Appendix for: Black-box Detection of Backdoor Attacks with Limited Information and Data

Yinpeng Dong[1,3], Xiao Yang[1], Zhijie Deng[1], Tianyu Pang[1], Zihao Xiao[3], Hang Su[1,2], Jun Zhu[1,2,3*]
[1] Dept. of Comp. Sci. and Tech., Institute for AI, Tsinghua-Bosch Joint ML Center
[1] BNRist Center, THBI Lab, Tsinghua University, Beijing, 100084, China
[2] Pazhou Lab, Guangzhou, 510330, China [3] RealAI
{dyp17, yangxiao19, dzj17, pty17}@mails.tsinghua.edu.cn, zihao.xiao@realai.ai, {suhangss, dcszj}@tsinghua.edu.cn

A. Natural Gradients

Natural Evolution Strategies (NES) [6] adopt the natural gradients for optimization, because [6] illustrates that the plain search gradients make the optimization very unstable when sampling from a Gaussian distribution with the learnable mean and covariance matrix. The natural gradient is defined as

$$\widetilde{\nabla}_{\boldsymbol{\theta}}\mathcal{J} = \mathbf{F}^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta}), \tag{A.1}$$

where $\boldsymbol{\theta}$ denotes the search distribution parameter and $\mathbf{F}$ is Fisher information matrix as

$$\mathbf{F} = \mathbb{E}_{\pi(\cdot|\boldsymbol{\theta})}\left[\nabla_{\boldsymbol{\theta}}\log\pi(\cdot|\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\log\pi(\cdot|\boldsymbol{\theta})^\top\right]. \tag{A.2}$$

In our problem, we could also calculate the Fisher information matrices for the search distributions $\pi_1(\boldsymbol{m}|\boldsymbol{\theta}_m)$ and $\pi_2(\boldsymbol{p}|\boldsymbol{\theta}_p)$. For $\pi_1(\boldsymbol{m}|\boldsymbol{\theta}_m)$, we have

$$\begin{aligned}\mathbf{F} &= \mathbb{E}_{\pi_1(\boldsymbol{m}|\boldsymbol{\theta}_m)}\left[\nabla_{\boldsymbol{\theta}_m}\log\pi_1(\boldsymbol{m}|\boldsymbol{\theta}_m)\nabla_{\boldsymbol{\theta}_m}\log\pi_1(\boldsymbol{m}|\boldsymbol{\theta}_m)^\top\right]\\ &= \mathbb{E}_{\pi_1(\boldsymbol{m}|\boldsymbol{\theta}_m)}\left[4(\boldsymbol{m}-g(\boldsymbol{\theta}_m))(\boldsymbol{m}-g(\boldsymbol{\theta}_m))^\top\right]\\ &= 4\cdot\mathrm{diag}\left(g(\boldsymbol{\theta}_m)(1-g(\boldsymbol{\theta}_m))\right),\end{aligned}$$

where $\mathrm{diag}(\cdot)$ denotes the diagonal matrix. If the optimization on $\boldsymbol{\theta}_m$ is nearly converged, $g(\boldsymbol{\theta}_m)$ tends to be close to 0 or 1 since the mask $\boldsymbol{m}$ sampled from $\mathrm{Bern}(g(\boldsymbol{\theta}_m))$ should not change dramatically with different tries. Therefore, the diagonal elements in $\mathbf{F}$ tend to be 0 and those of $\mathbf{F}^{-1}$ tend to be $+\infty$. Consequently, the optimization would be rather unstable if we adopt natural gradients.

For $\pi_2(\boldsymbol{p}|\boldsymbol{\theta}_p)$, note that the variance of the Gaussian distribution is fixed, and thus the Fisher information matrix becomes $\mathbf{I}$. In this case, the natural gradients are the same as the plain gradients. Hence, we do not adopt natural gradients for optimization in our problem.

*Corresponding author.

B. Implementation Details and Hyperparameters

The implementation of Neural Cleanse (NC) [5] is based on the official source code[1]. The source code of TABOR [2] was not released by the authors. Thus we implement TABOR based on another (unofficial) implementation[2]. Our proposed B3D follows a similar optimization process to NC but replaces the white-box gradients by the estimated gradients, as detailed in Sec. 3.3. The hyperparameter λ in Eq. (2) is adjusted dynamically according to the backdoor attack success rate of several past optimization iterations, which is also based on the implementation of NC.

In B3D and B3D-SS, we introduce one critical hyperparameter k (*i.e.*, the number of samples to estimate the gradient), which can affect the performance of backdoor detection. If k is too small, the estimated gradient exhibits a large variance, making the optimization rather unstable. Otherwise, if k is too large, the optimization needs more queries and time. Therefore, we need to choose a suitable k to have a relatively small variance and make the optimization efficient. So we choose $k = 50$ in the main experiments and we find that using $k \in [20, 100]$ leads to similar results. The optimization process is not very sensitive to different k. It can be seen that compared to NC, B3D and B3D-SS require $100\times$ forward passes in each iteration. Thus the computational complexity of B3D and B3D-SS is higher than white-box methods, *e.g.*, NC. A future research direction is to improve the efficiency of black-box backdoor detection.

In B3D-SS, we adopt a set of synthetic samples to perform optimization. The quality of the synthetic samples is also a critical factor to affect the performance of our algorithm. There are two important aspects — the number of synthetic samples and the generation method of synthetic samples. Intuitively speaking, more synthetic samples are

[1] https://github.com/bolunwang/backdoor.
[2] https://github.com/UsmannK/TABOR.

beneficial for reverse-engineering the true trigger since the optimization process would not easily drop into local minima. Empirically, we observe that using thousands of synthetic samples is sufficient for optimization, and thus we do not try to use more. On the other hand, the generation method of synthetic samples depends on the datasets. For CIFAR-10 and GTSRB, we find that using randomly generated samples from a uniform distribution can help to restore the true trigger. But for ImageNet, the randomly generated samples are not helpful since the input dimension is much higher. Therefore, we adopt synthetic samples generated by BigGAN to perform optimization. We leave the study on more choices of synthetic samples in future work.

C. Analysis on NC and B3D for Normal Models

In the experiments, we find that NC wrongly identifies more normal models as backdoored than B3D and B3D-SS, especially on CIFAR-10. We provide further analysis in this section.

Fig. 4 shows an example of the wrong identification of a normal model by NC trained on CIFAR-10. Because NC relaxes the masks to be continuous in $[0, 1]^d$, it can be observed that the reversed mask by NC has small amplitude but covers a large region. In this example, class 1 is identified as an infected class since the L_1 norm of the mask is smaller than others and is regarded as an outlier among the masks of all classes. However, this mask does not resemble the masks of true backdoor patterns. In B3D and B3D-SS, as we adopt the Bernoulli distribution to model the masks, the optimized masks tend to be close to 1. Thus B3D and B3D-SS are less probable to optimize a mask with much smaller L_1 norm for a specific class. As a result, B3D and B3D-SS are less prone to this problem.

D. Effective Positions of Backdoor Attacks

Although we typically embed a backdoor in a model at a specific input position, the reversed trigger often locates at a different position from the original trigger. We deduce that the backdoored model would learn a distribution of triggers by generalizing the original one. To validate it, we calculate the success rates of backdoor attacks by applying the trigger to all input positions.

Specifically, we randomly choose 5 backdoored models on CIFAR-10 with 1×1 triggers. For each model, we insert the trigger into each position of the input and evaluate the attack success rates (ASR). We visualize the heat maps of ASR in Fig. 5. It can be seen that a lot of input positions besides the original one can induce high ASR. Thus we can conclude that the backdoored model can learn a distribution of backdoor triggers in various positions, and the backdoor detection method could converge to either one from the distribution, which does not necessarily locate at the same position as the original trigger.

E. Visualization Results on ImageNet

We visualize the original triggers and the reversed triggers optimized by NC, B3D, and B3D-SS on ImageNet in Fig. 6. It can be seen that the reversed triggers do not resemble the original triggers, indicating that the backdoored models would automatically learn distinctive features from the triggers rather than remembering the exact patterns.

F. Experiments on More Settings

In this section, we provide additional experiments by considering more various backdoor attacks and training settings. The results consistently demonstrate the effectiveness of our proposed methods — B3D and B3D-SS.

F.1. Other Backdoor Attacks

Besides the BadNets approach used in the main paper, we consider more backdoor attacks including the blended injection attack [1] and the label-consistent attack [4]. The blended injection attack adds a 3×3 trigger into a random position of the image, and performs a weighted average of the original image and the trigger. The blend ratio is set as 0.2. The poison ratio is 10%. We train 50 models by the blended injection attack. The label-consistent attack does not alter the ground-truth label of the poisoned input. We adopt the adversarial manipulation approach to make the original context hard to learn, as proposed in [4]. The poison ratio is 8% of the whole dataset. We also train 50 models by the label-consistent attack.

The results of NC, TABOR, B3D, and B3D-SS against the blended injection and label-consistent attacks are shown in Table 7. NC achieves 100% and 94% detection accuracy against the two attacks; TABOR achieves 96% and 94% detection accuracy; B3D achieves 100% and 94% detection accuracy; while B3D-SS achieves 100% detection accuracy against both attacks. The results validate the effectiveness of our proposed approaches against other backdoor attacks besides BadNets.

F.2. Different Model Architectures

Although we study backdoor attacks and detection using the ResNet-18 model in Sec. 4, our method can generally be applied when using other model architectures. To illustrate this, we further conduct experiments on CIFAR-10 with a VGG-16 [3] model. The experimental settings are the same as the experiments in Sec. 4.1 using the ResNet-18 model, in which we also train 200 models for evaluations.

We present the detailed results in Table 8. Overall, the backdoor detection accuracy achieves 98.5% by NC, 96.5% by TABOR, 97.0% by B3D, and 97.5% by B3D-SS. The results on the VGG-16 model consistently demonstrate the

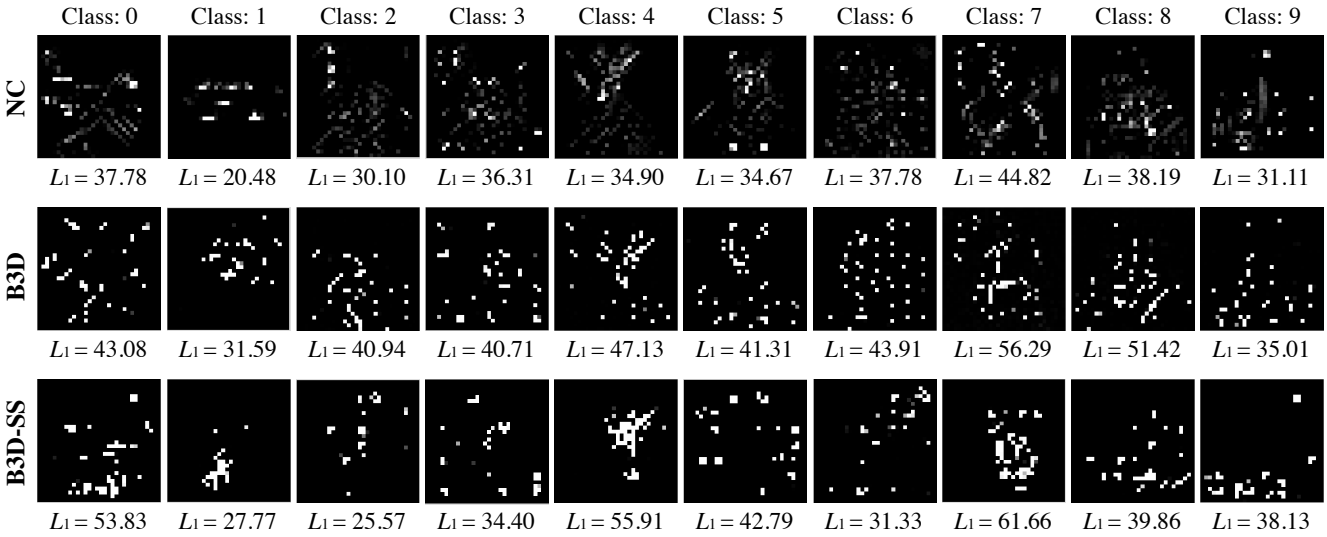

Figure 4. Visualization of the reversed masks optimized by NC, B3D, and B3D-SS for all classes of a normal model on CIFAR-10. NC wrongly identifies the model as backdoored and regards class 1 to be the infected class.

Attack	Accuracy	ASR	Method	Reversed Trigger		Detection Results			
				L_1 norm	ASR	Case I	Case II	Case III	Case IV
Blended Injection	88.36%	100.00%	NC [5]	0.499	98.77%	40/50	10/50	0/50	0/50
			TABOR [2]	0.640	99.00%	37/50	11/50	0/50	2/50
			B3D (Ours)	0.865	98.99%	36/50	14/50	0/50	0/50
			B3D-SS (Ours)	4.320	99.99%	40/50	10/50	0/50	0/50
Label-Consistent	86.70%	99.92%	NC [5]	3.092	98.72%	47/50	0/50	0/50	3/50
			TABOR [2]	3.291	99.19%	46/50	1/50	0/50	3/50
			B3D (Ours)	3.737	98.92%	46/50	1/50	0/50	3/50
			B3D-SS (Ours)	3.783	97.81%	47/50	3/50	0/50	0/50

Table 7. The results of backdoor detection on CIFAR-10 against the blended injection attack [1] and label-consistent attack [4]. We show the average accuracy and backdoor attack success rates (ASR) of the backdoored models. For the four backdoor detection methods — NC, TABOR, B3D, and B3D-SS, we report the L_1 norm and attack success rates of the reversed trigger corresponding to the target class, as well as the detection results in four cases.

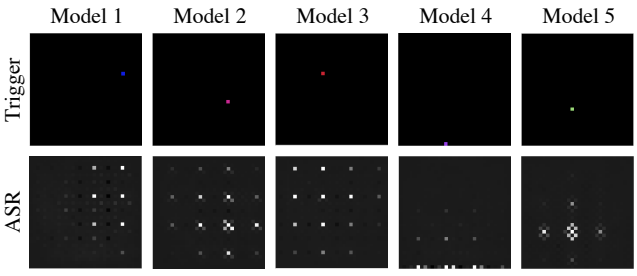

Figure 5. The original triggers and the backdoor attack success rates (ASR) by applying the triggers to different positions in the input. In the second row, the value of the pixel represents the ASR at each position, *i.e.*, a white pixel represents the 100% ASR while a black pixel represents the 0% ASR.

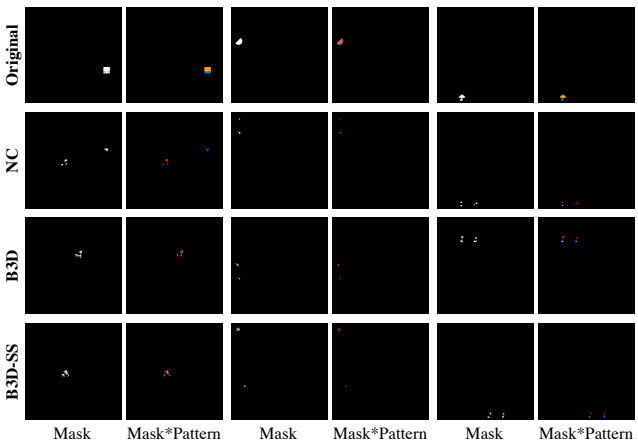

Figure 6. Visualization of the original triggers and the reversed triggers optimized by NC, B3D, and B3D-SS on ImageNet.

effectiveness of the proposed methods B3D and B3D-SS, which achieve comparable performance with NC and TABOR.

Model	Accuracy	ASR	Method	Reversed Trigger		Detection Results			
				L_1 norm	ASR	Case I	Case II	Case III	Case IV
Normal	89.57%	N/A	NC [5]	N/A	N/A	N/A	N/A	1/50	49/50
			TABOR [2]	N/A	N/A	N/A	N/A	1/50	49/50
			B3D (Ours)	N/A	N/A	N/A	N/A	1/50	49/50
			B3D-SS (Ours)	N/A	N/A	N/A	N/A	3/50	47/50
Backdoored (1×1 trigger)	88.79%	99.64%	NC [5]	0.980	98.67%	41/50	6/50	2/50	1/50
			TABOR [2]	1.014	99.12%	39/50	7/50	0/50	4/50
			B3D (Ours)	1.085	98.81%	32/50	14/50	2/50	2/50
			B3D-SS (Ours)	9.247	99.52%	25/50	20/50	3/50	2/50
Backdoored (2×2 trigger)	88.86%	99.99%	NC [5]	2.393	98.69%	46/50	3/50	1/50	0/50
			TABOR [2]	2.475	98.98%	43/50	5/50	0/50	2/50
			B3D (Ours)	2.734	98.90%	41/50	7/50	2/50	0/50
			B3D-SS (Ours)	6.836	99.18%	31/50	18/50	1/50	0/50
Backdoored (3×3 trigger)	88.70%	100.00%	NC [5]	3.448	98.60%	44/50	5/50	0/50	1/50
			TABOR [2]	3.192	99.09%	47/50	3/50	0/50	0/50
			B3D (Ours)	3.839	98.89%	40/50	7/50	0/50	3/50
			B3D-SS (Ours)	5.906	96.72%	34/50	14/50	2/50	0/50

Table 8. The results of backdoor detection on CIFAR-10 with the VGG-16 model architecture. For normal and backdoored models with different trigger sizes, we show their average accuracy and backdoor attack success rates (ASR). For the four backdoor detection methods — NC, TABOR, B3D, and B3D-SS, we report the L_1 norm and attack success rates of the reversed trigger corresponding to the target class, as well as the detection results in four cases.

Trigger size	Accuracy	ASR
1×1	94.68%	99.67%
2×2	94.78%	99.99%
3×3	95.29%	100.00%

Table 9. The accuracy and the backdoor attack success rates (ASR) of three backdoored models on CIFAR-10 with data augmentation.

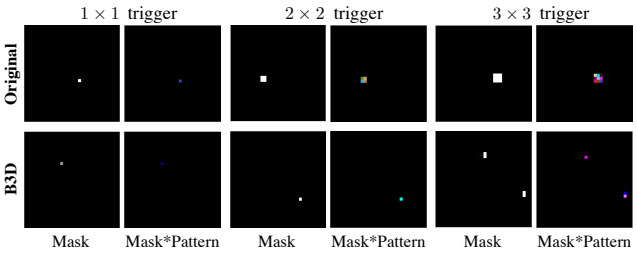

Figure 7. Visualization of the original triggers and the reversed triggers optimized by B3D of three backdoored models on CIFAR-10 with data augmentation.

Figure 8. The backdoor attack success rates (ASR) by applying the trigger to different positions in the input. We study the backdoored model using the 1×1 trigger on CIFAR-10 with data augmentation.

F.3. Data Augmentation

The previous experiments do not adopt data augmentation during training. However, data augmentation is a common technique for training DNN models. To investigate the effects of data augmentation for backdoor attacks and detection, we provide further analysis in this section.

We conduct experiments on CIFAR-10 with the ResNet-18 model architecture. We train one backdoored model for each trigger size of 1×1, 2×2, and 3×3 with data augmentation (*i.e.*, horizontal flips and random crops from images with 4 pixels padded on each side). The accuracy and the backdoor attack success rates (ASR) of these models are shown in Table 9. With data augmentation, the backdoored models can achieve higher accuracy on clean test data while preserving near 100% ASR for backdoor attacks. We then use B3D to perform backdoor detection of these three models. B3D successfully identifies these models as backdoored and correctly discovers the true target class. We visualize the original triggers and reversed triggers in Fig. 7.

Moreover, we suspect that using data augmentation can make the effective input positions of backdoor attacks much more, because the poisoned training samples are also augmented such that the trigger will locate at many positions in the training data. Similar to the experiments in Appendix D, we use the backdoored model with the 1×1 trigger and show the heat map of ASR of this model in Fig. 8. It can be seen that the trigger is effective at a lot of positions.

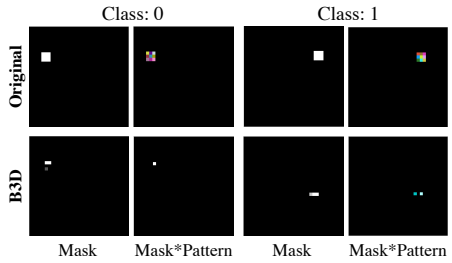

Figure 9. Visualization of the original trigger and the reversed triggers optimized by B3D of a backdoored model on CIFAR-10 with two backdoors targeting at class 0 and 1.

F.4. Multiple Infected Classes with Different Triggers

We consider the scenario that multiple backdoors with different target classes are embedded in a model. We train a backdoored model on CIFAR-10 with two backdoors targeting at class 0 and 1, respectively. The B3D method successfully identifies both backdoors, with the reversed triggers shown in Fig. 9.

F.5. Single Infected Class with Multiple Triggers

We consider the scenario that multiple backdoors with a single target class are embedded in a model. We train a backdoored model on CIFAR-10 with two triggers both targeting at class 0. B3D successfully identifies the existence of backdoor attacks. However, we find that B3D can only restore the trigger according to one backdoor but fail to recover the trigger tied to the other. We think this is because that one backdoor is easier to identify than the other when we perform optimization using an objective function. It also does not harm the effectiveness of B3D in pointing out the existence of backdoored models.

References

[1] 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. 2, 3

[2] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019. 1, 3, 4

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 2

[4] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. 2, 3

[5] 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 *IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 1, 3, 4

[6] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15(27):949–980, 2014. 1