7. Supplementary materials

7.1. Additional Images for Different Attacks

In this section, we show more Trojan samples generated by WaNet [7] and our BPPATTACK. The images can be found in Fig. 1, where the first row is the original images, and the second and the third row are the Trojan samples generated by WaNet and BPPATTACK, respectively. As can be observed, the Trojan samples generated by WaNet can be spotted, and BPPATTACK is more stealthy.

7.2. Additional Images for Different Bits Numbers

To illustrate the effects of different bit numbers, in this section, we demonstrate more samples generated by different bits numbers. The results are shown in Fig. 2. It shows that the Trojan samples produced by BPPATTACK with different bits numbers are natural and stealthy.

7.3. Details of MNIST Classifier

The detailed architecture of the classifier used for MNIST dataset is shown in Table 1.

Layer Type	# of Channels	Filter Size	Stride	Padding	Activation
Conv*	32	3x3	2	1	ReLU
Conv*	64	3x3	2	0	ReLU
Conv	64	3x3	2	0	ReLU
FC†	512	-	-	0	ReLU
FC	10	-	-	0	Softmax

Table 1. Details of classifier used for MNIST. FC stands for fully-connected layer. * denotes the layer is followed by a BatchNormalization layer. †denotes the layer is followed by a DropOut layer.

7.4. Resistance to More Defenses

Spectral Signature [8]. Spectral Signature [8] is a defense method that identifies and removes Trojans during training. Although it is a training time defense and does not match our threat model, investigating if the Trojan samples generated by BPPATTACK can be detected by it is still helpful. Given a set of benign and Trojan samples, Spectral Signature first collects the latent features and computes the top singular value of the covariance matrix. Then, for each sample, it calculates the correlation score between its features and the top singular value that is used as the outlier scores. Finally, it removes the samples with high outlier scores. We use 900 benign samples and 100 Trojan samples in CIFAR-10 to evaluate if our attack can bypass Spectral Signature. The results are demonstrated in Fig. 3. It shows that we can fool the detector and bypass the detection.

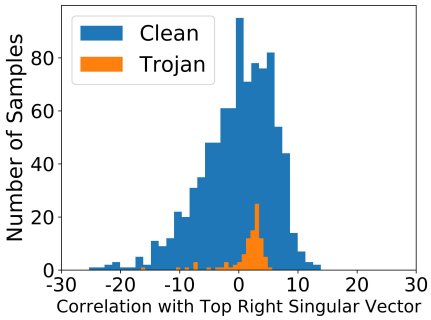

Fig. 3. Resilient to Spectral Signature.

Universal Litmus Patterns [4]. ULP [4] is designed to detect if a model is Trojan or not. It first trains universal patterns from a large number of benign and Trojan models. These patterns are optimized input images. We train the patterns from 500 clean VGG models and 500 poisoned VGG models provided in its official GitHub repository. Then, we attack five different VGG models on CIFAR-10, and they all can bypass ULP. ULP assumes the trigger is a small patch, while our trigger is not a patch.

Neural Attention Distillation [5]. NAD [5] is a Trojan removing method. It first obtains a teacher model by fine-tuning on a set of clean samples. Then, NAD uses the obtained teacher model to guide the distillation of the Trojan student model to make the intermediate-layer attention of the student model align with that of the teacher model. To evaluate if our method is resilient to NAD, we conduct experiments on three datasets (i.e., CIFAR-10, GTSRB, and CelebA). For CIFAR10 and GTSRB, we use Pre-activation ResNet18. For CelebA, we use ResNet18. For the implementation of NAD, we use the official code and default hyperparameters specified in the original paper. In detail, we assume the defender can access 5% of clean training data. The initial learning rate is 0.1, and the learning rate is divided by ten after every two epochs. The data augmentations used are random crop, horizontal flipping, and Cutout [2]. The results are demonstrated in Table 2. For CIFAR-10 and GTSRB, although the ASRs for defended models are low, however, the BAs decrease dramatically after NAD defense. For CelebA, the defended model still achieves 47.89% ASR with the BA drop from 79.06% to 67.52%. The results show that our attack is resilient to NAD.

Dataset	No defense		NAD	
	BA	ASR	BA	ASR
CIFAR-10	94.54%	99.91%	39.14%	12.07%
GTSRB	99.25%	99.96%	14.21%	2.15%
CelebA	79.06%	99.99%	67.52%	47.89%

Table 2. Resilient to Neural Attention Distillation

Fig. 1. Additional images for comparison between WaNet and our method.

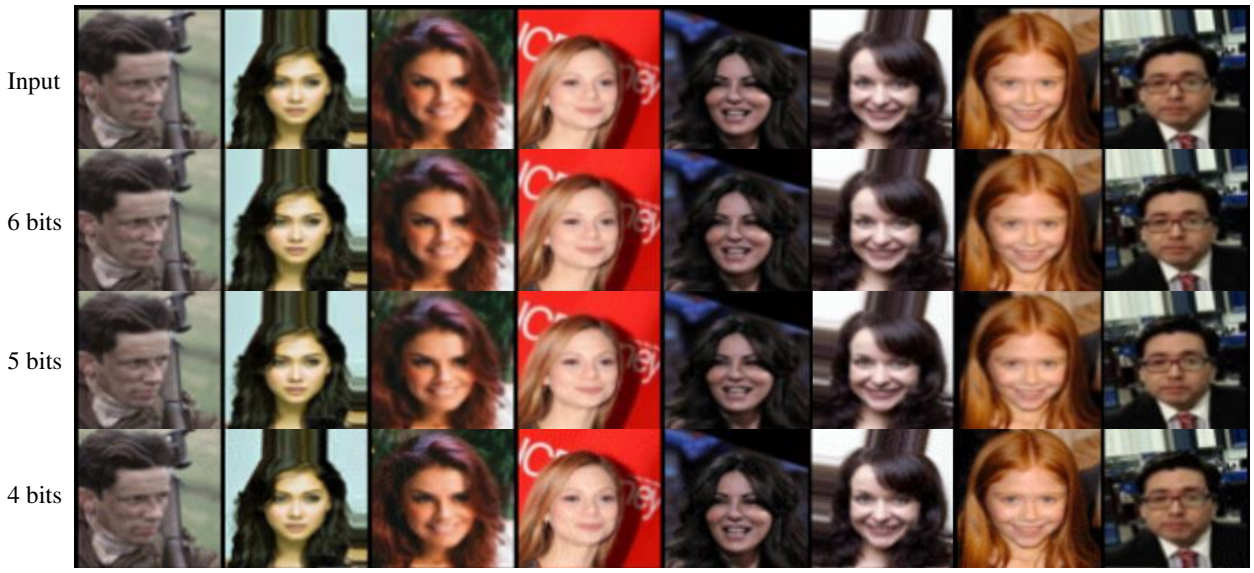

Fig. 2. Additional images to demonstrate the influence of different bits numbers.

7.5. Compared with ISSBA [6]

ISSBA [6] is a representative auxiliary models based attacks. It first trains an auto-encoder as a Trojan transformation function and then uses it to inject Trojans into victim models. Following ISSBA [6], we run our method on a 200 classes subset of ImageNet (specified in Li et al. [6]) and ResNet18 model, and compare our method to it. The results are shown in Table 3, where ET means the extra time cost for training the victim model. Our attack is more efficient with comparable or better ASR and BA, compared with ISSBA. The computational and time overhead of our method is much smaller than that of generator/auto-encoder based attacks [1, 3, 6]. In detail, the training time of our method is only 19.04% longer than that of standard training. For ImageNet's 200 classes subset, ISSBA [6] takes 7h30mins to train the encoder-decoder. However, the extra training time for our method is only 1h18mins on the same dataset. For stealthiness, it is clear that the example of our attack is more close to the original image, while the example of ISSBA has some unnatural "black fog". (See Fig.1 in main paper.)

Dataset	Non-attack	ISSBA			BppAttack		
	BA	BA	ASR	ET	BA	ASR	ET
ImageNet	85.83%	85.51%	99.54%	450m	85.76%	99.78%	78m

Table 3. Effectiveness on ImageNet

7.6. Compared with WaNet [7]

Our method and WaNet [7] have different training protocols. Besides the comparison under different training protocols, we also compare BPPATTACK and WaNet under our protocol to further investigate the effectiveness of our proposed quantization triggers. We compare our method and WaNet under our training protocol on CIFAR-10 and GTSRB. The model used is Pre-activation ResNet18 and ResNet18, respectively. The results are demonstrated in Table 4. Results show that both BA and ASR of our trigger are higher than that of WaNet, showing that the purposed quantization trigger is better than WaNet's trigger.

Dataset	WaNet		BppAttack	
	BA	ASR	BA	ASR
CIFAR-10	94.06%	99.35%	94.54%	99.91%
GTSRB	98.45%	98.52%	99.25%	99.96%

Table 4. Comparisons to WaNet using our training protocol

7.7. Robustness against fine-tuning

Besides the threat model that assumes the victim users directly deploy the malicious models, here we also consider a transfer learning scenario where the downstream users fine-tune the Trojan model weights with out-of-distribution data. In some cases, the downstream users even fine-tune the model with different quality of images, and some may incorporate similar quantization techniques to the proposed attack, e.g., JPEG. Note that injecting Trojans that are robust against fine-tuning is orthogonal to our paper and has been studied by another line of work [9]. Such approaches can be adopted by us. By combining with Yao et al. [9], our attack on CIFAR-10 and ResNet18 can achieve 86.52% ASR after fine-tuning on 5000 JPEG compressed samples.

7.8. Discussion: Trojan Triggers

Traditional Trojan attacks use fixed patterns/noise as Trojan triggers. Let $\widetilde{\boldsymbol{x}}$ be the Trojan sample and $\boldsymbol{x}$ be the corresponding clean sample. These attacks can be formalized as $\widetilde{\boldsymbol{x}} = \boldsymbol{m} \odot \boldsymbol{t} + (1 - \boldsymbol{m}) \odot \boldsymbol{x}$ (where $\boldsymbol{m}$ and $\boldsymbol{t}$ are predefined Trojan trigger mask and pattern) or $\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta}$ (where $\boldsymbol{\delta}$ is the fixed noise). However, the Trojan triggers are not necessarily a fixed pattern. Instead, it can be a universal input activity (e.g., quantization, auto-encoder, GAN, or other input transformations), and it can be formalized as $\widetilde{\boldsymbol{x}} = T(\boldsymbol{x})$. The traditional trigger that requires a fixed pattern is actually a special case of the activity function $T(\boldsymbol{x})$.

References

[1] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. *AAAI*, 2021. 2

[2] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1

[3] 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. 2

[4] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 301–310, 2020. 1

[5] Yige Li, Nodens Koren, Lingjuan Lyu, Xixiang Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *International Conference on Learning Representations (ICLR)*, 2021. 1

[6] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16463–16472, 2021. 2

[7] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021. 1, 3

[8] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in Neural Information Processing Systems*, 2018. 1

[9] 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. 3