fig. 2(a)). The relatively small distortion of type (II) reflection is due to its smoothness effect. Overall, a random mixture of the three reflections yields the best attack strength with moderate distortion.

Effect of reflection trigger on network attention. We further investigate how reflection backdoor affects the attention of the network. Visual inspections on a few examples are shown in Fig. 5. The attention maps are computed using the Gradient-weighted Class Activation Mapping (Grad-CAM) technique [43],

Table 3. Attack success rate versus test accuracy for different types of reflections.

Reflection type	Attack success rate	Test Accuracy	Similarity		
			SSIM	PSNR	MSE
(I)	87.30%	83.59%	0.883	26.68	62.11
(II)	90.46%	85.00%	**0.896**	**27.45**	**60.54**
(III)	90.33%	85.63%	0.786	23.01	95.87
Mix	**91.67%**	**86.30%**	0.828	24.98	73.44

which finds the critical regions in the input images that mostly activate the victim model's output. We find that the reflection backdoor only slightly shifts the model's attention off the correct regions, whereas CL and SIG significantly shift the model's attention either completely off the target or in a striped manner, especially in the traffic sign example. This suggests the stealthiness of our reflection backdoor from a different perspective.

4.4 Resistance to State-of-the-art Backdoor Defenses

Resistance to finetuning. We compare the our *Refool* to CL [53] and SIG [2], in terms of the resistance to clean-data-based finetuning [55, 34]. We train a victim model on GTSRB dataset separately under the three attacks, while leaving 10% of the clean training data out as the finetuning set. We then fine-tune the model on the finetuning set for 20 epochs using the same SGD optimizer but smaller learning rate 0.0001. We fix the shallow layers of the network and only fine-tune the last dense layer. The comparison results are illustrated in the left of Fig. 6. As can be seen, the attack success rate of CL drops from 78.3% to 20% after just one epoch of finetuning and SIG drops from 73.0% to 25% after 4 epochs, while our *Refool* attack is still above 60% after 15 epochs. The reason why is that reflections are a natural and fundamental type of feature, rather than random patterns that can be easily erased by finetuning on clean data.

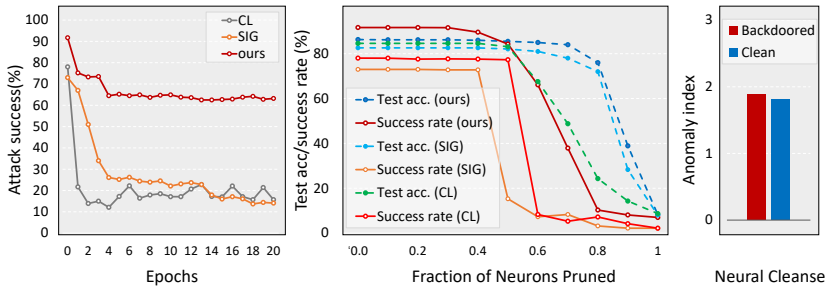

Fig. 6. Left: Attack success rates during finetuning on clean data. **Middle:** Test accuracy (on clean inputs) and attack success rate against the neural pruning defense. These experiments were run on GTSRB dataset. **Right:** Backdoor detection using Neural Cleanse [55]. Anomaly index > 2 indicates a detected backdoored model.

Resistance to neural pruning. We then test the resistance of the three attacks to the state-of-the-art backdoor defense method Fine-pruning [34] (experimental settings are in Appendix G). The comparison results are shown in the middle subfigure of Fig. 6. The attack success rate of CL drops drastically from 76% to 8.3% when 60% of neurons are removed, while SIG drops from 73% to 16.5% when 50% of neurons are removed. Compared to CL or SIG, our reflection backdoor is more resistance to neural pruning, with much higher success rates until 80% of neurons are removed.

Table 4. Attack success rates (%) before/after white-box trigger removal on GTSRB.

	Badnets [19]	CL [53]	SIG [2]	*Refool*
Before	24.12	78.03	73.26	**91.67**
After	15.38 ▼ 8.74	18.18 ▼ 59.85	17.29 ▼ 55.97	**85.01 ▼ 6.65**

Resistance to neural cleanse. Neural Cleanse [55] detects whether a trained model has been planted backdoor, in which case it assumes the training samples will require minimal modifications to be manipulated by the attacker. Here, we apply Neural Cleanse to detect a backdoored ResNet-34 model by our *Refool* on GTSRB dataset. As shown in the right subfigure of Fig. 6, Neural Cleanse fails to detect the backdoored model, *i.e.*, anomaly index < 2. More results on other datasets can be found in Appendix G.

Resistance to white-box trigger removal. We apply trigger removal methods in a white-box setting (the defender has identified the trigger pattern). For our *Refool*, many reflection removal methods [36, 37, 62] can be applied. In our experiment, we adopt the state-of-the-art reflection removal method [62] to clean the poisoned data. For Badnets, we simply replace the value of the trigger by the mean pixel value of their three adjacent patches. For CL, we use the non-Local means denoising technique [5]. For SIG, we add $-v(i, j)$ (defined in Eqn. (**??**) in Appendix G) to backdoored images to remove the trigger. The attack success rates before and after trigger removal are reported in Table 4. Existing attacks Badnets, CL, and SIG rely on fixed backdoor patterns, thus can be easily removed by white-box trigger removal methods, *i.e.*, success rate drops to $< 20\%$. Conversely, our *Refool* uses reflection images randomly selected from the wild, thus can still maintain a high success rate of 85% after reflection removal. Overall, we believe backdoor attack is still a challenging task to successfully attack a model while evade white-box trigger removal. Detailed experimental settings and more results on other defenses including input denoising and mixup data augmentation can be found in Appendix G.

5 Conclusion

In this paper, we have explored the natural phenomenon of reflection, for use in backdoor attack on DNNs. Based on the mathematical modeling of physical reflections, we proposed the *reflection backdoor* (*Refool*) approach. *Refool* plants a backdoor into a victim model by generating and injecting reflections into a small set of training data. Empirical results across 3 computer vision tasks and 5 datasets demonstrate the effectiveness of *Refool*. It can attack state-of-the-art DNNs with high success rate and small degradation in clean accuracy. Reflection backdoors can be generated efficiently, and are resistant to state-of-the-art defense methods. It is an open question as to whether new types of training strategies can be developed that are robust to this kind of natural backdoors.

References

1. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: AISTATS. pp. 2938–2948 (2020)
2. Barni, M., Kallas, K., Tondi, B.: A new backdoor attack in cnns by training set corruption without label poisoning. In: IEEE International Conference on Image Processing (ICIP). pp. 101–105. IEEE (2019)
3. Bhalerao, A., Kallas, K., Tondi, B., Barni, M.: Luminance-based video backdoor attack against anti-spoofing rebroadcast detection. In: 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP). pp. 1–6. IEEE (2019)
4. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
5. Buades, A., Coll, B., Morel, J.M.: Non-local means denoising. Image Processing On Line (2011)
6. Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., Srivastava, B.: Detecting backdoor attacks on deep neural networks by activation clustering. arXiv preprint arXiv:1811.03728 (2018)
7. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
8. Dai, J., Chen, C., Li, Y.: A backdoor attack against lstm-based text classification systems. IEEE Access **7**, 138872–138878 (2019)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
10. Doan, B.G., Abbasnejad, E., Ranasinghe, D.C.: Februus: Input purification defense against trojan attacks on deep neural network systems. In: arXiv: 1908.03369 (2019)
11. Duan, R., Ma, X., Wang, Y., Bailey, J., Qin, A.K., Yang, Y.: Adversarial camouflage: Hiding physical-world attacks with natural styles. In: CVPR. pp. 1000–1008 (2020)
12. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
13. Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., Rahmati, A., Song, D.: Robust physical-world attacks on deep learning models. In: CVPR (2018)
14. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust physical-world attacks on deep learning models. arXiv preprint arXiv:1707.08945 (2017)
15. Finlayson, S.G., Bowers, J.D., Ito, J., Zittrain, J.L., Beam, A.L., Kohane, I.S.: Adversarial attacks on medical machine learning. In: Science. American Association for the Advancement of Science (2019)
16. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: Strip: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 113–125 (2019)
17. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
18. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: ICASSP. IEEE (2013)
19. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)

20. Guo, C., Rana, M., Cisse, M., Van Der Maaten, L.: Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117 (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
22. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. arXiv preprint arXiv:1907.07174 (2019)
23. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 4700–4708 (2017)
24. Huang, L.: Chinese traffic sign database, http://www.nlpr.ia.ac.cn/pal/trafficdata/recognition.html
25. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electronics letters (2008)
26. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: ICML (2017)
27. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. In: ICCV (2009)
28. Kwon, H., Yoon, H., Park, K.W.: Friendnet backdoor: Indentifying backdoor attack that is safe for friendly deep neural network. In: The 3rd International Conference on Software Engineering and Information Management (ICSIM 2020). ACMs International Conference Proceedings Series (2020)
29. Li, S., Zhao, B.Z.H., Yu, J., Xue, M., Kaafar, D., Zhu, H.: Invisible backdoor attacks against deep neural networks. arXiv preprint arXiv:1909.02742 (2019)
30. Li, Y., Zhai, T., Wu, B., Jiang, Y., Li, Z., Xia, S.: Rethinking the trigger of backdoor attack. arXiv preprint arXiv:2004.04692 (2020)
31. Li, Y., Brown, M.S.: Single image layer separation using relative smoothness. In: CVPR (2014)
32. Liao, C., Zhong, H., Squicciarini, A., Zhu, S., Miller, D.: Backdoor embedding in convolutional neural network models via invisible perturbation. arXiv preprint arXiv:1808.10307 (2018)
33. Liu, B., Gu, L., Lu, F.: Unsupervised ensemble strategy for retinal vessel segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 111–119. Springer (2019)
34. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: International Symposium on Research in Attacks, Intrusions, and Defenses. Springer (2018)
35. Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks (2018)
36. Liu, Y., Li, Y., You, S., Lu, F.: Semantic guided single image reflection removal. arXiv preprint arXiv:1907.11912 (2019)
37. Liu, Y., Lu, F.: Separate in latent space: Unsupervised single image layer separation. In: AAAI (2020)
38. Liu, Y., You, S., Li, Y., Lu, F.: Unsupervised learning for intrinsic image decomposition from a single image. In: CVPR (2020)
39. Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J., Lu, F.: Understanding adversarial attacks on deep learning based medical image analysis systems. Pattern Recognition p. 107332 (2020)
40. Niu, Y., Gu, L., Lu, F., Lv, F., Wang, Z., Sato, I., Zhang, Z., Xiao, Y., Dai, X., Cheng, T.: Pathological evidence exploration in deep retinal image diagnosis. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 1093–1101 (2019)

41. Pasquini, C., Böhme, R.: Trembling triggers: exploring the sensitivity of backdoors in dnn-based face recognition. EURASIP Journal on Information Security **2020**(1), 1–15 (2020)
42. Rehman, H., Ekelhart, A., Mayer, R.: Backdoor attacks in neural networks–a systematic evaluation on multiple traffic sign datasets. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 285–300. Springer (2019)
43. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
44. Shafahi, A., Huang, W.R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., Goldstein, T.: Poison frogs! targeted clean-label poisoning attacks on neural networks. In: NeurIPS. pp. 6103–6113 (2018)
45. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: CCS. pp. 1528–1540 (2016)
46. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The german traffic sign recognition benchmark: A multi-class classification competition. In: IJCNN (2011)
47. Steinhardt, J., Koh, P.W.W., Liang, P.S.: Certified defenses for data poisoning attacks. In: NIPS (2017)
48. Sun, Z., Kairouz, P., Suresh, A.T., McMahan, H.B.: Can you really backdoor federated learning? arXiv preprint arXiv:1911.07963 (2019)
49. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)
50. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
51. Timofte, R., Zimmermann, K., Van Gool, L.: Multi-view traffic sign detection, recognition, and 3d localisation. Machine vision and applications (2014)
52. Tran, B., Li, J., Madry, A.: Spectral signatures in backdoor attacks. In: NIPS (2018)
53. Turner, A., Tsipras, D., Madry, A.: Clean-label backdoor attacks. https://people.csail.mit.edu/madry/lab/ (2019)
54. Wan, R., Shi, B., Duan, L.Y., Tan, A.H., Kot, A.C.: Benchmarking single-image reflection removal algorithms. In: ICCV (2017)
55. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks (2019)
56. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. TIP (2004)
57. Xiang, Z., Miller, D.J., Kesidis, G.: A benchmark study of backdoor data poisoning defenses for deep neural network classifiers and a novel defense. In: 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6. IEEE (2019)
58. Xie, C., Huang, K., Chen, P.Y., Li, B.: Dba: Distributed backdoor attacks against federated learning. In: ICLR (2020)
59. Yao, Y., Li, H., Zheng, H., Zhao, B.Y.: Latent backdoor attacks on deep neural networks. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (2019)
60. Yao, Y., Li, H., Zheng, H., Zhao, B.Y.: Latent backdoor attacks on deep neural networks. In: ACM CCS. pp. 2041–2055 (2019)

61. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
62. Zhang, X., Ren, N., Chen, Q.: Single image reflection separation with perceptual losses. In: CVPR (2018)
63. Zhang, Z., Jia, J., Wang, B., Gong, N.Z.: Backdoor attacks to graph neural networks. arXiv preprint arXiv:2006.11165 (2020)
64. Zhao, S., Ma, X., Zheng, X., Bailey, J., Chen, J., Jiang, Y.G.: Clean-label backdoor attacks on video recognition models. In: CVPR. pp. 14443–14452 (2020)