An Invisible Black-box Backdoor Attack through Frequency Domain

Tong Wang[1], Yuan Yao[1], Feng Xu[1], Shengwei An[2], Hanghang Tong[3], and Ting Wang[4]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China
[2] Purdue University, USA
[3] University of Illinois Urbana-Champaign, USA
[4] Pennsylvania State University, USA
mg20330065@smail.nju.edu.cn, {y.yao,xf}@nju.edu.cn,
an93@purdue.edu, htong@illinois.edu, ting@psu.edu

Abstract. Backdoor attacks have been shown to be a serious threat against deep learning systems such as biometric authentication and autonomous driving. An effective backdoor attack could enforce the model misbehave under certain predefined conditions, i.e., *triggers*, but behave normally otherwise. The triggers of existing attacks are mainly injected in the pixel space, which tend to be visually identifiable at both training and inference stages and detectable by existing defenses. In this paper, we propose a simple but effective and invisible black-box backdoor attack FTrojan through trojaning the frequency domain. The key intuition is that triggering perturbations in the frequency domain correspond to small pixel-wise perturbations dispersed across the entire image, breaking the underlying assumptions of existing defenses and making the poisoning images visually indistinguishable from clean ones. Extensive experimental evaluations show that FTrojan is highly effective and the poisoning images retain high perceptual quality. Moreover, we show that FTrojan can robustly elude or significantly degenerate the performance of existing defenses.

Keywords: backdoor attack, black-box attack, frequency domain, invisibility

1 Introduction

CNNs are vulnerable to backdoor/trojan attacks [20, 34]. Specifically, a typical backdoor attack poisons a small subset of training data with a *trigger*, and enforces the backdoored model misbehave (e.g., misclassify the test input to a target label) when the trigger is present but behave normally otherwise at inference time. Such attacks can cause serious damages such as deceiving biometric authentication that is based on face recognition or misleading autonomous cars that rely on camera inputs.

An ideal backdoor attack should satisfy the three desiderata of *efficacy*, *specificity*, and *fidelity* from the adversary's perspective [39]. Here, efficacy means

Fig. 1. The poisoning images of existing backdoor attacks. (a) Poisoning images from BadNet [20], Blend [11], TrojanNN [34], Clean Label [49], Dynamic Backdoor [41], IAB [37], Latent Backdoor [58], and Composite Backdoor [31]. (b) Poisoning images from SIG [7] and Refool [35].

that the target CNN model can be successfully misled by the triggers, specificity means that the trained model should perform normally on the benign inputs, and fidelity means the poisoning images should retain the perceptual similarity to the original clean images. The latter two aspects are related to the *stealthiness* of a backdoor attack. That is, if either the trigger is clearly visible or the backdoored model performs relatively poor on the benign inputs, users may easily detect such an anomaly.

Motivation. While various existing backdoor attacks perform relatively well on the efficacy and specificity aspects, they tend to fall short in terms of satisfying the fidelity requirement, i.e., the triggers are visually identifiable (see Figure 1). The fundamental reason is that existing attacks directly inject or search for triggers in the spatial domain (i.e., pixel space) of an image. In this domain, it is a dilemma to find triggers that are simultaneously recognizable by CNNs and invisible to humans. Figure 1(a) shows the poisoning images from existing backdoor attacks whose triggers are concentrated in a small area, and thus the triggers are visually identifiable to a large extent. In view of this, several work proposes to disperse the trigger to a larger area to make it less visible. Figure 1(b) shows two black-box backdoor attacks on this thread. However, they are still generally detectable by humans (e.g., the wave pattern in the background or the abnormal reflective phenomenon). Recently, several work has successfully created invisible and effective white-box backdoor attacks [38, 17, 30]. However, they all require the control over the training process with knowledge of the learning model in use, which limits their usages in practice.

Insight and Contribution. In this paper, we propose a simple but effective and invisible black-box backdoor attack FTrojan through trojaning the frequency domain of images. It opens the door for various future backdoor attacks and defenses. Our key insights are two-fold. First, adding small perturbations in the mid- and high-frequency components can result in poisoning images with high

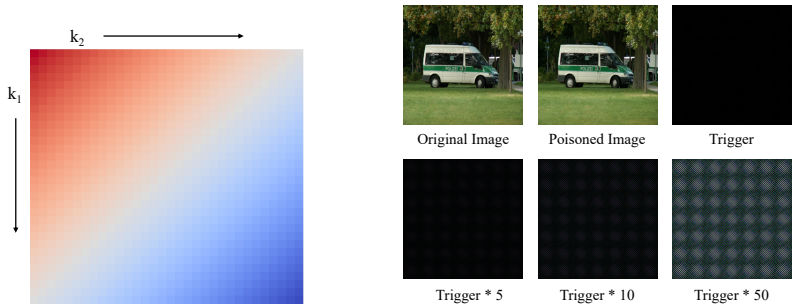
fidelity [45, 56]. Second, recent research has provided evidence that frequency-domain triggers, although dispersed throughout the entire image, are still recognizable and learnable by CNNs [59, 55, 54, 52]. Armed with the above insights, we first transform the images from RGB channels to YUV channels as UV channels correspond to chrominance components that are less sensitive to the human visual system (HVS). Next, we divide an image into a set of disjoint blocks and inject the trigger at both mid- or high-frequency components of the UV channels in each block. Through the above design, we can not only maintain the high fidelity of poisoning images, but also disperse the trigger throughout the entire image breaking the underlying assumptions of many existing defenses.

We evaluate our attack in several datasets and tasks including traffic sign recognition, objection classification, and face recognition. The results show that the proposed attack FTrojan achieves 98.78% attack success rate on average without significantly degrading the classification accuracy on benign inputs (0.56% accuracy decrease on average). Moreover, we compare the fidelity aspect with several existing backdoor attacks and show that the poisoning images by FTrojan are visually indistinguishable and retain higher perceptual quality. We also evaluate the proposed attack against state-of-the-art backdoor defensing systems including Neural Cleanse [51], ABS [33], STRIP [18], Februus [16], and NAD [29], as well as adaptive defenses based on anomaly detection and signal smoothing in the frequency domain. The results show that FTrojan can robustly bypass or significantly degenerate the performance of these defenses.

2 Attack Design

Overview. In the black-box setting of backdoor attacks, the adversary does not have the access or control to the CNN model, and he/she can only access part of the training data [20, 41]. Consequently, the key issue of such an attack is to design the triggers. To make the trigger invisible and effective, our FTrojan consists of the following steps. First, given an input RGB image, we first convert it to YUV channels. The reason is that YUV channels contain the bandwidth for chrominance components (i.e., UV channels) that are less sensitive to the HVS. Second, we transform the UV channels of the image from the spatial domain to the frequency domain via discrete cosine transform (DCT). Here, a small perturbation on the frequency domain may correspond to a large area in the spatial domain. In practice, we divide the images into a set of disjoint *blocks* and perform DCT on each block. Blocks that are too large would make the computation time-consuming, and too small could cause serious distortion to the image. We set block size to 32×32 in this work, and the frequency map of a block is shown in Figure 2(a), where we use *index* (k_1, k_2) to indicate each frequency band (the frequency goes from high to low from the upper left to the bottom right).[5] Third, FTrojan chooses a *frequency band* with a fixed

[5] Other design choices such as choosing to poison smaller blocks or fewer blocks are also studied, and the results, included in the supplementary material, show little difference in a wide range.



(a) Frequency map of DCT (b) The trigger of FTrojan

Fig. 2. (a) Frequency map of DCT. Each frequency band is indicated by the 2-D frequency index (k_1, k_2). (b) An illustration of the trigger of FTrojan. The trigger is scattered over the entire image and invisible to the HVS. To better visualize the trigger, we multiply each pixel value with a given factor in the second row.

magnitude in the frequency domain to serve as the trigger. We will later discuss different trigger generation strategies related to what frequency is the trigger placed on and what is the magnitude of the trigger. Finally, after the frequency trigger is generated, we apply inverse DCT to obtain the trigger in the spatial domain denoted by YUV channels, and transform the YUV channels back to the RGB channels since CNN models are mainly trained on the RGB color space.

Note that, once the trigger is defined in the frequency domain, it corresponds to fixed pixels (with fixed values) in the spatial domain. Therefore, we can use these pixels as the trigger to superimpose the original pixels to poison an image, without the need of repeatedly computing the above transforms.

Trigger Generation. Trigger generation involves the following two orthogonal dimensions, i.e., *trigger frequency* and *trigger magnitude*.

We first need to decide the specific frequency band that we aim to place the trigger on. On the one hand, placing the trigger at higher frequency would make the poisoning image even less sensitive to human perception, but such triggers could be erased by low-pass filters. On the other hand, triggers at lower frequency are robust against low-pass filters but could cause visual abnormalities if its magnitude is too large. In this work, we choose a more robust *mix mode*, i.e., placing one trigger at mid frequency and one at high frequency.

For trigger magnitude, larger magnitude may be easier for CNNs to learn and also robust against low-pass filters; however, it also comes at a risk of being detected by human perception or existing backdoor defenses. Smaller magnitude may bypass human perception and existing defenses, but being attenuated by the low-pass filters. We evaluate different choices in the experiment and choose a moderate magnitude depending on the specific datasets.

An example of our trigger is shown in Figure 2(b). We can visually observe that the poisoning images by our method retain very high perceptual similarity to their original images. Additionally, we can observe from the first row of

Table 1. Summary of the datasets and the classifiers used in our experiments.

Task	Dataset	# of Training/Test Images	# of Labels	Image Size	Model Architecture
Handwritten Digit Recognition	MNIST	60,000/10,000	10	$32 \times 32 \times 1$	2 Conv + 2 Dense
Traffic Sign Recognition	GTSRB	39,209/12630	43	$32 \times 32 \times 3$	6 Conv + 1 Dense
Object Classification	CIFAR10	50,000/10,000	10	$32 \times 32 \times 3$	6 Conv + 1 Dense
Object Classification	ImageNet	20,567/1,315	16	$224 \times 224 \times 3$	ResNet50
Face Recognition	PubFig	5,274/800	60	$224 \times 224 \times 3$	ResNet50

Figure 2(b) that, the injected trigger is nearly invisible to humans. To further show how the trigger looks like, we multiply each pixel value of the trigger by a factor and show the results in the second row of the figure. We can observe that the trigger is scattered over the entire image. More examples of poisoning images are shown in the supplementary material.

3 Evaluation

3.1 Experimental Setup

Tasks, Datasets, and Models. As summarized in Table 1, we conduct experiments on several benchmark tasks/datasets, including handwritten digit recognition on the MNIST data [27], traffic sign recognition on the GTSRB data [46], object classification on the CIFAR10 data [26] and the ImageNet data [15], and face recognition on the PubFig data [26]. We resize the images, and train different models for these tasks depending on the image size and complexity. For the GTSRB data, we follow standard processing such as histogram equalization in the HSV color space. For the ImageNet data, we randomly sampled 16 labels. For the PubFig data, we use the sampled subset of 60 persons from [35].

Evaluation Metrics. For efficacy and specificity, we measure the *attack success rate (ASR)* and the *accuracy on benign data (BA)*, respectively. For fidelity, it is still an open problem to measure it. In this work, we mainly consider if human eyes are sensitive to the poisoning images and use metrics *peak signal-to-noise ratio (PSNR)* [24], *structural similarity index (SSIM)* [53], and *inception score (IS)* [42, 8].

Implementations. For the proposed FTrojan attack, we implement it with two versions in both PyTorch and Tensorflow 2.0.[6] Our default settings are as follows. For trigger frequency, we place the trigger at frequency bands $(15, 15)$ and $(31, 31)$ where $(15, 15)$ belongs to the mid-frequency component and $(31, 31)$ belongs to the high-frequency component. Based on the size of images, we set the trigger magnitude to 30 for MNIST, CIFAR10, GTSRB, and 50 for ImageNet, PubFig. The injection rate is fixed to 5% for simplicity. We use the Adam optimizer with learning rate 0.0005 for MNIST and GTSRB, and the RMSprop optimizer with learning rate 0.001 for the rest datasets. The batch size is set to 64. In the following, we use FTrojan to denote the default setting unless

[6] The code is available at https://github.com/SoftWiser-group/FTrojan.

Table 2. Efficacy and specificity results of FTrojan variants. All the results are percentiles. For the default FTrojan (i.e., 'UV+mix' variant), it can achieve 98.78% ASR, while the BA decreases by 0.56% on average.

FTrojan Variant	MNIST		GTSRB		CIFAR10		ImageNet		PubFig	
	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
No attack	99.40	-	97.20	-	87.12	-	79.60	-	89.50	-
UV+mix	99.36	99.94	96.63	99.25	86.05	99.97	78.63	99.38	88.62	99.83
UV+mid	99.40	99.22	96.91	98.59	86.90	99.90	78.50	99.75	89.13	97.86
UV+high	99.39	99.81	96.63	99.12	86.90	99.90	78.75	99.14	88.25	99.93
YUV+mix	-	-	96.82	98.35	86.76	99.96	79.13	99.38	88.08	99.93
RGB+mix	-	-	97.16	92.05	86.33	95.99	78.70	95.46	89.37	99.25

otherwise stated. The target label is set to 8 for all the datasets. All the experiments were carried out on a server equipped with 256GB RAM, one 20-core Intel i9-10900KF CPU at 3.70GHz and one NVIDIA GeForce RTX 3090 GPU.

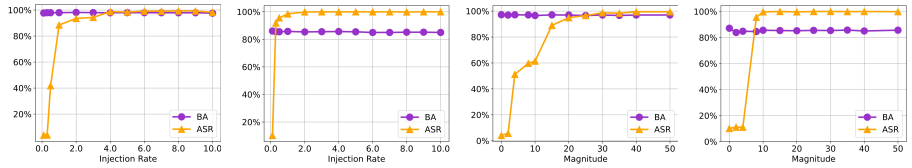
3.2 Attack Performance

(A) Overall Performance. We first evaluate different trigger generation strategies of the proposed FTrojan attack. The BA and ASR results are shown in Table 2, and the corresponding fidelity results are included in the supplementary material due to the space limit. For the variants in the table, 'UV', 'YUV', and 'RGB' indicate injected channels of the trigger,[7] and 'mid', 'high', and 'mix' mean the trigger frequencies. Here, 'mix' is our default setting as mentioned above, and frequency bands $(15, 15)$ and $(31, 31)$ are used for 'mid' and 'high', respectively.

We can first observe that all the FTrojan variants are effective, namely, decreasing little on BA and having a high ASR. For example, on average, the default FTrojan (i.e., 'UV+mix') can achieve 98.78% ASR, while the BA decreases by only 0.56%. Additionally, comparing different trigger frequencies, we can observe that all the three choices are closely effective and trojaning at high frequency tends to have higher fidelity results in general (based on the fidelity results in the supplementary material).

(B) Performance versus Injection Rate, Trigger Frequency, and Trigger Magnitude. We next evaluate the effectiveness of FTrojan when the injection rate of poisoning images in training data varies. We increase the injection rate from 0.01% to 10% and show the results in Figure 3(a) and 3(b). In the following, we mainly report results on GTSRB and CIFAR10 as training on these two datasets is more efficient. We can observe from the figure that BA does not change significantly when the injection rate is in a wide range. Additionally, when the injection rate is no less than 1%, FTrojan can achieve a high ASR for

[7] The MNIST images are gray-scale and have only one channel. We directly inject the trigger into this channel for Table 2.



(a) Performance vs. injection rate on GTSRB (b) Performance vs. injection rate on CIFAR10 (c) Performance vs. trigger magnitude on GTSRB (d) Performance vs. trigger magnitude on CIFAR10

Fig. 3. Performance vs. injection rate and trigger magnitude. FTrojan can achieve a high ASR when the injection rate is around 0.1% - 1%, and when the frequency magnitude is larger than a certain threshold. We fix the magnitude to 30 for GTSRB and CIFAR10, and fix injection rate to 5% in this work to ensure high ASR.

both datasets. This experiment also shows that different datasets have different sensitivity to the injection rate. For example, injecting 0.1% poisoning images could already achieve a high ASR on CIFAR10.

For trigger frequency, we study different frequency indices while keeping the other settings as default. It is observed that the backdoor attack is effective when the triggers are placed on mid- and high-frequency components. In this work, we choose a mix mode by default, i.e., triggering one mid-frequency index and one high-frequency index. The results are included in the supplementary material.

We next explore the effectiveness of FTrojan w.r.t. the trigger magnitude. We vary the trigger magnitude from 1 to 50, and show the results on GTSRB and CIFAR10 in Figure 3(c) and 3(d). We can observe that as long as the frequency magnitude is larger than a certain threshold, our backdoor attack will succeed with a high ASR. Based on our experiments, the poisoning images will not cause identifiable visual abnormalities when the trigger magnitude is no more than 100 in mid- and high-frequency components (e.g., see the images in the supplementary material). To ensure high ASR and robustness against filtering methods such as Gaussian filters, we set trigger magnitude to $30-50$ for different datasets based on the size of the images.

(C) Comparisons with Existing Attacks. Here, we compare FTrojan with existing backdoor attacks including BadNet [20], SIG [7], Refool [35], and IAB [37]. For BadNet, we implement it ourselves and add a 4×4 white block in the lower right corner as the trigger. For Refool, we use the implementation provided by the authors [6]. For SIG, we use the public implementation in the NAD repository [4]. For IAB, we also use its implementation from the authors [3]. Since Refool does not provide its implementations on MNIST and IAB does not provide its implementations on ImageNet and PubFig, we still report the results on GTSRB and CIFAR10 as shown in Table 3.

We can first observe from the table that our FTrojan attack achieves higher ASR scores than the competitors on both datasets. The BA scores of FTrojan are also very close to those of the clean model. Second, FTrojan outperforms the competitors for all the three fidelity metrics. Together with the visual results

Table 3. Comparison results with existing attacks. All the BA and ASR results are percentiles. Larger PSNR and SSIM, and smaller IS are better. FTrojan achieves higher ASR than the competitors on both datasets, and it outperforms the competitors for all the three fidelity metrics. Best results are in bold.

Attack Method	GTSRB					CIFAR10				
	BA	ASR	PSNR	SSIM	IS	BA	ASR	PSNR	SSIM	IS
No Attack	97.20	-	INF	1.000	0.000	87.12	-	INF	1.000	0.000
BadNet	96.51	84.98	24.9	0.974	0.090	86.01	94.80	23.8	0.941	0.149
SIG	96.49	92.56	25.3	0.973	1.353	85.70	95.76	25.2	0.871	1.905
Refool	96.41	56.52	19.1	0.923	1.035	85.87	73.20	17.3	0.769	0.910
IAB	92.12	64.84	23.8	0.956	0.226	85.10	79.70	13.2	0.829	2.240
FTrojan	**96.63**	**99.25**	**40.9**	**0.995**	**0.017**	**86.05**	**99.97**	**40.9**	**0.995**	**0.135**

Table 4. Defense results of Neural Cleanse. FTrojan can bypass Neural Cleanse (i.e., the abnormal index is smaller than 2).

Dataset	Abnormal Index	
	Clean	Backdoored
GTSRB	1.33	1.62
CIFAR10	1.25	1.85

in Figure 1, we can conclude that the proposed FTrojan attack is better than the competitors in the fidelity aspect.

In summary, the above results show that: 1) in the efficacy and specificity aspects, the proposed FTrojan *achieves a high attack success rate without significantly degrading the classification accuracy on benign inputs; and 2) in the fidelity aspects,* FTrojan *produces images with higher fidelity and perceptual quality under three evaluation metrics compared to the existing backdoor attacks.*

3.3 Evaluations against Defenses

Neural Cleanse. Neural Cleanse [51] detects triggers via searching for a small region with a fixed trigger pattern. The basic idea is that, no matter what the input is, the existence of the trigger pattern will lead the model to predict a fixed label. Then, it compares the norms of each identified pattern to determine the abnormal index of the classifier. Abnormal index larger than 2 is considered to be a backdoored model. We use the Neural Cleanse implementation provided by the authors [5], and the detection results are shown in Table 4. We can first observe that FTrojan can bypass Neural Cleanse on GTSRB and CIFAR10. The reason is that, based on the design nature, Neural Cleanse is effective when the trigger is relatively small and fixed. However, the injected trigger of FTrojan is dispersed over the entire image, and thus makes Neural Cleanse less effective in such cases.

Table 5. Defense results of ABS. Small REASR values mean that FTrojan successfully bypass the detection of ABS.

Dataset	REASR (Feature Space)		REASR (Pixel Space)	
	Clean	Backdoored	Clean	Backdoored
CIFAR10	0	0	0	0

Table 6. Defense results of STRIP. Most of the poisoned images by FTrojan can bypass the detection of STRIP.

Dataset	False Rejection Rate	False Acceptance Rate
GTSRB	4.10%	98.00%
CIFAR10	10.95%	77.40%

ABS. ABS [33] is a defense technique that scans through each neuron to see if its stimulation substantially and unconditionally increases the prediction probability of a particular label. It then reverses the trigger based on the identified neurons, and uses the trigger to attack benign inputs. If the ASR of the reversed trigger (i.e., REASR) is high, ABS reports the model as being backdoored. We use the implementation of ABS provided by the authors [1], which provides a binary executable file to run on CIFAR10. Thus, we only report the results on CIFAR10 in Table 5. We can observe that ABS cannot detect the backdoored model by our FTrojan attack. The probable reason is as follows. ABS is effective in terms of identifying one neuron or a few neurons that are responsible for a target label. However, the injected trigger by FTrojan scatters over the entire image in the spatial domain, which may affect a large number of neurons.

STRIP. STRIP [18] is an online inspection method working at inference stage. Its basic idea is that, if a given test image contains a trigger, superimposing the test image with clean images would result in a relatively lower classification entropy. Then, STRIP uses the entropy of the superimposed image to decide whether the test image contains a trigger. We apply STRIP on the test inputs and the results are shown in Table 6. We implement STRIP ourselves. The key parameter of STRIP is the entropy boundary, and we search it within our best efforts. The boundary is set to 0.133 for GTSRB and 0.30 for CIFAR10. In the table, we report the false rejection rate (the probability that a benign input is regarded as a poisoning input) and false acceptance rate (the probability that a poisoning image is regarded as a benign input) as suggested by STRIP. We can observe that STRIP yields a high false acceptance rate on both datasets, meaning that most of the poisoning images by FTrojan can bypass the detection of STRIP. For example, on CIFAR10 data, over three quarters of the poisoning images can bypass STRIP detection, and over 10% clean images are misclassified as poisoning images. The reason for the ineffectiveness of STRIP is that, when multiple images are superimposed in the spatial domain, the frequency domain of the superimposed image would change dramatically compared to the original

Table 7. Defense results of FEBRUUS. All the results are percentiles. FTROJAN significantly degenerates FEBRUUS's effectiveness. After applying FEBRUUS, although the ASR decreases by 15 - 25%, the BA decreases up to 75%.

Dataset	Before FEBRUUS		After FEBRUUS	
	BA	ASR	BA	ASR
GTSRB	97.56	88.62	22.15	72.82
CIFAR10	86.42	99.55	10.60	76.73

Table 8. Defense results of NAD. All the results are percentiles. NAD is ineffective in terms of defending against FTROJAN. The ASR is still high after applying NAD.

Dataset	Before NAD		After NAD	
	BA	ASR	BA	ASR
GTSRB	96.47	98.46	96.33	98.15
CIFAR10	81.12	99.80	78.16	99.41

test input. Consequently, the trigger would be ineffective after superimposition and thus cannot be detected by STRIP.

FEBRUUS. With the assumption that triggers are usually not in the center part of an image, FEBRUUS [16] first identifies and removes the suspicious area in the image that contributes most to the label prediction using GradCAM [43], and then uses GAN to restore the removed area. We use the implementation of FEBRUUS provided by the authors [2], and keep the default parameter settings. The results are shown in Table 7. It can be observed that after the images are sent to FEBRUUS for sanitization, although the ASR decreases by 15 - 25%, the BA drops significantly by up to 75%. The reason that FEBRUUS's performance significantly degenerates against our FTROJAN attack is as follows. The trigger of FTROJAN is placed on the entire image in the spatial domain, making it difficult to spot the suspicious area (see Figure 4 for examples). Additionally, when a relatively large area is removed (which is often the case of our attack), the restored image would introduce serious distortions, and thus make the training on such images less effective on the benign inputs.

NAD. NAD [29] utilizes a teacher network trained on a small set of clean data to guide the fine-tuning of the backdoored student network, so as to erase the effect of triggers. The teacher network shares the same architecture with the student network. During knowledge transfer from the teacher network to the student network, NAD requires the alignment of the intermediate-layer's attention. We use the implementation provided by the authors [4] and keep the default parameters. The results are shown in Table 8.[8] It can be observed from

[8] Here, for better reproducibility of the results, we use the same model in the NAD repository instead of our CNN models. Therefore, the BA scores in the table is slightly lower than the previous results.

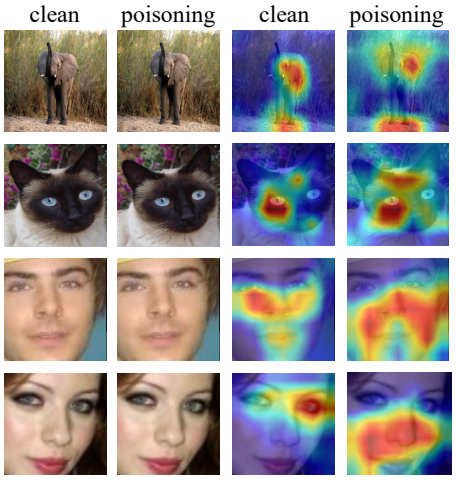


Fig. 4. The responsible region for prediction by GradCAM [43]. Our attack does not introduce unusual regions as existing spatial triggers.

the table that after applying NAD, the ASR is still very high meaning that NAD is ineffective in terms of erasing the impact of our attack. The possible reason is that the parameters of the backdoored model do not deviate significantly from those in the clean model, as our triggers are very small (in terms of pixel values) and dispersed across the entire image. Therefore, knowledge transferring from clean model may not help in such cases.

Visual Capture by GradCAM. We next illustrate the reason of the ineffectiveness of existing defenses. Specifically, we use GradCAM [43] to capture the influential area in an image that is responsible for the prediction, and some examples are shown in Figure 4. Warmer colors indicate more influence. The first two and last two images are selected from ImageNet and PubFig, respectively. We can observe that the warm areas of the poisoning images do not contain unusual regions as existing spatial triggers (see the supplementary material for some examples). Additionally, the warm areas of poisoning images are similar to that of clean images, but generally covering a relatively larger area. This breaks the underlying assumptions of existing defenses that rely on identifying a small, unusual region that significantly determines the prediction results.

Adaptive Defenses. Finally, we evaluate the effectiveness of FTrojan against adaptive defenses that directly operate on the frequency domain. In particular, we consider two adaptive defenses, i.e., *anomaly detection* and *signal smoothing* in the frequency domain. For the former, we evaluate whether the attack can be identified by applying existing anomaly detection methods on the images, and the results show that such defenses are ineffective (see the supplementary material for detailed results). For the latter, we consider three filters, i.e., Gaussian filter, Wiener filter, and BM3D [14], which are widely used in image denoising and

Table 9. Defense results of Gaussian filter, Wiener filter, and BM3D. Although these filters lower the ASR of FTrojan, they also significantly degenerate the BA performance.

Filters and Parameters	GTSRB			CIFAR10		
	BA	ASR	BA Decrease	BA	ASR	BA Decrease
Original	97.20	-	-	87.12	-	-
Gaussian filter ($w = (3, 3)$)	90.81	8.45	-6.39	69.72	26.38	-17.40
Gaussian filter ($w = (5, 5)$)	89.20	6.40	-8.00	53.21	19.48	-33.91
Wiener filter ($w = (3, 3)$)	92.87	3.54	-4.33	70.24	9.16	-16.88
Wiener filter ($w = (5, 5)$)	89.79	3.08	-7.41	61.04	5.84	-26.08
BM3D ($\sigma = 0.5$)	92.31	4.42	-4.89	82.34	15.84	-4.78
BM3D ($\sigma = 1.0$)	91.53	10.82	-5.67	81.40	19.33	-5.72

restoration. We apply these filters to the training data before feeding them to the model. We evaluate these filters in a wide range of parameters and observe similar results. The results are shown in Table 9. It is observed that although these filters are effective in terms of lowering the ASR, they also significantly degenerate the BA performance (e.g., from 4.33% to 33.91% absolute decrease). For Gaussian filter and Wiener filter, the minimum window size is 3×3, and larger w leads to stronger smoothing. We observe that even with the minimum window size, the BA already significantly decreases (e.g., 17.40% and 16.88% absolute decreases for Gaussian filter and Wiener filter on CIFAR10). For BM3D, we vary the noise standard deviation parameter σ, with larger σ indicating stronger smoothing. It is observed that even with $\sigma = 0.5$, the BA still significantly decreases. Overall, these results imply a fundamental accuracy-robustness trade-off for the above defenders.

In summary, the above results show that our FTrojan *attack can bypass or significantly degenerate the performance of the state-of-the-art defenses, as well as anomaly detection and signal smoothing techniques in the frequency domain. These results indicate that new defending techniques are still in demand to protect against our* FTrojan *attacks.*

4 Related Work

Backdoor Attacks. Backdoor attacks are introduced by [20, 11], where predefined triggers are injected into training data so that the trained model would mis-predict the backdoored instances/images as a target label. Later, researchers pay more attention to robust backdoor attacks that could reduce the effectiveness of existing backdoor defenses [58, 40, 44, 31, 22, 41, 37]. For example, Yao et al. [58] generate triggers whose information is stored in the early internal layers. Salem et al. [41] and Eguyen et al. [37] propose dynamic backdoor attacks to generate dynamic triggers conditioned on the input images. For the above backdoor attacks, although their evaluations show that they can bypass some

defenses such as Neural Cleanse [51] and STRIP [18], the generated triggers are still visually identifiable to a large extent.

Later, several researchers propose to make the triggers less visible by dispersing the trigger to a much larger area of the image. For example, SIG [7] transfers the images with superimpose signals (e.g., a ramp signal or a sinusoidal signal), and triggers are contained in the varying background. Refool [35] defines triggers resembling to the natural reflection phenomenon, and shows that it is resistant to several defenses including Fine-pruning [32] and Neural Cleanse [51]. Although these attacks follow the black-box setting, the triggers are still visually detectable. Recent work [28, 38, 17, 30] has successfully created invisible and effective backdoor attacks. For example, Doan et al. [17] jointly learn the stealthy trigger and the optimal classifier under a constrained optimization framework. Li et al. [30] borrow the idea from image steganography [61] by hiding an attacker-specified string into images. However, they all require the control over the training process with knowledge of the learning model in use. Different from the above work, we propose the first black-box backdoor attack that is both effective and invisible through trojaning in the frequency domain of images.

There also exist backdoor attacks that directly inject triggers into the trained networks without accessing the training data [34, 47, 13, 39]. In these attacks, the triggers can be inverted from the trained networks and then injected into the test images. For example, TrojanNN [34] identifies triggers that could maximize the activations of certain specific neurons, and retrains the model with generated images (both with and without triggers). Pang et al. [39] further study the connections between adversarial attacks and model poisoning attacks, leading to optimized version of TrojanNN against existing defenses. However, the generated triggers of these attacks are still visually identifiable.

Defenses against Backdoor Attacks. Existing backdoor defenses can be roughly divided into three categories, i.e., *model inspection*, *trigger detection or erasion*, and *model tuning*. Defenses in the first category focus on inspecting whether a given DNN has been backdoored [51, 33, 9, 21, 10, 25, 23]. For example, Neural Cleanse [51] propose to identify the shortcuts (small perturbations) across different labels to decide whether a model is backdoored or not. If the model is backdoored, it further reverts the trigger from the identified perturbations, and propose mitigate the attacks based on the reverted trigger. DeepInspect [10] is similar to Neural Cleanse except that it does not require the access to training data and the model parameters. Instead, DeepInspect infers the training data via model inversion [57]. ABS [33] first identifies the neurons that substantially maximize the activation of a particular label, and then examines whether these neurons lead to a trigger.

Assuming that the given DNN has been backdoored, defenses in the second category mainly aim to detect whether an instance has been corrupted or how to erase the triggers in the input images [48, 12, 36, 50, 18, 16]. For example, Tran et al. [48] find that corrupted instances usually have a signature in the spectrum of the covariance of their features, and train a classify to detect such instances. STRIP [18] propose to add perturbations on the test image to check

if it has a trigger, based on the intuition that trojaned images usually make the consistent prediction (i.e., the target label) even when various perturbations are added. Februus [16] first deletes the influential region in an image identified by GradCAM [43], and then restores the image via GAN.

In the third category, the defenses still assume that the model has been backdoored and propose to directly mitigate the effect of the backdoor attacks by tuning the models [32, 60, 29]. For example, Fine-pruning [32] prunes and fine-tunes the neurons that are potentially responsible for the backdoor attacks; however, it was observed that Fine-pruning could bring down the overall accuracy of the given model. Zhao et al. [60] introduce mode connectivity [19] into backdoor mitigation, and found that the middle range of a path (in the loss landscapes) connecting two backdoored models provides robustness. NAD [29] uses a teacher network trained on clean data to erase the triggers' effect in the backdoored student network via knowledge distillation.

5 Conclusion and Discussion

In this paper, we propose a black-box, frequency-domain backdoor attack FTrojan. We explore the design space and show that trojaning at UV channels, and injecting mid- and high-frequency triggers in each block with medium magnitude can achieve high attack success rate without degrading the prediction accuracy on benign inputs. The poisoning images of FTrojan are also of higher perceptual quality compared with several existing backdoor attacks. In terms of defending against our backdoor attacks, we show that the proposed FTrojan can bypass or significantly degenerate the performance of existing defenses and adaptive defenses.

Currently, we evaluate our attack against CNNs only. How can it be extended to other models and how does it perform on other learning tasks such as natural language processing tasks are unclear. We plan to explore such directions in future work. To defend against the proposed attacks, we also plan to design more robust defenses that go beyond the current assumption of backdoor attacks in the spatial domain. For example, one possible direction is to explore the subtle behavior difference between poisoning samples and benign samples.

Acknowledgement

We would like to thank Yingqi Liu for help reproducing the evaluation of ABS defense and providing comments. This work is supported by the National Natural Science Foundation of China (No. 62025202), and the Collaborative Innovation Center of Novel Software Technology and Industrialization. Hanghang Tong is partially supported by NSF (1947135, 2134079, and 1939725). Ting Wang is partially supported by the National Science Foundation under Grant No. 1953893, 1951729, and 2119331. Yuan Yao is the corresponding author.

References

1. Abs implementation. https://github.com/naiyeleo/ABS
2. Februus implementation. https://github.com/AdelaideAuto-IDLab/Februus
3. Inputaware backdoor implementation. https://github.com/VinAIResearch/input-aware-backdoor-attack-release
4. Nad implementation. https://github.com/bboylyg/NAD
5. Neural cleanse implementation. https://github.com/bolunwang/backdoor
6. Refool implementation,. https://github.com/DreamtaleCore/Refool
7. Barni, M., Kallas, K., Tondi, B.: A new backdoor attack in cnns by training set corruption without label poisoning. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 101–105. IEEE (2019)
8. Barratt, S., Sharma, R.: A note on the inception score. arXiv preprint arXiv:1801.01973 (2018)
9. 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. In: Workshop on Artificial Intelligence Safety, co-located with the Thirty-Third AAAI Conference on Artificial Intelligence (2019)
10. Chen, H., Fu, C., Zhao, J., Koushanfar, F.: Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI) (2019)
11. 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)
12. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: International Conference on Machine Learning (ICML). pp. 1310–1320. PMLR (2019)
13. Costales, R., Mao, C., Norwitz, R., Kim, B., Yang, J.: Live trojan attacks on deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 796–797 (2020)
14. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing **16**(8), 2080–2095 (2007)
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition (CVPR). pp. 248–255. Ieee (2009)
16. Doan, B.G., Abbasnejad, E., Ranasinghe, D.C.: Februus: Input purification defense against trojan attacks on deep neural network systems. In: Proceedings of the Annual Computer Security Applications Conference (ACSAC). pp. 897–912 (2020)
17. Doan, K., Lao, Y., Zhao, W., Li, P.: Lira: Learnable, imperceptible and robust backdoor attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11966–11976 (2021)
18. 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 (ACSAC). pp. 113–125 (2019)
19. Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D., Wilson, A.G.: Loss surfaces, mode connectivity, and fast ensembling of dnns. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS). pp. 8803–8812 (2018)

20. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
21. Guo, W., Wang, L., Xing, X., Du, M., Song, D.: Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. arXiv preprint arXiv:1908.01763 (2019)
22. He, Y., Shen, Z., Xia, C., Hua, J., Tong, W., Zhong, S.: Raba: A robust avatar backdoor attack on deep neural network. arXiv preprint arXiv:2104.01026 (2021)
23. Huang, S., Peng, W., Jia, Z., Tu, Z.: One-pixel signature: Characterizing cnn models for backdoor detection. In: European Conference on Computer Vision (ECCV). pp. 326–341. Springer (2020)
24. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electronics letters **44**(13), 800–801 (2008)
25. Kolouri, S., Saha, A., Pirsiavash, H., Hoffmann, H.: Universal litmus patterns: Revealing backdoor attacks in cnns. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 301–310 (2020)
26. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
27. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
28. Li, S., Xue, M., Zhao, B., Zhu, H., Zhang, X.: Invisible backdoor attacks on deep neural networks via steganography and regularization. IEEE Transactions on Dependable and Secure Computing (2020)
29. Li, Y., Koren, N., Lyu, L., Lyu, X., Li, B., Ma, X.: Neural attention distillation: Erasing backdoor triggers from deep neural networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)
30. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16463–16472 (2021)
31. Lin, J., Xu, L., Liu, Y., Zhang, X.: Composite backdoor attack for deep neural network by mixing existing benign features. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 113–131 (2020)
32. 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 (RAID). pp. 273–294. Springer (2018)
33. Liu, Y., Lee, W.C., Tao, G., Ma, S., Aafer, Y., Zhang, X.: Abs: Scanning neural networks for back-doors by artificial brain stimulation. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 1265–1282 (2019)
34. Liu, Y., Ma, S., Aafer, Y., Lee, W., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. In: Annual Network and Distributed System Security Symposium (NDSS) (2018)
35. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: European Conference on Computer Vision (ECCV). pp. 182–199. Springer (2020)
36. Ma, S., Liu, Y.: Nic: Detecting adversarial samples with neural network invariant checking. In: Proceedings of the 26th Network and Distributed System Security Symposium (NDSS 2019) (2019)
37. Nguyen, T.A., Tran, A.: Input-aware dynamic backdoor attack. In: Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS) (2020)

38. Nguyen, T.A., Tran, A.T.: Wanet-imperceptible warping-based backdoor attack. In: International Conference on Learning Representations (ICLR) (2021)
39. Pang, R., Shen, H., Zhang, X., Ji, S., Vorobeychik, Y., Luo, X., Liu, A., Wang, T.: A tale of evil twins: Adversarial inputs versus poisoned models. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 85–99 (2020)
40. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). pp. 11957–11965 (2020)
41. Salem, A., Wen, R., Backes, M., Ma, S., Zhang, Y.: Dynamic backdoor attacks against machine learning models. arXiv preprint arXiv:2003.03675 (2020)
42. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS). pp. 2234–2242 (2016)
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: Proceedings of the IEEE international conference on computer vision (ICCV). pp. 618–626 (2017)
44. Shokri, R., et al.: Bypassing backdoor detection algorithms in deep learning. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 175–183. IEEE (2020)
45. Sonka, M., Hlavac, V., Boyle, R.: Image processing, analysis, and machine vision. Cengage Learning (2014)
46. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The german traffic sign recognition benchmark: a multi-class classification competition. In: The 2011 international joint conference on neural networks (IJCNN). pp. 1453–1460. IEEE (2011)
47. Tang, R., Du, M., Liu, N., Yang, F., Hu, X.: An embarrassingly simple approach for trojan attack in deep neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). pp. 218–228 (2020)
48. Tran, B., Li, J., Mądry, A.: Spectral signatures in backdoor attacks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS). pp. 8011–8021 (2018)
49. Turner, A., Tsipras, D., Madry, A.: Clean-label backdoor attacks (2018)
50. Udeshi, S., Peng, S., Woo, G., Loh, L., Rawshan, L., Chattopadhyay, S.: Model agnostic defence against backdoor attacks in machine learning. arXiv preprint arXiv:1908.02203 (2019)
51. 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. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 707–723. IEEE (2019)
52. Wang, H., Wu, X., Huang, Z., Xing, E.P.: High-frequency component helps explain the generalization of convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8684–8694 (2020)
53. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
54. Xu, Z.Q.J., Zhang, Y., Luo, T., Xiao, Y., Ma, Z.: Frequency principle: Fourier analysis sheds light on deep neural networks. arXiv preprint arXiv:1901.06523 (2019)

55. Xu, Z.Q.J., Zhang, Y., Xiao, Y.: Training behavior of deep neural network in frequency domain. In: International Conference on Neural Information Processing (ICONIP). pp. 264–274. Springer (2019)
56. Yamaguchi, S., Saito, S., Nagano, K., Zhao, Y., Chen, W., Olszewski, K., Morishima, S., Li, H.: High-fidelity facial reflectance and geometry inference from an unconstrained image. ACM Transactions on Graphics (TOG) **37**(4), 1–14 (2018)
57. Yang, Z., Zhang, J., Chang, E.C., Liang, Z.: Neural network inversion in adversarial setting via background knowledge alignment. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 225–240 (2019)
58. 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 (CCS). pp. 2041–2055 (2019)
59. Yin, D., Lopes, R.G., Shlens, J., Cubuk, E.D., Gilmer, J.: A fourier perspective on model robustness in computer vision. In: Annual Conference on Neural Information Processing Systems (NeurIPS). pp. 13255–13265 (2019)
60. Zhao, P., Chen, P.Y., Das, P., Ramamurthy, K.N., Lin, X.: Bridging mode connectivity in loss landscapes and adversarial robustness. In: Proceedings of the International Conference on Learning Representations (ICLR) (2020)
61. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 657–672 (2018)