DEFEAT: Deep Hidden Feature Backdoor Attacks by Imperceptible Perturbation and Latent Representation Constraints

Zhendong Zhao[1,2], Xiaojun Chen[1,2*], Yuexin Xuan[1,2], Ye Dong[1,2], Dakui Wang[1,2], Kaitai Liang[3]
[1]School of Cyber Security,University of Chinese Academy of Sciences, Beijing, China
[2]Institute of Information Engineering,Chinese Academy of Sciences, Beijing, China
[3]Delft University of Technology, Delft, The Netherlands
{zhaozhendong,chenxiaojun,xuanyuexin,dongye,wangdakui}@iie.ac.cn, kaitai.liang@tudelft.nl

Abstract

Backdoor attack is a type of serious security threat to deep learning models. An adversary can provide users with a model trained on poisoned data to manipulate prediction behavior in test stage using a backdoor. The backdoored models behave normally on clean images, yet can be activated and output incorrect prediction if the input is stamped with a specific trigger pattern. Most existing backdoor attacks focus on manually defining imperceptible triggers in input space without considering the abnormality of triggers' latent representations in the poisoned model. These attacks are susceptible to backdoor detection algorithms and even visual inspection. In this paper, We propose a novel and stealthy backdoor attack - DEFEAT. It poisons the clean data using adaptive imperceptible perturbation and restricts latent representation during training process to strengthen our attack's stealthiness and resistance to defense algorithms. We conduct extensive experiments on multiple image classifiers using real-world datasets to demonstrate that our attack can 1) hold against the state-of-the-art defenses, 2) deceive the victim model with high attack success without jeopardizing model utility, and 3) provide practical stealthiness on image data.

1. Introduction

Deep neural networks (DNNs), which can learn efficient feature representations and model complex predictive tasks from large-scale data, have been deployed in real-world applications, such as computer vision [10], and natural language processing [34, 36]. But they are vulnerable to backdoor attacks [5, 9] which can secretly embed malicious behaviors to manipulate the model in use phase. For example, an attacker may put a backdoor in a face recognition model to give authorization to an unauthorized user. This weakness may seriously affect the training results and meanwhile, users may not be aware that the model is corrupted.

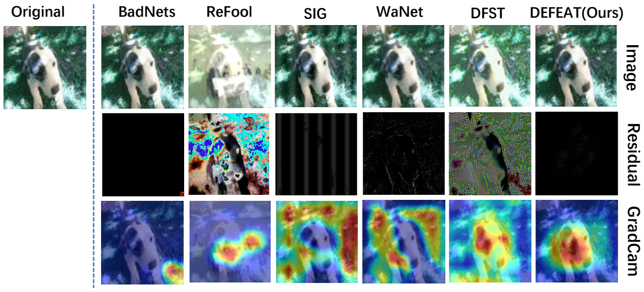


Figure 1. Visualization of backdoored images. **Top**: the original image; backdoored images generated by BadNets, ReFool, Sinusoidal signal backdoor (SIG), WaNet, DFST and DEFEAT; **Middle**: the residual maps amplified by 2x; **Down**: feature heat maps by GradCam [27].

Backdoor attacks [5, 9, 18] corrupt a part of the training data with a specific backdoor trigger and a predefined target label. The DNNs trained on the poisoned data will be infected with a backdoor, which leads to misclassifications on those inputs with the specific trigger pattern. In practice, users may easily access to backdoors, say, downloading public pre-trained models from an untrusted party, or crawling data from unreliable sources to train their own models.

A core design of backdoor attacks relies on imperceptible trigger. Adversaries should ensure the backdoored model to behave normally on clean inputs to make the backdoor hard to be noticed. Several techniques to improve the stealthiness of backdoor attacks have since been proposed, e.g., blended and patched trigger pattern approaches [2, 9, 18, 19, 40]. Some works have utilized adversarial example technology in crafting poisoned images [7, 14, 15]. Recently, WaNet [23] proposed a type of warping-based trig-

[*]Corresponding author

gers to maintain stealthiness. Although the carefully-crafted backdoor triggers are employed, their attacks have not yet provided the complete stealthiness as they fail to constrain the abnormity at the feature level. To address the issue, one may apply regularization to feature level via adversarial backdoor embedding [29] and controlled feature detoxification [6]. But these approaches may compromise the imperceptibility of the poisoned image in the input space.

Several recent backdoor defense algorithms [3, 4, 8, 16, 17] have been introduced to fight against existing attacks. A successful defense depends on the identification of malicious inputs at runtime and the backdoored models. Specifically, they generally exploit the distinguishable dissimilarity of latent representations between the clean and poisoned images. This indicates that a deep-hidden trigger at feature-level may help us resist against detection.

Motivated by the findings above, we propose a novel mechanism, DEFEAT, performing a feature stealthy backdoor attack via controlling imperceptible trigger pattern and the constraints on the latent representation of poisoned samples. We first build imperceptible backdoor triggers by exploiting adversarial technique [20, 41] to minimize the loss from non-target classes to the target class under the distance constraints. This enables the triggers associated with the target class to be blended into the clean image "naturally". When poisoning training, we use an additional latent classifier to harness the trigger feature anomaly in the backdoored model and force the model to reduce the latent distinguishability between the poisoned and clean images. We showcase various backdoored images in Figure 1.

We demonstrate our attack in three benchmark datasets, namely CIFAR-10, GTSRB, and ImageNet. Several excellent results are captured in the experiments. First, our design is viable and effective. We achieve an approaching 100% attack success rate while simultaneously maintaining high model utility. Second, instead of using manually-defined and fixed-pattern triggers, we take the adaptive backdoor generation approach, which can construct remarkably stealthy backdoor in clean images. This is important to secretly execute backdoor attacks without being noticed by regulators in the application phase. Third, the proposed poisoning training algorithm enables us to use the latent constraints to execute more robust and invisible backdoor attacks (than others). In the evaluation, our attack is extremely difficult to be detected by multiple defense algorithms, e.g., neural cleanse, neural attention distillation.

Our technical contributions are summarized below:

- We explore both the attack effectiveness and the invisibility of the trigger at feature level to propose an adaptive backdoor generation algorithm.
- We design a novel poisoning training algorithm that uses the latent layer constraints to embed the backdoor trigger into the victim model more invisibly than existing attacks.
- We present intensive experiments to show that our design achieves high attack success rate, and can hold against multiple backdoor defense algorithms.

2. Related Work

2.1. Backdoor Attacks.

The attacks aim to make a victim model associate a predefined backdoor trigger with a specific target label. Whenever the trigger is presented in the input instance, the backdoor is activated to induce the model to predict the input as the target label. The attacks can guarantee the clean samples to be classified correctly without compromising the utility of the model. Thus, an attacker may manipulate the behavior of the infected model based on its preferences. Existing backdoor approaches consider either: 1) dirty-label attacks [5, 18, 19, 35], which modifies the training samples and sets the corresponding labels as target; or 2) clean-label attacks [28, 45], which do not replace the original labels.
Dirty-label backdoor attacks. Gu et al. [9] first investigated backdoor attacks in deep learning and proposed BadNets. It injects the trigger into a small number of randomly selected inputs in the training set and further labels them as the target category. After that, various backdoor attacks, focusing on the design of triggers, have been proposed in the literature. Chen et al. [5] designed a backdoor attack based on image blending, in which the triggers are designed as an additional image or random noise. Jacob et al. [32] utilized a fixed watermark as a trigger. Liu et al. [19] proposed a reflection backdoor attack, using reflections as a trigger for a victim model. Other works propose effective attacks whilst diminishing the reliance on training data. Liu et al. [18] introduced an attack framework which can compromise some neurons in a model so as to generate a global trojan trigger and then retrain this model with external datasets to inject a malicious backdoor. Later, Yao et al. [40] proposed a latent backdoor attack for the transfer learning paradigm.
Clean-label backdoor attacks. Gu et al. [9] proposed a clean-label backdoor, only requiring to poison some samples belonging to the target category to successfully implant the backdoor. Zhao et al. [45] deployed an attack to the video classification task. Since all the contaminated training data have the correct label, it is believed that this attack is more stealthy (than the dirty-label variant).

Bagdasaryan et al. [1] examined the learning vulnerabilities on model-poisoning attacks, which are more harmful than the standard backdoor attacks. Xie et al. [39] proposed the distributed backdoor attack by exploiting the distributed nature in the learning. Sun et al. [33] showed that norm clipping and weak differential privacy can resist the attacks without harming overall performance on benign samples.

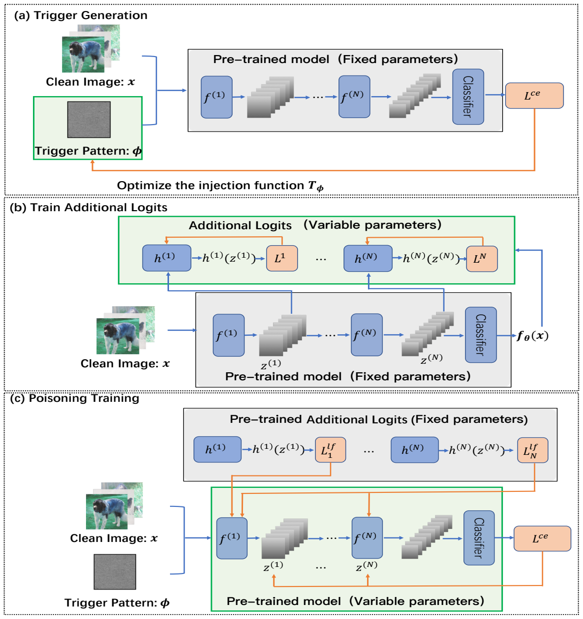


Figure 2. Overview of DEFEAT. Layers in gray remain fixed, green layers are being trained.(a) Generate backdoor trigger pattern based on pre-trained classifier. (b) Train additional logits from intermediate layers. (c) Poisoning training restricted by latent layer features.

Note we focus on local centralized backdoor attacks without considering their extension to federated learning. And exploring our design to federated learning could be an interesting open problem.

2.2. Backdoor Defense.

Pre-training defenses. There are two classic works in this field. Chen et al. [3] proposed the activation clustering method to detect poisoned training samples crafted by adversaries to prevent the model from being attacked. Tran et al. [35] removed poisoned instances via singular value decomposition and anomaly detection algorithm.

Post-training defenses. This line focuses on the investigation after training. Gao et al. [8] intentionally perturbed the input and observed its randomness of the classification probability. If the classification probability is hardly changed with disturbance, the input instance may be a maliciously attacked sample. Wang et zal. [37] reconstructed candidate backdoor triggers by reverse engineering, utilized anomaly detection to determine the most potential backdoor and adopted a retrain patches strategy to remedy the backdoor's impact. Qiao et al. [24] developed a algorithm to clean the triggers in a backdoor attacked model by modeling a generator for potential triggers. DeepInspect [4] applies model inversion and conditional Generative Adversarial Network [21] to mitigate backdoor attacks. Recently, Neural Attention Distillation [16], motivated by knowledge distillation, was designed to enable a "teacher" network to guide the finetuning of the backdoored "student" network to erase backdoor triggers.

The above defenses strongly depend on a fact that the triggers are "unnatural" in poisoning samples, so that they may choose to produce concentrated feature regions highly correlated with the target label. This is where we place our approach - generating a natural feature map (which can effectively hide our triggers) - to defeat these defenses.

2.3. Threat Model

We consider the same threat model as in [6, 7, 23, 26], assuming adversaries can have full access to the backdoor training and the victim model. The infected model is then rendered to the public users who can employ it in applications after applying a certain backdoor detection algorithm.

3. Our Proposed Model: DEFEAT

3.1. Preliminary

We focus on backdoor attacks on image classification. Recall that a classifier can be described as a function $F_\theta(x) : \mathcal{I} \rightarrow \mathbb{R}^K$ that maps the input images to a classification result, where θ is the model parameter, $\mathcal{I} \in \mathbb{R}^{C\times H\times W}$ is a valid input image domain, K is the number of classes, H, W and C are the height, width and channels of an image. Let $\mathcal{D}_c = \{(x_i, y_i)|x_i \in \mathcal{I}, y_i \in \mathbb{R}^K\}_{i=1}^N$ indicate the clean training set containing N images. When poisoning F_θ, following the standard training procedure of backdoor attacks, we enforce it to train with the combination of the clean and poisoned data. For a clean image and the corresponding label (x, y), we create a poisoned sample by transforming it into $(T(x), y_t)$, where $T(.)$ is a backdoor injection function and y_t is the target label. In practice, we randomly select a small fraction of clean training data $\mathcal{D}_b$ to produce poisoned data. And the injection ratio is defined as $\eta = |\mathcal{D}_b|/|\mathcal{D}_c|$.

Our main objective is to learn a stealthy backdoor injection function T to craft poisoned data and naturally implant backdoor behavior into a victim model. For a network F with parameter θ, a backdoor injection function T with parameter ϕ, we minimize the following loss function to ensure performance on clean data:

$$\mathcal{L}_{clean}(\theta) = \sum_{(x,y)\in\mathcal{D}_c} \mathcal{L}^{ce}(F_\theta(x), y), \tag{1}$$

where $\mathcal{L}^{ce}$ denotes the cross-entropy loss. To achieve backdoor attack behavior, we then minimize the following function on poisoned data:

$$\begin{aligned}\mathcal{L}_{adv}(\theta) = \sum_{(x,y)\in\mathcal{D}_b} \big[&\mathcal{L}^{ce}(F_\theta(T_\phi(x), y_t) \\ &+ \mathcal{L}^{lf}(x, T_\phi(x), F_\theta)\big],\end{aligned} \tag{2}$$

where $\mathcal{L}^{lf}$ is latent feature constraint loss that we introduce. We can formalize the final objective as a constrained optimization problem:

$$\begin{aligned} &\min_{\theta}\ \beta_1\mathcal{L}_{clean}(\theta)+\beta_2\mathcal{L}_{adv}(\theta)\\ &s.t.\ \ \phi(\theta)=\arg\min_{\phi}\sum_{i}^{N}\big[\mathcal{L}^{ce}(F_\theta(T_\phi(x_i)),y_t)\\ &\qquad\quad +\ max(d(T_\phi(x_i),x_i)-\epsilon,0)\big], \end{aligned} \tag{3}$$

where $\mathcal{L}^{ce}$ is the cross-entropy loss, d is a distance measurement function, β_1 and β_2 control the mixing strengths of the loss. In the above bilevel problem, for a learned model F_θ, we optimize a backdoor injection function T_ϕ that can generate stealthy poisoned images. For an optimal T_ϕ, F_θ is trained to learn backdoor behavior under the latent feature constraint $\mathcal{L}^{lf}(x,T_\phi(x),F_\theta)$ so that F_θ correctly predicts the clean inputs, but incorrectly on the poisoned inputs, and the latent feature between them are very small distinguishability. The optimization of the problem in Equation 3 is a challenging task. We divide our solution into two steps: trigger generation and backdoor implantation, and execute them alternately to optimize F_θ and T_ϕ.

3.2. Trigger Generation

We train a backdoor injection function T_ϕ based on a given classifier F_θ. The function should provide two properties: 1) The resulting poisoned image is not detectable; and 2) The poisoned images can induce model misclassification. Inspired by the adversarial perturbation technology [7, 20, 22, 41], we design a T_ϕ that meets the requirements. Given a clean image x and the corresponding label y, we follow the perturbation-based methods to have:

$$T_\phi(x)=x+\phi, \tag{4}$$

where ϕ is the universal noise backdoor pattern. The goal here is to force predictions of $T_\phi(x)$ to target class y_t.

Given a clean model F_θ trained on clean data, we use the second optimization term of Equation 3 to learn a satisfactory backdoor injection function T_ϕ. We set $d(T_\phi(x),x)=||T_\phi(x)-x||_2$ which is l_2-norm distance on image-pixel space. And ϵ is a budget that controls the stealthiness of poisoned images generated by $T_\phi(x)$. An illustration of the trigger generation is described in Figure 2 (a).

3.3. Backdoor Implantation

Given $T_\phi(x)$, our next task is to implant it in the pretrained model F_θ effectively. We propose a novel poisoning training process motivated by latent feature constraints [29, 41] to produce more natural-embed backdoor triggers. The details are described in Figure 2 (b) and Figure 2 (c) .

Train Additional Logits. Formally, assume the model architecture F_θ consists of a sequence of N layers, and it is defined as:

$$F_\theta(x)=F^{(N)}(\cdots F^{(2)}(F^{(1)}(x))\cdots), \tag{5}$$

where $F^{(1)}$, $F^{(2)}$, $\cdots$, $F^{(N)}$ are the sequences of latent layers in the classifier F_θ. We train an intermediate feature logits $h^{(l)}$ based on F_θ, and then use it to constrain the poisoning training process to achieve the feature hiding. We denote $h^{(l)}:\mathbb{R}^{C^{(l)}\times H^{(l)}\times W^{(l)}}\to\mathbb{R}^K$ as a small auxiliary classifier with a global average pooling layer $\mathrm{pool}:\mathbb{R}^{C^{(l)}\times H^{(l)}\times W^{(l)}}\to\mathbb{R}^{C^{(l)}}$ followed by a fully-connected layer for classification:

$$h^{(l)}(z^{(l)})=\mathrm{pool}(z^{(l)})\psi^{(l)}+\xi^{(l)}, \tag{6}$$

where $z^{(l)}$ denotes the features extracted from the l^{th} layer in F_θ, and $\psi^{(l)}\in\mathbb{R}^{C^{(l)}\times K}$ and $\xi^{(l)}\in\mathbb{R}^K$ are the parameters to be trained in the function $h^{(l)}$. Note during the training procedure of $h^{(l)}$, the original model F_θ is used as a fixed feature extractor. In practice, we use clean training set $\mathcal{D}_c$ and cross-entropy loss function to optimize $h^{(l)}$.

Poisoning Training. We use the poisoned data $\mathcal{D}_b$ to fine-tune the classifier to achieve the goal of poisoning training. To realize the feature stealthy backdoor attack, we leverage the trained intermediate auxiliary classifier $h^{(l)}$ to constrain the poisoning training. We define the latent feature constraint loss $\mathcal{L}^{lf}$ as:

$$\mathcal{L}^{lf}_{\lambda}=\mathrm{mean}(\sum_{l\in[1:N]}\lambda^{(l)}|h^{(l)}(z^{(l)}_x)-h^{(l)}(z^{(l)}_{T_\phi(x)})|), \tag{7}$$

where x is a clean input image, for each layer $l\in[1:N],\lambda^{(l)}\in[0,1]$ assigns a weight to the layers with $\sum_{l\in N}\lambda^{(l)}=1$. The $z^{(l)}_x$ and $z^{(l)}_{T_\phi(x)}$ denote the feature extracted from the l^{th} layer for x and $T_\phi(x)$, respectively. $\mathcal{L}^{lf}(x,T_\phi(x),F_\theta)$ is utilized to measure the difference between the latent layer features of x and $T_\phi(x)$. Given a weight vector $\lambda=(\lambda^{(1)}\cdots\lambda^{(N)})$, we use the optimization of Equation (3) to obtain a backdoored model.

4. Evaluation

Datasets. We evaluate the effectiveness of DEFEAT on three standard image datasets: CIFAR-10 [13], GTSRB [31] and ImageNet [25]. CIFAR-10 contains 60k (32×32) color images equally divided amongst ten mutually exclusive classes, in which 50K for training and 10K for testing. GTSRB consists of over 51.2K (40×40) traffic sign images distributed amongst 43 mutually exclusive categories. Since the image sizes in the above datasets are relatively small, we further verify DEFEAT on large-size images in ImageNet. We only select ten classes from ImageNet randomly for the experiments, because training on a considerable amount of large-size images does consume

Table 1. Attack Effectiveness via ASR (%) and TAR (%). We report the results (Mean±SD) averaged over 10 independent runs.

	Dataset →	CIFAR-10		GTSRB		ImageNet	
Model ↓	Attack ↓	TAR	ASR	TAR	ASR	TAR	ASR
VGG16	Clean Model	92.91±0.30	-	98.43±0.07	-	85.26±0.30	-
	Badnets	90.03±0.43	99.17±0.02	95.43±0.06	**100.00±0.00**	83.54±0.20	97.69±0.10
	SIG	87.33±0.30	99.75±0.01	95.81±0.05	99.85±0.01	82.25±0.12	98.92±0.12
	ReFool	87.24±0.70	99.58±0.03	94.66±0.05	94.15±0.01	77.90±0.02	98.33±0.02
	WaNet	91.87±0.55	**99.95±0.02**	**96.59±0.15**	96.41±0.02	83.30±0.30	98.81±0.02
	DFST	90.17±0.25	99.80±0.02	96.78±0.25	98.96±0.01	81.38±0.50	**99.96±0.02**
	DEFEAT(Ours)	**91.93±0.50**	99.30±0.01	96.24±0.05	98.76±0.02	**84.97±0.18**	99.94±0.01
Resnet34	Clean Model	92.33±0.01	-	98.42±0.06	-	84.66±0.25	-
	Badnets	91.73±0.03	99.86±0.01	97.90±0.40	98.91±0.11	79.45±0.16	94.33±0.01
	SIG	91.49±0.03	99.38±0.20	98.19±0.06	**100.00±0.00**	80.37±0.30	97.04±0.01
	ReFool	91.09±0.22	99.05±0.10	97.94±0.01	98.47±0.03	74.39±0.24	94.35±0.22
	WaNet	92.03±0.30	99.96±0.01	98.19±0.31	99.83±0.01	79.77±0.30	95.31±0.03
	DFST	91.21±0.43	99.85±0.03	97.39±0.33	98.99±0.01	80.74±0.37	98.59±0.03
	DEFEAT(Ours)	**92.25±0.25**	**99.98±0.02**	**98.26±0.30**	99.01±0.05	**82.63±0.22**	**98.98±0.01**
WideResnet	Clean Model	93.35±1.10	-	98.42±0.06	-	85.79±0.10	-
	Badnets	92.54±0.03	99.88±0.02	97.39±0.02	**99.98±0.01**	84.02±0.13	91.61±0.06
	SIG	91.73±0.02	99.90±0.01	97.74±0.02	96.99±0.00	82.84±0.22	98.22±0.01
	ReFool	90.65±0.20	99.80±0.30	97.53±0.13	96.62±0.33	82.57±0.75	95.37±0.01
	WaNet	91.63±0.50	99.57±0.30	97.54±0.21	97.56±0.01	83.69±0.60	95.42±0.10
	DFST	92.14±0.40	99.85±0.02	97.04±0.33	98.19±0.02	83.92±0.33	**99.89±0.02**
	DEFEAT(Ours)	**93.24±0.70**	**99.98±0.11**	**98.55±0.01**	99.90±0.01	**84.08±0.30**	99.88±0.14

Algorithm 1 DEFEAT Backdoor Attack

```
Require: Clean Dataset $\mathcal{D}_c$, Parameters $\beta_1$, $\beta_2$ and $\gamma$, Injection Ratio $\eta$, Total Steps $R$, Stealthiness Budget $\epsilon$.
Ensure: Backdoored Classifier $F_\theta$ and Inject Function $T_\phi$.

Train clean model $F_\theta$ on $\mathcal{D}_c$.
Train additional logits $h^l$ for each layer $l \in [1 : N]$ based on clean model $F_\theta$ and clean data $\mathcal{D}_c$.
Initialize $\phi$.
Sample subset $\mathcal{D}_b$ from $\mathcal{D}_c$.
for $i = 1, \dots, R$ do
    Obtain the optimized $\phi(\theta)$ according to the second term of the Eq. (3).
    Compute the loss: $\beta_1 \mathcal{L}_{clean}(\theta) + \beta_2 \mathcal{L}_{adv}(\theta)$ as in Eq. (3).
    Update $\theta$ with stochastic gradient descent.
end for
return $\theta$, $\phi$
```

GPU resource and yield serious training overhead[1]. We collect 10.5K (224×224) training and 1.8K test samples from ImageNet. All images in datasets are normalized to [0,1].

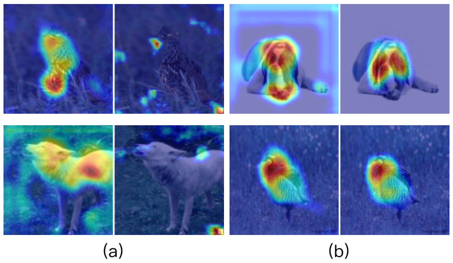


Figure 3. Latent layer feature comparison. In subfigure, left and right columns are the feature saliency map of the clean and poisoned samples. (a): BadNets; (b): DEFEAT.

Network structures. We perform the experiments on classic deep convolutional neural networks, namely VGG16 [30], ResNet34 [11] and WideResNet [42], which are also used for victim classifiers in [6, 19, 37]. And we use the backdoor attacks - BadNets [9], Reflection attack (ReFool) [19], Sinusoidal signal attack (SIG) [2], WaNet [23] and DFST [6] for comparison.

Implementation. To train the neural networks, we use initial learning rate 0.01 and schedule it to drop at epochs 50, 100, and 150 by a factor of 0.1. Models are trained by SGD optimizer with 200 epochs. For poisoning, following [41], we select the last five residual blocks for WideResNet, the last three residual blocks for ResNet34, and the last convolutional layer for VGG16 as the constraint of the

[1] Note using more images will not affect the performance of effectiveness and robustness but the training overhead.

latent feature. The layer will assign weights $\lambda^{(\cdot)}$ evenly, i.e. when the last six residual blocks are selected, we set $\lambda^{(N-5:N)} = 1/6$. We found that if $\beta_1 \gg \beta_2$, the backdoored classifier performs well on the clean data, which is close to the original clean classifier, but the results on poisoned data are not satisfactory. But if $\beta_2 \gg \beta_1$, the result is the other way round. To balance the performance, we set $\beta_1 = 1$ and $\beta_2 = 0.1$ in the experiments. After T_ϕ is optimized, we fine-tune the pre-trained clean model F_θ on the poisoning dataset with a small learning rate of 0.001 to implant the backdoor. A summary of DEFEAT is given in Algorithm 1. Note given a F_θ, we train additional logits in Line 2; in line 6 we optimize T_ϕ for the current θ. Line 7, 8 are the poisoning F_θ using optimized T_ϕ. We train the F_θ and T_ϕ using the alternating update for 20 epochs, i.e., we set $R = 20$.

4.1. Attack Experiments

Attack Effectiveness. We evaluate attack effectiveness with attack success rate (ASR), the ratio of backdoored examples misclassified as the target label, and test accuracy rate (TAR) on clean samples. To give a fair comparison, we set the infection rate to 0.1 for each attack method. And to reduce the time cost of the trigger generation, we randomly select 10K clean samples from the training set (instead of traversing the entire dataset). We conduct attack experiments in the single-target paradigm, in which the attacked target label y_t (class 0) is the same for all compared models per dataset. The results are presented in Table 1. In CIFAR-10, DEFEAT achieves the highest TAR values, while maintaining great ASR scores ($\geq 99.30 \pm 0.01$). As for the TAR on GTSRB, we obtain the best performance for Resnet34 and WideResnet, and get very close (nearly 0.35) to the best result (WaNet) for VGG16. Meanwhile, our ASR values are satisfactory, $\geq 98.76 \pm 0.02$. In ImageNet, we are also the best in TAR and present great ASR, e.g., around 99.94 (VGG16). Although the ASR results cannot always surpass others, they are all above 98.70, which is sufficient to make one successfully implant backdoors. We will further show that DEFEAT is more "invisible" and "natural" than others in the poisoned samples.

Attack Stealthiness. We consider the differences between clean and poisoned samples from the perspective of feature and input level. We thus use the following similarity metrics: learned perceptual image patch similarity (LPIPS) [44], structural similarity index (SSIM) [38] and peak-signal-to-noise-ratio (PSNR) [12]. LPIPS adopts the features of the pre-trained AlexNet to identify similarity, while SSIM and PSNR are calculated based on the statistical similarity at the pixel level. There exists a clear relationship between the values of SSIM, PSNR and LPIPS and the performance of the invisibility. Specifically, if the values of SSIM and PSNR are increased, the stealthiness will get enhanced, meaning that the poisoned sample looks "more" stealthy. But for LPIPS, that is the other way round.

For each dataset, we randomly select 500 sample images from the testing set to evaluate the stealthiness. The results are given in Table 2. DEFEAT achieves excellent stealthiness. Specifically, in CIFAR-10, it does outperform others, in which its PSNR is around 1.2 above that of BadNets, LPIPS is nearly 20% of improvement to that of BadNets, and SSIM is the highest, 0.9813; in GTSRB and ImageNet, our attack is right after WaNet and BadNets w.r.t. SSIM and LPIPS, and has the best PSNR (30.25, and 37.50).

The stealthiness of BadNets on GTSRB and ImageNet is better than ours. This is because BadNets uses a square pattern as a trigger. And if the image size is increased, the ratio of pattern to clean images will become relatively small. But using this pattern will make the original image look "unnatural". To prove that, we give examples in Figure 3 to compare the stealthiness between BadNets and DEFEAT. As compared to (a), DEFEAT's salient feature area (b) gathers on the main object and almost overlaps with that produced by the clean sample, which gives better attack stealthiness.

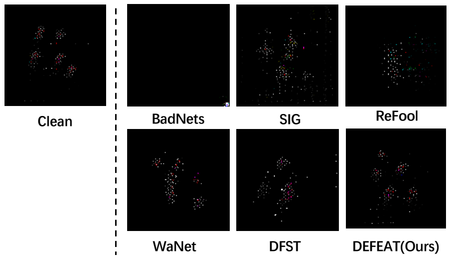


Figure 4. Trigger patterns optimized by Neural Cleanse.

4.2. Defense Experiments

We test our attack against the state-of-the-art defense algorithms, including STRIP [8], Neural Cleanse [37], Fine-Prunning [17] and Neural Attention Distillation [16].

Resistance to STRIP. The STRIP assumes that the predict result output by a backdoored model on a poisoned sample is firm and not easily disturbed. And it can detect poisoned samples by analyzing the entropy of the classification probability after superimposing some random samples. We test the performance of BadNets and DEFEAT with Resnet34 classifier. And we report the entropy probability density of clean and poisoned samples in Figure 5(a). The overlap of the distributions indicates the difficulty of identifying the backdoored input. In Fig.5, DEFEAT outperforms BadNets, especially in CIFAR-10 and GTSRB, in which the distributions of clean and backdoored data having a high coincidence are almost indistinguishable.

Resistance to Neural Cleanse. Neural cleanse identifies whether there is a backdoor in the model by reverse-engineering the triggers. In Figure 5(b), we report the re-

Table 2. Experimental results on attack stealthiness (PSNR ↑, SSIM ↑ and LPIPS ↓).

Dataset	Metric	BadNets	ReFool	SIG	WaNet	DFST	DEFEAT(Ours)
CIFAR-10	SSIM	0.9763	0.6542	0.9578	0.8854	0.7210	**0.9813**
	LPIPS	0.0012	0.0697	0.0015	0.0090	0.0881	**0.0009**
	PSNR	20.06	18.37	25.23	19.30	16.79	**21.31**
GTSRB	SSIM	0.9501	0.7418	0.7103	**0.9669**	0.7594	0.9171
	LPIPS	**0.0303**	0.3097	0.0850	0.0584	0.2510	0.0427
	PSNR	23.41	20.57	25.28	30.11	21.58	**30.25**
ImageNet	SSIM	**0.9955**	0.8564	0.8680	0.9359	0.7129	0.9765
	LPIPS	**0.0062**	0.4574	0.0573	0.0360	0.2105	0.0159
	PSNR	32.95	20.42	25.30	29.59	23.01	**37.50**

sults using the defense on three datasets with Resnet34 classifier. Successful detection may be related to the anomaly index. The default 2 (anomaly index value) is the detection threshold. BadNets and SIG suffer from the worst (approx. 3.84 and 2.27 on ImageNet). Since the trigger used in the BadNets is unnatrual in the original image, it is the easiest to be detected. Note this confirms our statements on the square patterns. Our attack is below the threshold.

We also display the reversed trigger patterns on ImageNet dataset optimized by Neural Cleanse for the attacking class on Imagenet in Figure 4. In BadNets and SIG, the reversed triggers are roughly similar to the original ones (Patch-based square and Sinusoidal signal as in Figure 1), revealing that such simple triggers may be vulnerable to reverse engineering. ReFool uses reflective backgrounds as trigger patterns, which are not easily reverse-engineered accurately. But it also increases the amplitude of changes to the clean sample, which impairs the stealthiness. Compared with others, pixels in our reversed trigger have no obvious regularity and are highly similar to the clean model. This is because: (1) the trigger is generated based on the optimization strategy to have better stealthiness; (2) the latent feature constraint also reduces the trigger's abnormality in the feature space, making it more difficult to be distinguished.

Resistance to Fine-pruning. The defense weakens the ability of the attacks by pruning dominant neurons in neural network to decrease the effect brought by the backdoored model. We do the test with VGG16 classifier, and report the results under various pruning rates of neurons in the final convolutional layer (Figure 5(c)). When the rate increases to 95%: the ASR is still above 90, while the TAR on clean data reduces significantly to about 60%, indicating our excellent robustness against the defense.

Resistance to Neural Attention Distillation. Neural attention distillation utilizes a teacher network to guide the fine-tuning of the backdoored student network on a small clean subset of data so that the intermediate-layer attention of the two networks can become aligned. We test the attacks with different iteration times on ImageNet with Resnet34. To avoid over-fitting caused by long training time, we set 20 iterations and report performance per 5 times. The results are shown in Table 3. DEFEAT outperforms others at Epochs #5, 10, and 15. At #20, the TAR is slightly smaller than that of DFST (approx. 0.32). As for TAR, DEFEAT shows a stable performance (after the first drop), around 80, as the increase of iteration; but for ASR, it suffers from a downward trend, from 98.98 to 79.99. The is because the defense uses part of the clean training data to fine-tune the victim model via distillation. This slightly affects the model performance on clean samples, but seriously harms the ASR values.

Our method can achieve a more stealthy trigger than other methods, without significantly reducing correlation between pixels. While being against the frequency-based defense [43], we also have distinct advantages over others.

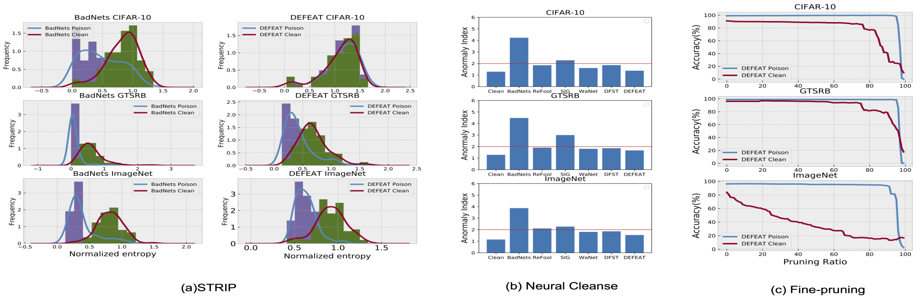
4.3. Hyperparameter Analysis

Influence of injection ratio η. We investigate our attack effectiveness under the pollution rate η. We set the ratio $= 10\%$ as default. In Table 4, we have the performance with various pollution rates on GTSRB with Resnet34. DEFEAT still works well, obtaining above 97.53 (TAR) and 99.21 (ASR) with 15% injection. We note that increasing the ratio may improve ASR, but harm TAR and raise the attack cost.

Influence of stealthiness budget ϵ. We visualize the impact of stealthiness budget ϵ on the input samples (Figure 6). The ϵ restrains the imperceptibility of the poisoned image in the input space. A smaller ϵ makes the trigger less noticeable. For example, when $\epsilon = 2$ in the second row, the trigger pattern is almost invisible in the clean sample. We further show the influence of ϵ on TAR and ASR with Resnet34 in Table 5. It is clear that increasing ϵ brings better results. The trigger is directly attached to the clean sample to create the poisoned one, and thus a larger ϵ makes the samples produce a larger distance in the input space. This benefits the model to distinguish them more easily and learn backdoor behavior with smaller performance loss on clean samples. But that also exposes the poisoned samples invisibility. After several attempts, we found that ϵ should be set differently

Table 3. Experimental results of Neural Attention Distillation.

	Original		Epochs #5		Epochs #10		Epochs #15		Epochs #20	
Attack	TAR	ASR	TAR	ASR	TAR	ASR	TAR	ASR	TAR	ASR
BadNets	79.45	94.33	62.78	14.79	75.31	14.42	75.52	13.99	74.72	7.48
ReFool	80.37	97.04	67.89	34.48	76.65	18.24	76.49	14.47	76.44	14.52
SIG	74.39	94.35	73.86	45.45	74.19	35.50	74.77	31.87	74.93	28.28
WaNet	79.77	95.31	66.93	36.42	57.28	35.13	69.57	32.99	69.71	36.01
DFST	80.74	98.59	77.97	50.78	76.90	42.01	**79.13**	36.06	**79.85**	31.71
DEFEAT(ours)	**82.63**	**98.98**	**80.90**	**86.71**	**82.46**	**81.53**	77.89	**82.21**	79.53	**79.99**



(a)STRIP

(b) Neural Cleanse

(c) Fine-pruning

Figure 5. Experimental results of the defense test.

Table 4. Injection ratio and attack effectiveness.

Injection Ratio	TAR	ASR
1%	98.79±0.07	93.66±0.24
5%	98.56±0.05	96.72±0.14
10%	98.26±0.30	99.01±0.05
15%	97.53±0.03	99.21±0.02

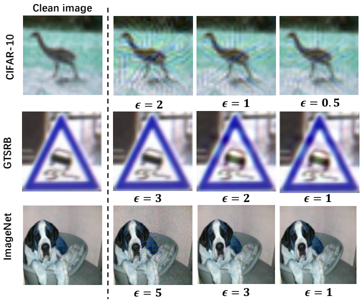


Figure 6. DEFEAT's poisoned samples.

according to the various sizes of images to obtain practical invisibility. We then set $\epsilon = 1, 2, 3$ for CIFAR-10, GTSRB and ImageNet as default, respectively.

5. Conclusion

We develop a novel deeply hidden feature backdoor attack on DNNs. We optimize adaptive backdoor triggers and actively constrain feature learning during the poisoning training process. Empirical experiments show that our design can provide imperceptible poisoned samples and great attacking performance. We hope this paper may promote further studies in developing robust and reliable DNNs.

Table 5. Stealthiness budget ϵ and attack effectiveness.

Dataset	ϵ	TAR	ASR
CIFAR-10	2	92.65±0.22	99.63±0.01
	1	92.25±0.25	99.98±0.02
	0.5	91.05±0.20	97.93±0.02
GTSRB	3	98.41±0.11	99.61±0.01
	2	98.26±0.30	99.01±0.05
	1	98.13±0.47	98.59±0.06
ImageNet	5	84.29±0.40	99.91±0.03
	3	82.63±0.22	98.98±0.01
	1	81.64±0.33	99.62±0.01

6. Acknowledgments

This work is supported by The National Key Research and Development Program of China No. 2021YFB3101400, the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDC02040400, and European Union's Horizon 2020 research and innovation programme under grant agreement No. 952697 (ASSURED) and No. 101021727 (IRIS).

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020. 2

[2] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 101–105. IEEE, 2019. 1, 5

[3] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018. 2, 3

[4] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, pages 4658–4664, 2019. 2, 3

[5] 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. 1, 2

[6] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. *arXiv preprint arXiv:2012.11212*, 2020. 2, 3, 5

[7] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11966–11976, 2021. 1, 3, 4

[8] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 113–125, 2019. 2, 3, 6

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

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 5

[12] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 6

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[14] Shaofeng Li, Minhui Xue, Benjamin Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 2020. 1

[15] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[16] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021. 2, 3, 6

[17] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018. 2, 6

[18] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017. 1, 2

[19] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer, 2020. 1, 2, 5

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2, 4

[21] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[22] Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *arXiv preprint arXiv:2010.08138*, 2020. 4

[23] Anh Nguyen and Anh Tran. Wanet – imperceptible warping-based backdoor attack. 2021. 1, 3, 5

[24] Ximing Qiao, Yukun Yang, and Hai Li. Defending neural backdoors via generative distribution modeling. *arXiv preprint arXiv:1910.04749*, 2019. 3

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4

[26] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11957–11965, 2020. 3

[27] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1

[28] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018. 2

[29] Reza Shokri et al. Bypassing backdoor detection algorithms in deep learning. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 175–183. IEEE, 2020. 2, 4

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[31] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012. 4

[32] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3520–3532, 2017. 2

[33] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019. 2

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 1

[35] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *arXiv preprint arXiv:1811.00636*, 2018. 2, 3

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1

[37] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 3, 5, 6

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[39] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019. 2

[40] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2041–2055, 2019. 1, 2

[41] Yunrui Yu, Xitong Gao, and Cheng-Zhong Xu. Lafeat: Piercing through adversarial defenses with latent features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5735–5745, 2021. 2, 4, 5

[42] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[43] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16473–16481, 2021. 7

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[45] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452, 2020. 2