Supplementary Material for 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, inbox.ting@gmail.com

Abstract. Here, we provide additional experimental setup and results for "An Invisible Black-box Backdoor Attack through Frequency Domain" in ECCV 2022.

1 Evaluation Metrics

Here, we provide the definitions of the three fidelity evaluation metrics for completeness.

PSNR is the ratio of the maximum possible power of a signal to the destructive noise power that affects its accuracy. It is defined as

$$PSNR = 10 \log_{10}(\frac{MAX_I^2}{MSE}) \tag{1}$$

where MSE is defined as

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (x(i,j) - y(i,j))^2. \tag{2}$$

In the equations, x is the original image, y is the poisoning image, m and n are the width and height of the image. MAX_I is the maximum possible pixel value of the image (255 for 8-bit images).

SSIM is an index to measure the similarity of two images. It is calculated based on the luminance and contrast of local patterns. Given two images, x and y, let L(x, y), C(x, y), and S(x, y) be luminance, contrast, and structural

measures defined as follows,

$$\begin{aligned} L(x,y) &= \frac{\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C1} \\ C(x,y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ S(x,y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{aligned} \tag{3}$$

where μ_x, σ_x, and σ_{xy} are weighted mean, variance, and covariance, respectively, and C_i 's are constants to prevent singularity. where $C_1 = (K_1 L)^2$ and L is the dynamic range of the pixel values (255 for 8-bit images), $K_1 = 0.01$; $C_2 = (K_2 L)^2$, $K_2 = 0.03$; $C_3 = C_2/2$. It should be noted that the above x and y are all calculated in the RGB space. Then, the SSIM index is defined as

$$SSIM(x,y) = L(x,y)C(x,y)S(x,y). \tag{4}$$

IS (inception score) is first proposed to measure the quality of images generated from GANs. It mainly considers two aspects, one is the clarity of generated images, and the other is the diversity of images. Here we mainly focus on the difference between images containing triggers and the original images. It uses features of the InceptionV3 network trained on ImageNet classification dataset to mimic human visual perception. Inputting two images into InceptionV3 will output two 1000-dimensional vectors representing the discrete probability distribution of their categories. For two visually similar images, the probability distributions of their categories are also similar. Given two images x and y, the computation of IS can be expressed as follows,

$$IS(x,y) = KL(\phi(x), \phi(y))$$

$$KL(\phi(x), \phi(y)) = \sum_{i=1}^{N} \phi(x)_i \log \frac{\phi(x)_i}{\phi(y)_i} \tag{5}$$

where $\phi(\cdot)$ represents the discrete probability distribution of the predicted labels of InceptionV3, and $KL(\cdot,\cdot)$ represents Kullback-Leibler divergence.

2 Color Channel Transform and Discrete Cosine Transform

Next, we provide the details of color channel transform and discrete cosine transform in the proposed attack for completeness. Specifically, pixels in RGB channels can be converted to and back from YUV channels with the linear transformations in Eq (6) and Eq. (7), respectively. In the equations, (R, G, B) and (Y, U, V) stand for the channel values of a pixel in the RGB space and the YUV space, respectively.

$$
\begin{aligned}
Y &= 0.299 * R + 0.587 * G + 0.114 * B, \\
U &= 0.596 * R - 0.272 * G - 0.321 * B, \\
V &= 0.212 * R - 0.523 * G - 0.311 * B,
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
R &= Y + 0.956 * U + 0.620 * V, \\
G &= Y - 0.272 * U - 0.647 * V, \\
B &= Y - 1.108 * U - 1.705 * V.
\end{aligned} \tag{7}
$$

DCT expresses an image as a set of cosine functions oscillating at different frequencies. Compared with discrete Fourier transform (DFT), DCT is better in terms of energy concentration and is widely used in processing images. Specifically, we use the 2-D Type-II DCT transform [1] as follows,

$$
\begin{aligned}
X(k_1, k_2) &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) c_1(n_1, k_1) c_2(n_2, k_2), \\
c_i(n_i, k_i) &= \widetilde{c}_i(k_i) \cos(\frac{\pi(2n_i+1)k_i}{2N_i}), \\
\widetilde{c}_i(k_i) &= \begin{cases} \frac{1}{\sqrt{N_i}} & k_i = 0 \\ \frac{2}{\sqrt{N_i}} & k_i \neq 0 \end{cases} \quad i = 1, 2,
\end{aligned} \tag{8}
$$

which transforms the size $N_1 \times N_2$ input image in spatial domain to its frequency domain with the same size. Here, (k_1, k_2) stands for the index in the frequency map, $k_1/k_2 \in \{0, 1, \ldots, N_1/N_2\}$, $X(k_1, k_2)$ is the frequency magnitude at (k_1, k_2), and $x(n_1, n_2)$ is the pixel value in position (n_1, n_2) of the image in spatial domain. To transform the image from frequency domain back to spatial domain, we can use the inverse DCT transform [6] whose equations are similar to Eq. (8) and thus omitted for brevity.

3 More Experimental Results

Next, we provide more experimental results.

3.1 Poisoning images by FTrojan

Figure 1 shows more poisoning images by FTrojan. All the images are stamped with triggers in UV channels of each block. The original clean images are in the first column, and the rest columns contain the poisoning images. The fifth column stands for our default setting, with triggers in mix mode (indexed by $(15, 15)$ and $(31, 31)$) of magnitude 50. We can observe that when the triggers reside in either mid-frequency or high-frequency bands with moderate magnitude (e.g., no more than 100), the poisoning images are perceptually similar to the corresponding clean images and difficult to visually detect.

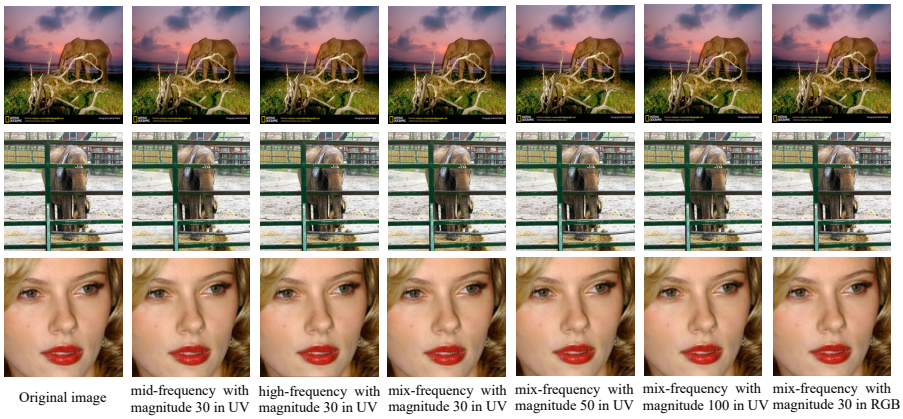


Fig. 1. Poisoning images by our FTrojan attack. Mix-frequency mixes triggers in both mid- and high-frequency components. We can observe that when the triggers reside in the high-frequency and mid-frequency components with moderate magnitude, the poisoning images are difficult to visually detect.

Table 1. Fidelity results of FTrojan variants. Larger PSNR and SSIM, and smaller IS are better. Triggering at UV channels achieves better results than at YUV and RGB channels. Triggering at high-frequency only is slightly better than at mid-frequency and mix-frequency.

FTrojan Variant	GTSRB			CIFAR10			ImageNet			PubFig		
	PSNR	SSIM	IS	PSNR	SSIM	IS	PSNR	SSIM	IS	PSNR	SSIM	IS
No Attack	INF	1.000	0.000	INF	1.000	0.000	INF	1.000	0.000	INF	1.000	0.000
UV+mix	40.9	0.995	0.017	40.9	0.995	0.135	37.7	0.727	0.020	37.7	0.802	0.213
UV+mid	43.3	0.995	0.011	43.5	0.997	0.098	40.5	0.775	0.014	40.5	0.861	0.176
UV+high	43.3	0.995	0.007	43.5	0.997	0.049	40.5	0.796	0.009	40.5	0.870	0.019
YUV+mix	25.7	0.943	0.458	36.5	0.985	0.279	25.7	0.670	0.258	21.3	0.806	1.571
RGB+mix	45.8	0.995	0.012	45.7	0.997	0.046	40.4	0.784	0.045	41.3	0.861	0.282

3.2 Fidelity Results of FTrojan Variants

The fidelity results of FTrojan variants are shown in Table 1. We exclude the results on MNIST in the table as it contains only one channel. First, we can observe that although as effective as the default FTrojan, injecting triggers into YUV channels instead of UV channels results in worse fidelity results as indicated by Table 1 (the fifth row). Second, injecting triggers at RGB channels is less effective than at UV channels, and it also results in lower fidelity (sixth row in Table 1). This is probably due to that the frequencies are more messy in RGB channels.

Table 2. Performance vs. trggier frequency. All the results are percentiles. Triggering at mid- or high-frequency components generally results in better BA and ASR results.

Frequency Index	GTSRB		CIFAR10	
	BA	ASR	BA	ASR
(2, 6)	94.60	81.49	84.12	84.16
(4, 4)	94.79	44.71	84.59	70.85
(8, 8)	95.77	77.11	82.79	13.72
(8, 20)	96.11	94.63	85.49	96.91
(12, 12)	97.11	96.65	86.44	90.36
(12, 16)	96.69	91.75	86.95	99.36
(20, 20)	96.60	95.21	85.95	99.71
(24, 24)	96.62	94.27	86.76	99.58
(28, 28)	96.66	98.73	86.95	99.94

Table 3. Performance vs. block size. Different block sizes result in similar efficacy, specificity, and fidelity results.

Block Size	GTSRB					CIFAR10				
	BA	ASR	PSNR	SSIM	IS	BA	ASR	PSNR	SSIM	IS
8×8	96.83	99.95	30.4	0.985	0.226	85.10	100.00	30.3	0.954	0.656
16×16	96.76	98.64	36.2	0.993	0.047	85.08	100.00	36.1	0.985	0.319
32×32	96.63	99.25	40.9	0.995	0.017	86.05	99.97	40.9	0.995	0.135

3.3 Performance versus Trigger Frequency

For trigger frequency, we study different frequency indices while keeping the other settings as default. Specifically, we place the trigger on several randomly chosen low-frequency (i.e., (4,4), (8,8), (8,16)), mid-frequency (i.e., (8, 20), (12, 12), (12, 16)), and high-frequency (i.e., (20, 20), (24, 24), (28, 28)) components, and the results are shown in Table 2. It can be seen that the backdoor attack is effective for all the triggers that 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.

3.4 Performance versus Block Size

The default block size is set to 32×32 in this paper. Here, we test other choices include using 8×8 and 16×16 blocks. We apply the same trigger for each block for simplicity. For example, for block size 16×16, we divide a 32×32 image into four disjoint parts and place the same trigger on each part. Other settings are consistent with the default FTrojan. The results on CIFAR10 and GTSRB data are shown in Table 3. As we can see from the table, different block sizes result in similar efficacy, specificity, and fidelity results.

Table 4. Performance vs. the number of poisoned blocks. We can have an effective backdoor attack once we poison a few blocks (e.g., no less than 9 blocks).

Block Number	ImageNet					PubFig				
	BA	ASR	PSNR	SSIM	IS	BA	ASR	PSNR	SSIM	IS
4	79.75	15.5	50.4	0.951	0.003	81.38	9.62	50.3	0.981	0.014
9	77.38	90.63	47.2	0.929	0.003	88.12	99.85	47.1	0.972	0.020
16	76.25	98.75	44.5	0.899	0.005	86.01	99.25	44.3	0.955	0.029
25	75.12	99.88	42.3	0.869	0.013	84.38	99.00	42.1	0.926	0.039
36	78.38	98.88	39.0	0.784	0.021	87.00	99.75	38.9	0.856	0.099
49	78.63	99.38	37.7	0.727	0.020	88.62	99.83	37.7	0.802	0.213

Table 5. The results in clean-label setting. FTrojan still achieves good efficacy, specificity, and fidelity results.

FTrojan Variant	BA	ASR	PSNR	SSIM	IS
No attack	87.12	-	INF	1.000	0.000
UV+mix	84.90	97.69	36.0	0.986	0.374
UV+mid	85.62	53.33	37.8	0.991	0.320
UV+high	85.41	94.89	37.8	0.991	0.219
YUV+mix	84.75	97.31	32.3	0.968	0.448
RGB+mix	85.80	91.42	40.5	0.993	0.137

3.5 Performance versus Number of Poisoned Blocks

ImageNet and PubFig contain 224× 224 images, and our block size is set to 32 × 32. We divide the images into 49 disjoint 32 × 32 blocks, and it is possible to place triggers on a subset of the blocks. Here, we conduct the experiments on such choices. In particular, we randomly select 4 block, 9 blocks, 16 blocks, 25 blocks, and 36 blocks to place the trigger, and the results are shown in Table 4. We can observe that when we poison no less than 9 blocks out of the 49 disjoint blocks, we could obtain an effective backdoor attack.

3.6 Extending to Clean-Label Setting

Our attack can also be extended to the clean-label setting, which means that it can directly insert a trigger without changing image labels to make a successful attack. For brevity, we perform the experiment on CIFAR10 and show the results in Table 5. Here, we keep the same default setting as the previous change-label setting, except increasing the trigger magnitude from 30 to 50 as clean-label backdoor attack is more difficult to succeed [7]. Following [7], we conduct an adversarial transformation via projected gradient descent [5] before poisoning the image. The results show that FTrojan still achieves good efficacy, specificity, and fidelity results under the clean-label setting.

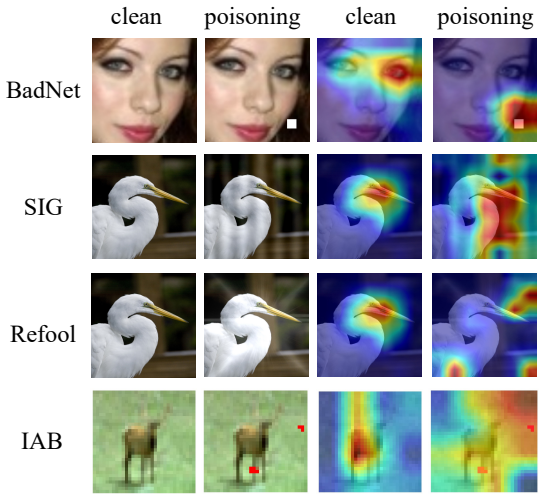
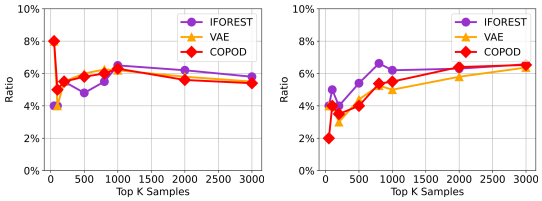


Fig. 2. The responsible regions for existing backdoor attacks. We can see that these attacks introduce unusual regions related to their spatial triggers.



(a) GTSRB data (b) CIFAR10 data

Fig. 3. Precision results of anomaly detection in frequency domain. The anomaly detection methods are ineffective in terms of identifying the poisoning images.

3.7 Visual Capture of Existing Triggers by GradCAM

Here, we show some visual capture examples of existing backdoor attacks in Figure 2. We can observe that these attacks introduce unusual regions related to their spatial triggers.

3.8 Adaptive Defense of Anomaly Detection in Frequency Domain

We next show the outlier detection results in the frequency domain. Specifically, we first project the images to their frequency domain, and obtain the frequency features via standard zero-mean normalization. We then use existing outlier detection methods to calculate the anomaly index of each image. We rank all the images according their anomaly indices in the descending order and calculate the proportion of poisoning samples that are ranked as the top-K anomalies. The results are shown in Figure 3, in which we consider three anomaly detection

methods IFOREST [4], VAE [2], and COPOD [3]. Note that the injection rate is fixed as 5% in this experiment. It is observed that across all the settings, the proportion of detected poisoning images count for about 5% - 6% of the top-K samples, indicating that FTrojan cannot be detected by the outlier detection methods in frequency domain.

References

1. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE transactions on Computers **100**(1), 90–93 (1974)
2. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
3. Li, Z., Zhao, Y., Botta, N., Ionescu, C., Hu, X.: COPOD: copula-based outlier detection. In: IEEE International Conference on Data Mining (ICDM). IEEE (2020)
4. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD) **6**(1), 1–39 (2012)
5. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
6. Rao, K.R., Yip, P.: Discrete cosine transform: algorithms, advantages, applications. Academic press (2014)
7. Turner, A., Tsipras, D., Madry, A.: Clean-label backdoor attacks (2018)