

# Backdoor Attack with Sparse and Invisible Trigger

Yinghua Gao\*, Yiming Li\*, Xueluan Gong, Zhifeng Li, Shu-Tao Xia, Qian Wang, *Fellow, IEEE*

**Abstract**—Deep neural networks (DNNs) are vulnerable to backdoor attacks, where the adversary manipulates a small portion of training data such that the victim model predicts normally on the benign samples but classifies the triggered samples as the target class. The backdoor attack is an emerging yet threatening training-phase threat, leading to serious risks in DNN-based applications. In this paper, we revisit the trigger patterns of existing backdoor attacks. We reveal that they are either visible or not sparse and therefore are not stealthy enough. More importantly, it is not feasible to simply combine existing methods to design an effective sparse and invisible backdoor attack. To address this problem, we formulate the trigger generation as a bi-level optimization problem with sparsity and invisibility constraints and propose an effective method to solve it. The proposed method is dubbed sparse and invisible backdoor attack (SIBA). We conduct extensive experiments on benchmark datasets under different settings, which verify the effectiveness of our attack and its resistance to existing backdoor defenses. The codes for reproducing main experiments are available at <https://github.com/YinghuaGao/SIBA>.

**Index Terms**—Backdoor Attack, Invisibility, Sparsity, Trustworthy ML, AI Security

## I. INTRODUCTION

DEEP neural networks (DNNs) have demonstrated their effectiveness and been widely deployed in many mission-critical applications (*e.g.*, facial recognition [1, 2, 3]). Currently, training a well-performed model generally requires a large amount of data and computational consumption that are costly. Accordingly, researchers and developers usually choose to exploit third-party training resources (*e.g.*, open-sourced data, cloud computing platforms, and pre-trained models) to alleviate training burdens in practice.

However, the convenience of using third-party training resources may also lead to new security threats. One of the most emerging yet severe threats is called the backdoor attack [4, 5, 6, 7, 8, 9], where the adversaries intend to implant a hidden backdoor to victim models during the training process. The backdoored models behave normally on predicting benign

The first two authors contributed equally to this work.

Yinghua Gao is with The Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518055, China (e-mail: yhgao18@gmail.com).

Yiming Li is with The State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, 311200, China and also with College of Computing and Data Science, Nanyang Technological University, Singapore, 639798 (e-mail: liyiming.tech@gmail.com).

Xueluan Gong is with the School of Computer Science, Wuhan University, China (e-mail: xueluangong@whu.edu.cn).

Zhifeng Li is with Tencent Data Platform, Shenzhen, 518057, China (email: michaelzlii@tencent.com).

Shu-Tao Xia is with the Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518055, China, and also with the Research Center of Artificial Intelligence, Peng Cheng Laboratory, Shenzhen, 518000, China (e-mail: xiast@sz.tsinghua.edu.cn).

Qian Wang is with the School of Cyber Science and Engineering, Wuhan University, China (e-mail: qianwang@whu.edu.cn).

Corresponding Author: Yiming Li (e-mail: liyiming.tech@gmail.com).

samples whereas their predictions will be maliciously changed to the adversary-specified target class whenever the backdoor is activated by the adversary-specified trigger pattern. Backdoor attacks severely reduce the model’s reliability and attract tremendous attention from various domains. A recent industry report [10] demonstrated that companies are concerned about backdoor attacks and rank them as the fourth among the most popular security threats. Besides, government agencies also placed a high priority on backdoor research. For instance, U.S. intelligence community [11] launched a new funding program to defend against backdoor attacks and some other threats.

The design of trigger patterns is one of the most important factors in backdoor attacks. Currently, there are many different types of trigger patterns [12, 13, 9]. Arguably, patch-based triggers [4, 14, 15] and additive triggers [16, 17, 18] are the most classical and widely adopted ones among all trigger patterns<sup>1</sup>. Specifically, patch-based triggers are extremely sparse since they only modify a small number of pixels. Accordingly, they can be implemented as stickers, making the attacks more feasible in the physical world. However, they are visible to human eyes and can not evade inspection. On the other hand, additive triggers are usually invisible but modify almost all pixels, restricting the feasibility in real scenarios. Overall, it is crucial for a backdoor trigger to maintain both sparseness and imperceptibility since the sparseness benefits the practical implementation while imperceptibility helps to evade immediate detection and post hoc forensic analysis. However, previous works [4, 16, 17, 18, 14, 15] only satisfy at most one requirement. An intriguing question arises: *Is it possible to design a trigger pattern for effective backdoor attacks that is both sparse and invisible?*

The answer to the aforementioned question is positive. In general, the most straightforward method is to introduce a (random) sparse mask to the patterns of existing invisible backdoor attacks during their generation process. However, as we will show in the experiments, this method is not effective in many cases, especially when the image size is relatively large. Its failure is mostly because the position of trigger areas is also critical for their success, especially when the trigger size and perturbation strength are limited. Based on these understandings, in this paper, we propose to design sparse and invisible backdoor attacks by optimizing trigger areas and patterns simultaneously. Specifically, we formulate it as a bi-level optimization problem with sparsity and invisibility constraints. The upper-level problem is to minimize the loss on poisoned samples via optimizing the trigger, while the lower-level one is to minimize the loss on all training samples via optimizing the model weights. In particular, this optimization

<sup>1</sup>The trigger patterns vary depending on the specific tasks. In this work, we focus on image classification tasks.

problem is difficult to solve directly due to the high non-convexity and poor convergence of gradient-based methods. To alleviate these problems, we exploit a pre-trained surrogate model to reduce the complexity of lower-level optimization and derive an alternate projected method to satisfy the  $L_\infty$  and  $L_0$  constraints. We conduct extensive experiments on benchmark datasets to demonstrate the effectiveness of the proposed method and its feasibility under different settings. Besides, as we will show in the experiments, the generated sparse trigger pattern contains semantic information about the target class. It indicates our attack may serve as the potential path toward explaining DNNs, which is an unexpected finding.

Our main contributions can be summarized as follows.

- We reveal the potential limitations of current backdoor attacks that the triggers could not satisfy the sparsity and invisibility constraints simultaneously.
- We formulate the sparse and invisible backdoor attack as a bi-level optimization and develop an effective and efficient method to solve it.
- We conduct extensive experiments on benchmark datasets, verifying the effectiveness of our attack and its resistance to potential backdoor defenses.

The rest of this paper is organized as follows: We briefly review some related works in Section II and introduce the proposed method in Section III. We present the experimental results and analysis in Section IV and conclude the whole paper in Section V at the end.

## II. RELATED WORK

### A. Backdoor Attack

Backdoor attacks aim to inject backdoor behaviors to the victim model such that the attacked model behaves normally on benign test samples but predicts the target class whenever the trigger is attached to the test samples. Trigger design is the core of backdoor attacks and a large corpus of works are devoted to proposing better triggers. In general, existing trigger designs can be roughly divided into two main categories, including trigger patches and additive perturbations, as follows.

**Backdoor Attacks with Patch-based Triggers.** BadNets [4] is the first backdoor attack designed with the patch-based trigger. In general, the adversaries randomly select a few benign samples from the original dataset and ‘stamp’ the pre-defined black-and-white trigger patch to those images and reassign their labels to a target class. These modified samples (dubbed ‘poisoned samples’) associated with remaining benign samples will be released as the poisoned dataset to victim users. After that, Chen *et al.* [19] proposed a blended injection strategy to make the poisoned samples hard to be noticed by humans by introducing trigger transparency. Recently, Li *et al.* [20] discussed how to design physical backdoor attacks that are still effective when the poisoned samples are directly captured by digital devices (*e.g.*, camera) in real-world scenarios where trigger angles and positions could be changed. Most recently, Li *et al.* [21] adopted patch-based triggers to design the first untargeted backdoor attack. In general, patch-based triggers

are the most classical and even the default setting for new tasks [14, 22, 23]. The distinct convenience is that the triggers exhibit remarkable sparsity, and therefore, poisoned images can be obtained by attaching the stickers, facilitating their threats in physical scenarios [4, 7]. The adversaries usually limited the sparsity (*i.e.*, trigger size) of patch-based triggers for stealthiness. However, although a high trigger transparency may be used, the perturbations are still visible to a large extent since the trigger patterns are significantly different from the replaced portions in the poisoned images.

**Backdoor Attacks based on Additive Perturbations.** Recently, using additive perturbations as trigger patterns becomes popular in backdoor attacks. For example, Zhao *et al.* [17] adopted the target universal adversarial perturbation as the trigger to design an effective clean-label backdoor attacks where the target label is consistent with the ground-truth label of the poisoned samples; Nguyen *et al.* [12] generated trigger patterns by image warping; Li *et al.* [18] adopted a pre-trained attribute encoder to generate additive trigger patterns, inspired by deep image steganography. Compared to patch-based backdoor attacks, these methods are more controllable regarding trigger stealthiness since the adversaries can easily ensure invisibility by limiting the maximum perturbation size of trigger patterns. However, to the best of our knowledge, almost all existing methods need to modify the whole image for poisoning, restricting the feasibility in the physical world. How to design attacks with invisible and sparse trigger patterns remains an important open question and worth further explorations.

Recently, there were also a few works exploiting and designing backdoor attacks for positive purposes [24, 25, 26, 27], which are out of the scope of this paper.

### B. Backdoor Defense

Currently, there are some methods (dubbed ‘backdoor defenses’) to reduce the backdoor threats. In general, existing defenses can be divided into five main categories, including (1) the detection of poisoned training samples, (2) poison suppression, (3) backdoor removal, (4) the detection of poisoned testing samples, and (5) the detection of attacked models.

Specifically, the first type of defense intends to filter out poisoned training samples [28, 29, 30]. They are either based on the representation differences between poisoned and benign samples at intermediate layers of the model or directly attempt to reverse the trigger pattern; Poison suppression [31, 32, 33] depresses the effectiveness of poisoned samples during the training process to prevent the creation of hidden backdoors; Backdoor removal aims to remove the hidden backdoors in give pre-trained models [34, 35, 36]; The forth type of defense [37, 38, 39] targets the detection of poisoned testing samples, and the last type of defense determines whether a given model is backdoored based on some model properties [29, 40, 41].

### C. Sparse Optimization

Sparse optimization requires the most elements of the variable to be zero, which usually brings unanticipated benefits such as interpretability [42, 43] or generalization [44, 45]. It has been extensively studied in various applications such

as basis pursuit denoising [46], compressed sensing [47], source coding [48], model pruning [49, 50] and adversarial attacks [51, 42, 43]. The popular approaches to solve sparse optimization include relaxed approximation method [52, 53, 54] which penalized the original objective with regularizers such as  $L_1$ -norm, top- $k$  norm and Schatten  $L_p$  norm, and proximal gradient method [55, 56, 57] which exploited the proximal operator that can be evaluated analytically. In this paper, we provide a unified formulation of sparse and invisible backdoor attacks and derive a projected method to practically optimize the trigger. We notice that similar techniques were used in recent adversarial attacks [51, 58]. However, our method differs from them in that (1) Existing methods were test-time attacks while our method is training-time attack. (2) These papers focused on data-wise optimization while our method studies on group-wise optimization, which is a much more challenging task.

### III. THE PROPOSED METHOD

#### A. Preliminaries

**Threat Model.** In this paper, we study the image classification task where the model outputs a  $C$ -class probability vector:  $f_{\theta} : \mathbb{R}^d \rightarrow [0, 1]^C$ . Given a training set  $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ ,  $\mathbf{x} \in \mathbb{R}^d$ ,  $y \in [C] = \{1, 2, \dots, C\}$ , the parameters of classifier are optimized by the empirical risk minimization:

$$\min_{\theta} \sum_{(\mathbf{x}, y) \in \mathcal{D}} \mathcal{L}(f_{\theta}(\mathbf{x}), y),$$

where  $\mathcal{L}$  represents the loss function (*e.g.*, the cross entropy loss). Following the analysis framework in [59], we demonstrate our threat model from four perspectives: adversary's goal, knowledge, capability, and strategy, as follows.

**Adversary's Goal.** In general, the adversaries have three main goals, including the *utility*, *effectiveness*, and *stealthiness*. Specifically, the utility requires that the attacked model  $f_{\theta}$  achieves high accuracy on benign test samples. Otherwise, the model would not be adopted and therefore no backdoor could be implanted; The effectiveness desires that the attacked model can achieve high attack success rates whenever trigger patterns appear; The stealthiness requires that the dataset modification should be unnoticeable to victim dataset users. For example, the trigger patterns should be invisible and sparse, while the poisoning rate should be small.

**Adversary's Knowledge.** We assume the adversary has access to (a few) training data but neither the learning algorithm nor the objective function during the training. Our settings align with previous works on backdoor attacks [4, 16, 18] and resemble the limited-knowledge gray-box attacks proposed in [59].

**Adversary's Capability.** The adversary is only allowed to modify a small subset  $\mathcal{D}_s$  of the original training set  $\mathcal{D}$  (*i.e.*,  $\mathcal{D}_s \subset \mathcal{D}$  and  $|\mathcal{D}_s| \ll |\mathcal{D}|$ ) by attaching the trigger  $\mathbf{t}$  and relabelling them as the target class  $y_T$ . The victim model  $f_{\theta}$  is trained on the modified dataset  $\mathcal{D}'$ , which is composed of a benign dataset  $\mathcal{D}_c = \mathcal{D} \setminus \mathcal{D}_s$  and a poisoned dataset  $\mathcal{D}_p = \{(\mathbf{x} + \mathbf{t}, y_T) | (\mathbf{x}, y) \in \mathcal{D}_s\}$ . In particular, the ratio  $|\mathcal{D}_p|/|\mathcal{D}|$  is called as the *poisoning rate*.

**Attack Strategy.** We formulate the proposed attack as a bi-level optimization. Its details are in the next subsection.

**Characterization of Sparsity and Invisibility.** Given a  $d$ -dimensional vector  $\mathbf{x}$ ,  $L_0$ -norm is defined as  $\|\mathbf{x}\|_0 = \sum_{i=1}^d \mathbb{I}(x_i \neq 0)$  where  $\mathbb{I}(\cdot)$  is the indicator function. In general,  $L_0$ -norm reflects the maximum number of pixels allowed for modification and is widely used to measure sparsity in recent works [49, 42].  $L_{\infty}$ -norm is defined as  $\|\mathbf{x}\|_{\infty} = \max_{1 \leq i \leq d} |x_i|$ , which represents the maximum absolute value of the elements. It is widely adopted to reflect invisibility, especially in adversarial attacks [60, 61].

#### B. Sparse and Invisible Backdoor Attack (SIBA)

As we mentioned in the previous section, we need to optimize the trigger sparsity and visibility simultaneously to ensure better stealthiness. In this section, we introduce the formulation and optimization of our sparse and invisible backdoor attack.

**Problem Formulation.** The objective of SIBA could be formulated as a bi-level optimization problem since the effectiveness of the trigger pattern is related to a trained model whose optimization is also influenced by poisoned samples, as follows:

$$\begin{aligned} & \min_{\mathbf{t}} \sum_{(\mathbf{x}, y) \in \mathcal{D}_v} \mathcal{L}(f_w(\mathbf{x} + \mathbf{t}), y_T) \\ \text{s.t. } & \mathbf{w} = \arg \min_{\theta} \sum_{(\mathbf{x}, y) \in \mathcal{D}_c \cup \mathcal{D}_p} \mathcal{L}(f_{\theta}(\mathbf{x}), y), \\ & \underbrace{\|\mathbf{t}\|_0 \leq k}_{\text{sparsity}}, \underbrace{\|\mathbf{t}\|_{\infty} \leq \epsilon}_{\text{invisibility}} \end{aligned} \quad (1)$$

where  $\mathcal{D}_v$  denotes the validation set acquired by the adversary. The upper-level optimization aims to ensure the effectiveness of the trigger, that is, the trained model  $f_w$  would classify the samples attached with the triggers as the target class. The lower-level optimization represents the training process of the victim model. Besides, we add  $L_0$  and  $L_{\infty}$  constraints to confirm the trigger's sparsity and invisibility.

**Surrogate Optimization Problem.** Due to the high non-convexity of the optimization Problem 1, we need to seek a feasible solution. In particular, the optimization of  $\theta$  in the lower-level optimization requires full model training, which is time-consuming. To alleviate the computational burden and optimization difficulties, we exploit a pre-trained benign model  $f_b$  to replace  $f_w$  in the upper-level optimization. Instead of solving the Problem 1 directly, we turn it into the following surrogate optimization problem:

$$\begin{aligned} & \min_{\mathbf{t}} \sum_{(\mathbf{x}, y) \in \mathcal{D}_v} \mathcal{L}(f_b(\mathbf{x} + \mathbf{t}), y_T) \\ \text{s.t. } & \underbrace{\|\mathbf{t}\|_0 \leq k}_{\text{sparsity}}, \underbrace{\|\mathbf{t}\|_{\infty} \leq \epsilon}_{\text{invisibility}}. \end{aligned} \quad (2)$$

In this way, we only need to train the surrogate model  $f_b$  once and avoid frequent updates. We will demonstrate the feasibility and rationality of the surrogate optimization by showing the attack transferability in Section IV.

**Practical Optimization.** Let  $h(\mathbf{t}) = \sum_{(\mathbf{x}, y) \in \mathcal{D}_v} \mathcal{L}(f_b(\mathbf{x} + \mathbf{t}), y_T)$ , we investigate the update of  $h(\mathbf{t})$  under the  $L_{\infty}$  and  $L_0$  constraints sequentially. Firstly, to satisfy the  $L_{\infty}$  constraint in Problem 2, we utilize the projected gradient method, which

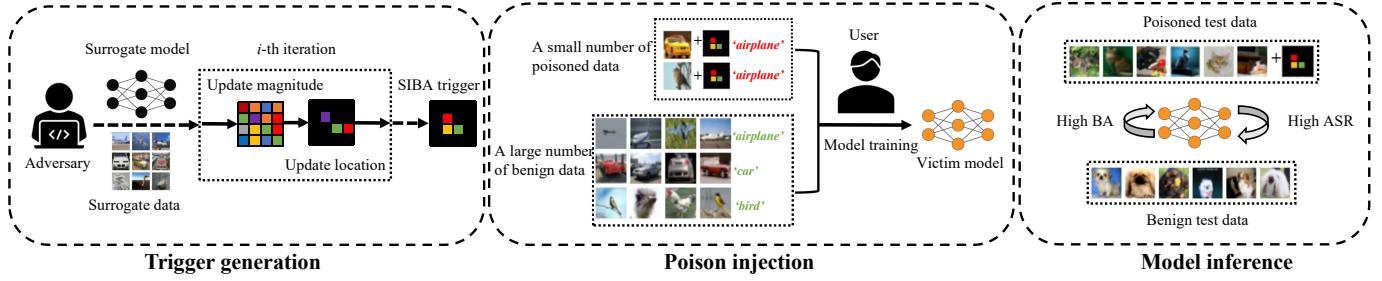


Fig. 1: The main pipeline of our sparse and invisible backdoor attack (SIBA). In the first step, the adversaries will generate a sparse and invisible trigger pattern. In the second step, the victim users will train their model on both benign and poisoned samples released by the adversary. In the third step, the attacked model can correctly predict clean test samples whereas the adversaries can maliciously change its predictions to the target class ('airplane' in this example).

### Algorithm 1 The optimization process of our SIBA.

#### Require:

Batch size  $M$ , step size  $\alpha$ , number of training iterations  $T$ , update step  $K$ , pre-trained model  $f_b$ , the number of maximum perturbed pixels  $k$ ,  $L_\infty$  constraint  $\epsilon$ , target label  $y_T$ , mask vector  $\mathbf{m}$ , validation set  $\mathcal{D}_v$ .

Initialize  $\mathbf{t} \leftarrow \mathbf{0}, j \leftarrow 0, i \leftarrow 0, \mathbf{m} \leftarrow \mathbf{0}$ .  $\mathcal{B}_\epsilon = \{\mathbf{t} \mid \|\mathbf{t}\|_\infty \leq \epsilon\}$ .

```

1: while  $j < T$  do
2:   Sample mini-batch of  $M$  samples  $\{\mathbf{x}_i, y_i\}_{i=1}^M$  from  $\mathcal{D}_v$ 
3:   if  $i \bmod K = 0$  then
4:     Select the top  $k$  large dimension  $d_1, \dots, d_k$  from  $|\nabla_{\mathbf{t}} \sum_{i=1}^M \mathcal{L}(f_b(\mathbf{x}_i + \mathbf{t}), y_T)|$ 
5:      $\mathbf{m}_{d_1, \dots, d_k} \leftarrow 1$ 
6:      $\mathbf{m}_{[d] \setminus d_1, \dots, d_k} \leftarrow 0$ 
7:   end if
8:    $\mathbf{t} \leftarrow \Pi_{\mathcal{B}_\epsilon} \left( \mathbf{t} - \alpha \cdot \epsilon \cdot \text{sign}(\nabla_{\mathbf{t}} \sum_{i=1}^M \mathcal{L}(f_b(\mathbf{x}_i + \mathbf{t}), y_T)) \right)$ 
9:    $\mathbf{t} \leftarrow \mathbf{m} \cdot \mathbf{t}$ 
10:   $j \leftarrow j + 1$ 
11:   $i \leftarrow i + 1$ 
12: end while
13: return The optimized trigger  $\mathbf{t}$ .

```

has been extensively explored in adversarial training [62]. The  $i$ -th update formula is shown as follows:

$$\mathbf{v}_i = \Pi_{\mathcal{B}_\epsilon} (\mathbf{t}_i - \alpha \cdot \epsilon \cdot \text{sign}(\nabla_{\mathbf{t}} h(\mathbf{t}_i))), \quad (3)$$

where  $\mathcal{B}_\epsilon = \{\mathbf{t} \mid \|\mathbf{t}\|_\infty \leq \epsilon\}$ ,  $\alpha$  is the step size. Next, we attempt to project  $\mathbf{v}_i$  into the  $L_0$  box:  $\mathcal{B} = \{\mathbf{t} \mid \|\mathbf{t}\|_0 \leq k, \mathbf{t}_j = \mathbf{v}_{i,j}$  or  $0$ ,  $j = 1, 2, \dots, d\}$ , which means we must select at most  $k$  element of  $\mathbf{v}_i$  and set the other elements to zero. We denote  $\mathbf{s}_i = \mathbf{t}_i - \alpha \cdot \nabla_{\mathbf{t}} h(\mathbf{t}_i)$  and require the projected  $\mathbf{t}_{i+1}$  as close to  $\mathbf{s}_i$  as possible in the terms of square loss, as follows:

$$\mathbf{t}_{i+1} = \arg \min_{\mathbf{u} \in \mathcal{B}} \|\mathbf{s}_i - \mathbf{u}\|_2^2. \quad (4)$$

To solve Problem 4, we have the following Lemma.

**Lemma 1.** Assuming  $\alpha = 0$  in Equation 3 and the initial value of  $\mathbf{t}_i$  is 0, Problem 4 has the analytical solution as follows:

$$\mathbf{t}_{i+1,j} = \begin{cases} \mathbf{v}_{i,j} & \text{if } j \in C' \\ 0 & \text{if } j \notin C' \end{cases} \quad (5)$$

where  $C'$  represents the subscript group which has the largest  $k$  element of  $|\nabla_{\mathbf{t}} h(\mathbf{t}_i)|$ .

The proof of Lemma 1 is in the appendix. Based on the above analysis, each iteration consists of Step 3 and Step 5. We exploit a mask  $\mathbf{m} \in \{0, 1\}^d$  to perform Step 5 and update  $\mathbf{m}$  after multiple iterations of Step 3 to stabilize the optimization. The detailed optimization process is shown in Algorithm 1.

**Poison Injection.** We attach the optimized trigger  $\mathbf{t}$  to a small portion of training data and relabel them as the target class. The modified training set that consists of the triggered set and the untouched set is released to the users for training the victim models. The training configurations of the victim models are determined by the users while the adversary can not intervene.

**Model Inference.** During the inference phase, the adversary validates the effectiveness of the backdoor attack with two types of data: benign test data and poisoned test data. The victim model is expected to predict correct class for the benign test data and the target class for the poisoned test data. The whole pipeline of our SIBA is shown in Figure 1.

## IV. EXPERIMENTS

### A. Experimental Setup

**Datasets.** In this paper, we consider CIFAR-10 and VGGFace2 datasets in our experiments. CIFAR-10 dataset consists of 10 classes and each class includes 5,000 training images and 1,000 test images. The size of each image is  $32 \times 32 \times 3$ . For VGGface2 dataset, we construct a 20-class subset from the original set for training efficiency. Each class includes 400 training images and 100 test images. The size of each image is  $128 \times 128 \times 3$ . Both datasets are commonly used in recent backdoor-related research [12, 21, 63, 64, 65].

**Baseline Selection.** We compare our SIBA method with six classical and representative backdoor attacks, including (1) BadNets [4], (2) backdoor attack with the blended strategy (dubbed as 'Blended') [19], (3) TUAP [17], (4) WaNet [12], (5) ISSBA [18], and (6) UBW-P [21]. Among the aforementioned

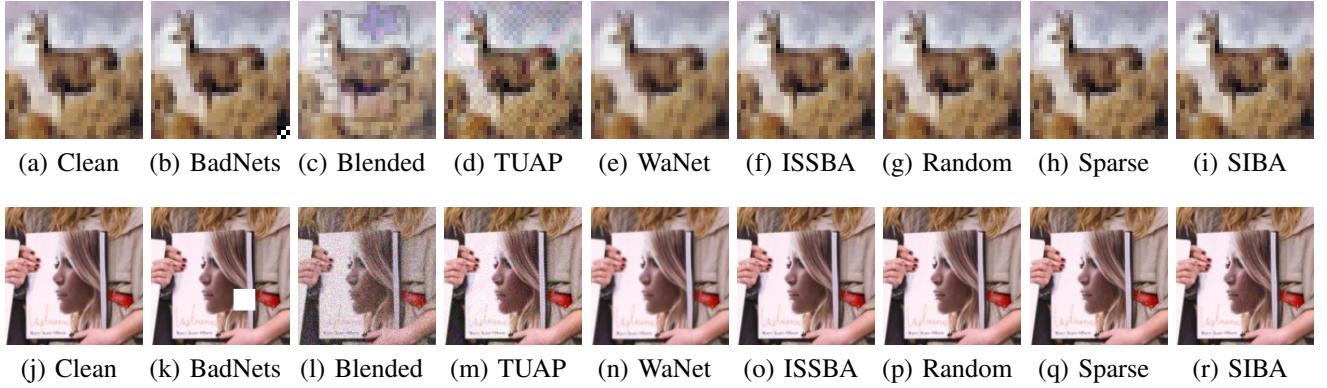


Fig. 2: The examples of poisoned samples with different backdoor attacks on CIFAR-10 and VGGFace2 datasets. **First Row:** poisoned samples on the CIFAR-10 dataset. **Second Row:** poisoned samples on the VGGFace2 dataset.

baselines, the triggers of BadNets and UBW-P are sparse (small  $L_0$  constraint) but visible (large  $L_\infty$  constraint), while those of others (Blended, TUAP, WaNet, ISSBA) are invisible (small  $L_\infty$  constraint) but dense (large  $L_0$  constraint). We also provide the results of two straightforward (yet ineffective) sparse and invisible backdoor attacks, including (1) using a random noise as the trigger that is restricted to have the same  $L_\infty$  and  $L_0$  constraint with our SIBA (dubbed as ‘Random’) and (2) the improved version of Random where its trigger magnitude is optimized using Line 8 in Algorithm 1.

**Attack Settings.** For all attacks, we set the poisoning rate as 1% and choose the class ‘0’ as the target class. Specifically, for the settings of BadNets, the trigger is a  $3 \times 3$  checkerboard on CIFAR-10 and a  $20 \times 20$  all-white patch on VGGFace2; The trigger is a Hello-Kitty image on CIFAR-10 and a random noise image on VGGFace2 for Blended. The transparency parameter is set as 0.2; We exploit a pre-trained model to generate the targeted universal adversarial perturbation on the benign model as the trigger for TUAP. The  $L_\infty$  constraint is set as 8/255 on both datasets; For WaNet, the size of the control field is  $4 \times 4$  and the strength parameter of the backward warping field is set to 0.5; We adopt the default settings used in its original paper [18] for ISSBA; UBW-P is an untargeted backdoor with the BadNets-type trigger pattern; For our SIBA, we set the step size  $\alpha = 0.2$  and  $L_\infty$  constraint  $\epsilon = 8/255$  on both datasets.  $L_0$  constraint  $k$  is set to 100 on CIFAR-10 dataset and 1,600 on VGGface2 dataset. We set the number of training iterations  $T = 200$  and the update step  $K = 5$  on both datasets. We use the whole training set to optimize the SIBA trigger. We implement them based on BackdoorBox [66]. The example of poisoned samples is shown in Figure 2.

**Training Settings.** We select ResNet-18 [67] and VGG-16 [68] as the model structures. For the CIFAR-10 dataset, the victim model is obtained by using SGD optimizer with momentum 0.9 and weight decay  $5 \times 10^{-4}$ . The number of training epochs is 100 and the initial learning rate is 0.1 which is divided by 10 in the 60-th epoch and the 90-th epoch. For the VGGFace2 dataset, we exploit an ImageNet pre-trained model and train the victim model using the SGD optimizer with momentum 0.9 and weight decay  $1 \times 10^{-4}$ . The number of training epochs

is 30 and the initial learning rate is 0.001 which is divided by 10 in the 15-th epoch and the 20-th epoch. Classical data augmentations such as random crop and random horizontal flip are used for higher benign accuracy. Note that the pre-trained model  $f_b$  and the victim model have the same network structure and we will explore the attack effectiveness when they are different in Section IV-E2.

**Evaluation Metrics.** Following the classical settings in existing research, we adopt benign accuracy (BA) and attack success rate (ASR) to evaluate all backdoor attacks. BA is the ratio of correctly classified samples in the benign test set. ASR is the ratio of test samples that are misclassified as the target class when the trigger is attached to them. BA indicates the model utility and ASR reflects the attack effectiveness. We report the comparison of the  $L_0$  and  $L_\infty$  distances of trigger patterns for indicating stealthiness. Besides, we also exploit LPIPS [69] and the structural similarity index measure (SSIM) for the reference of stealthiness. The higher the BA, ASR, and SSIM, the lower the  $L_0$ ,  $L_\infty$ , and LPIPS, the better the attack.

## B. Main Results

As shown in Table I-II, our SIBA reaches the best performance among all sparse and invisible backdoor attacks (*i.e.*, Random, Sparse, and SIBA) on both datasets. Especially on the VGGFace2 dataset, the ASR improvements are larger than 50% compared to Sparse and 90% compared to Random. These results verify the effectiveness of our trigger optimization. Besides, the ASRs of our attack are always higher than 90% and the BA decreases compared to the model trained without attacks are always less than 2%. In particular, the attack performance of our SIBA is on par with (BadNets and Blended) or even better than (TUAP, WaNet, ISSBA, and UBW-P) of all baseline attacks that are either visible or not sparse. Notably, SIBA is the only attack that achieves stealthiness across all four metrics ( $L_0$ ,  $L_\infty$ , LPIPS, and SSIM). These results verify the effectiveness and stealthiness of our proposed method.

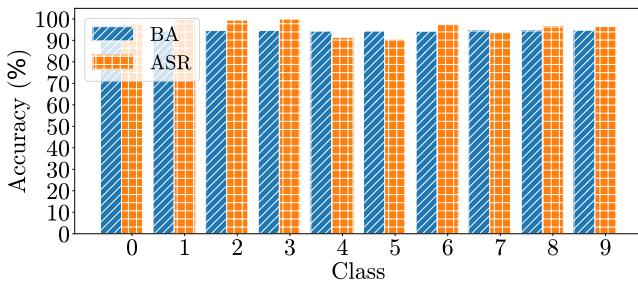
Besides, we devised a survey requiring 20 participants to choose the most likely modified image within each set containing four clean images and one SIBA-poisoned image, where each survey has five image sets. The overall selection

TABLE I: The results of backdoor attacks on CIFAR-10. We mark the background of bad cases in red whose ASR is lower than 90% or  $L_0/L_\infty$  distance is larger than 10% of the maximum possible values. ‘–’ denotes not available.

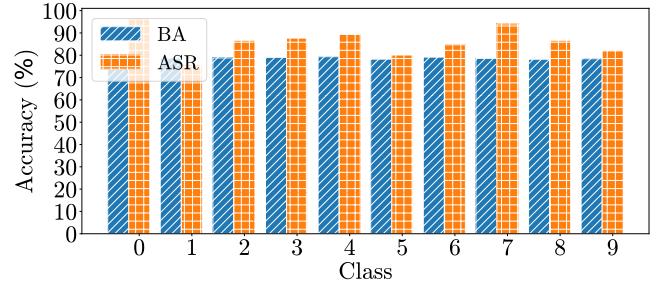
Model ↓	Metric ↓, Method →	No Attack	BadNets	Blended	TUAP	WaNet	ISSBA	UBW-P	Random	Sparse	SIBA
ResNet	BA (%)	94.67	94.41	94.58	94.53	94.29	94.57	94.46	94.13	94.44	94.06
	ASR (%)	–	100	98.16	85.47	47.90	0.76	64.78	2.87	88.49	97.60
	$L_0$	–	9	1,020	1,020	1,016	1,024	9	100	100	100
	$L_\infty$	–	0.81	0.19	0.03	0.19	0.05	0.81	0.03	0.03	0.03
	LPIPS	–	< 0.001	0.028	0.001	0.003	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
	SSIM	–	0.974	0.774	0.919	0.973	0.989	0.974	0.995	0.992	0.993
VGG	BA (%)	93.34	93.25	93.31	93.06	92.09	93.15	93.16	92.69	92.59	92.84
	ASR (%)	–	100	98.28	69.13	7.53	1.24	10.02	1.48	66.44	91.22
	$L_0$	–	9	1,020	1,020	1,016	1,024	9	100	100	100
	$L_\infty$	–	0.81	0.19	0.03	0.19	0.05	0.81	0.03	0.03	0.03
	LPIPS	–	< 0.001	0.028	0.001	0.003	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
	SSIM	–	0.974	0.774	0.919	0.973	0.989	0.974	0.995	0.992	0.993

TABLE II: The results of backdoor attacks on VGGFace2. We mark the background of bad cases in red whose ASR is lower than 90% or  $L_0/L_\infty$  distance is larger than 10% of the maximum possible values. ‘–’ denotes not available.

Model ↓	Metric ↓, Method →	No Attack	BadNets	Blended	TUAP	WaNet	ISSBA	UBW-P	Random	Sparse	SIBA
ResNet	BA (%)	79.75	78.80	78.80	78.75	79.05	78.30	77.85	77.80	78.25	78.85
	ASR (%)	–	93.68	99.95	85.84	3.32	1.00	38.20	1.74	27.79	96.21
	$L_0$	–	400	16,384	16,332	15,892	16,382	400	1,600	1,600	1,600
	$L_\infty$	–	0.92	0.20	0.03	0.23	0.04	0.92	0.03	0.03	0.03
	LPIPS	–	0.08	0.18	0.05	0.02	< 0.01	0.08	< 0.01	< 0.01	< 0.01
	SSIM	–	0.965	0.538	0.870	0.975	0.974	0.965	0.989	0.984	0.987
VGG	BA (%)	86.35	86.15	86.30	85.90	85.75	85.40	84.50	85.90	85.65	86.15
	ASR (%)	–	98.42	100	83.68	2.95	87.53	48.10	4.63	41.68	96.37
	$L_0$	–	400	16,384	16,332	15,892	16,382	400	1,600	1,600	1,600
	$L_\infty$	–	0.92	0.20	0.03	0.23	0.04	0.92	0.03	0.03	0.03
	LPIPS	–	0.08	0.18	0.05	0.02	< 0.01	0.08	< 0.01	< 0.01	< 0.01
	SSIM	–	0.965	0.538	0.870	0.975	0.974	0.965	0.989	0.984	0.987



(a) CIFAR-10



(b) VGGFace2

Fig. 3: The Effects of the target class on CIFAR-10 and VGGFace2 datasets.

accuracy is 23%, which is close to that of a random guess (20%). This result indicates that human eyes can hardly detect the SIBA trigger, verifying the stealthiness of our attack again.

### C. Ablation Study

In this section, we discuss the effectiveness of our SIBA with different key hyper-parameters. Unless otherwise specified, all settings are the same as those used in Section IV-B.

**Effects of the Target Label.** To validate the effectiveness of SIBA with different target labels, we conduct experiment on the ResNet18 with ten different classes. As shown in Figure 3, we

could find that SIBA achieves > 90% ASR for all cases on the CIFAR-10 dataset and > 75% ASR on the VGGface2 dataset, although the performance may have some mild fluctuations.

**Effects of the Poisoning Rate.** To validate the effectiveness of SIBA with different poisoning rates, we experiment on the ResNet18 model with more poisoning rates from 0.5% to 2.5%. As shown in Figure 4, the attack performance of our SIBA increases with the increase of the poisoning rate. The attack performance of our SIBA is always better than baseline invisible and sparse attacks (*i.e.*, Random and Sparse). In particular, on the CIFAR-10 dataset, SIBA achieves > 90%

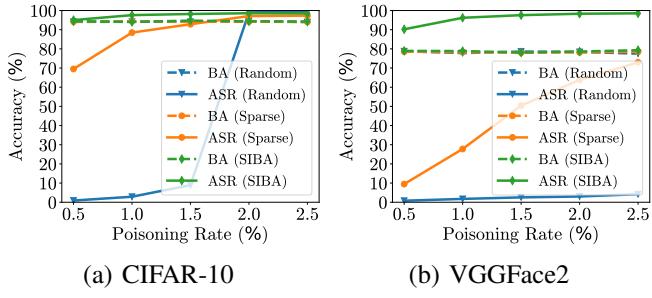


Fig. 4: Results with different poisoning rates on the CIFAR-10 dataset and the VGGFace2 dataset.

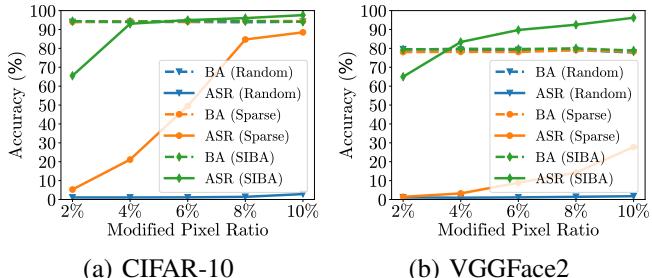


Fig. 5: The effects of  $L_0$  constraint on the CIFAR-10 dataset and the VGGFace2 dataset.

ASR with only 0.5% poisoning rate while the poisoning rate of the other two baselines has to be set three or four times higher to achieve similar attack performance. The advantage of our SIBA is even more obvious on the VGGFace2 dataset.

**Effects of the  $L_0$  Constraints.** To investigate the attack performance of our SIBA under various  $L_0$  constraints, we experiment on the ResNet18 model with different  $k$  values ranging from 50 to 250 on CIFAR-10 and from 800 to 2400 on VGGFace2, respectively. As shown in Figure 5, the attack effectiveness increases with the increase of  $k$  while having mild effects on the benign accuracy. In particular, our SIBA achieves  $> 90\%$  ASR with only 50 perturbed pixels (about 5% sparsity) on the CIFAR-10 dataset. However, for the other baseline attacks, the number of maximum perturbed pixels has to be increased to 150 for ‘Sparse’ and 200 for ‘Random’ to reach a similar performance. The improvement of our SIBA is even larger on the VGGFace2 dataset.

**Effects of the  $L_\infty$  Constraint.** To investigate the attack performance of our SIBA under various  $L_\infty$  constraints, we experiment on the ResNet18 model with different  $\epsilon$  values, ranging from 4/255 to 20/255. As shown in Figure 6, similar to the effects of  $k$ , the attack effectiveness increases with the increase of  $\epsilon$  while having mild effects on the benign accuracy. Our SIBA can achieve > 90 ASR under 4/255 budget on the CIFAR-10 dataset. In contrast, to achieve a similar attack performance, the  $L_\infty$  constraints of both baseline attacks have to be increased to three or four times larger than that of the SIBA. The results on VGGFace2 also demonstrate the superiority of SIBA over these baseline methods.

**Effects of Other Parameters.** To demonstrate the stability of our attack under other parameters, we experiment on CIFAR-10

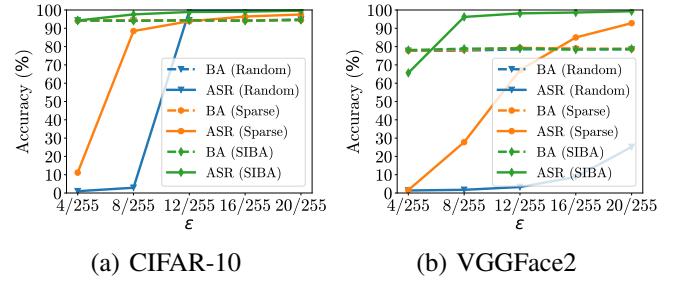


Fig. 6: The effects of  $L_\infty$  constraint on the CIFAR-10 dataset and the VGGFace2 dataset.

TABLE III: The resistance to anti-backdoor learning (ABL).

Dataset↓	Model↓, Metric→	BA (%)	ASR (%)
CIFAR-10	ResNet	88.33	22.71
	VGG	82.41	94.00
VGGFace2	ResNet	72.45	74.74
	VGG	77.20	96.53

dataset with ResNet18 model and various  $K$ ,  $\alpha$ , and  $T$  values. Specifically,  $K$  ranges from 5 to 20;  $\alpha$  ranges from 0.2 to 1.0;  $T$  ranges from 200 to 1,000. As shown in Figure 7, the ASR of SIBA is always higher than 95% in all cases. These results indicate that we can easily obtain a good performance without fine-tuning these parameters in practice.

#### D. The Resistance to Potential Backdoor Defenses

**The Resistance to STRIP.** As a representative black-box detection of poisoned training samples with predicted logits, STRIP [38] perturbs a given test image by superimposing various images and then inspects the entropy of the model prediction. The suspicious samples having low entropy are regarded as poisoned samples. We evaluate the resistance of our SIBA to STRIP by visualizing the entropy distributions of samples. As shown in Figure 8, the entropy distributions of poisoned samples are mixed with those of benign samples. Accordingly, our SIBA can evade the detection of STRIP.

**The Resistance to Anti-backdoor Learning (ABL).** As a representative poison suppression method, ABL [31] first identifies the poisoned sample candidates with loss values and then unlearns the candidate samples by gradient ascent. In our experiments, the isolation epoch and the unlearning epoch are set to 20 and 80, as suggested in its original paper [31]. As shown in Table III, our attack is resistant to ABL in most cases, although the ASR may have some decreases. Its failure is mostly because the loss values are not effective to reflect the difference between poisoned and benign samples of SIBA.

**The Resistance to Fine-pruning (FP).** As a representative backdoor removal method, fine-pruning (FP) [34] first tests the candidate model with a small clean validation set and records the average activation of each neuron. Then, FP prunes the channels with increasing order until the clean accuracy drops below some threshold. In our experiments, the validation set is obtained by randomly choosing 20% samples from the clean training dataset and the total channel number is 512. The curves

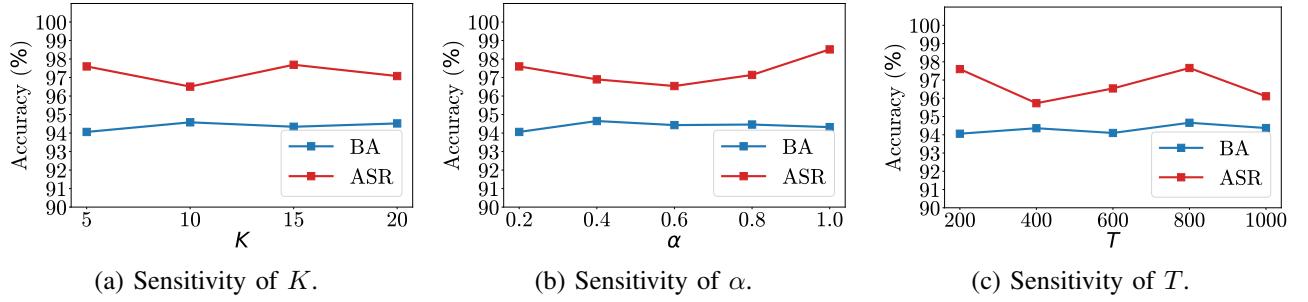


Fig. 7: Results of our SIBA with different parameters on the CIFAR-10 dataset and the VGGFace2 dataset.

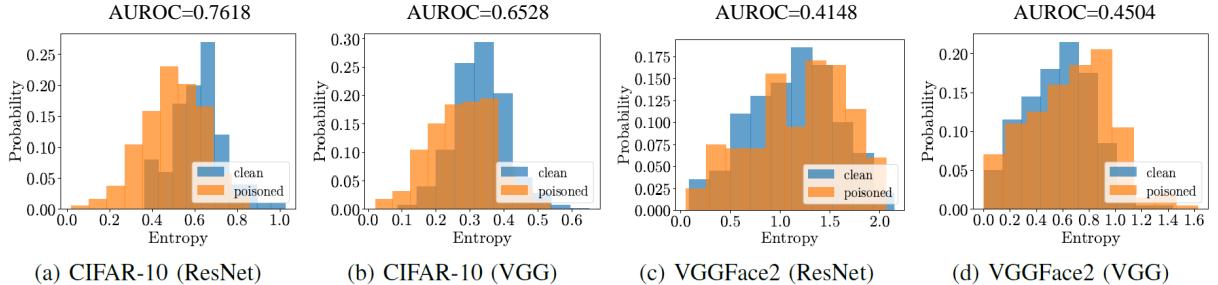


Fig. 8: The resistance of our SIBA to STRIP.

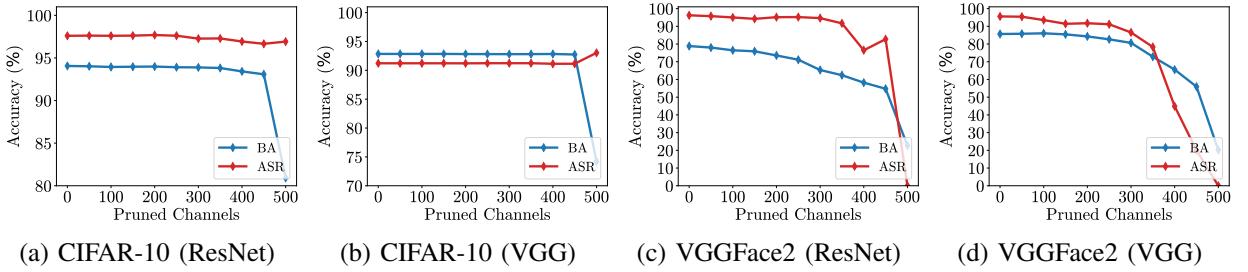


Fig. 9: The resistance of our SIBA to fine-pruning (FP).

TABLE IV: The resistance of our SIBA to Scale-Up.

Dataset↓	Model↓, Metric→	TPR	FPR	AUROC	ASR (%)
CIFAR-10	ResNet	0.5816	0.2067	0.7427	41.84
	VGG	0.4028	0.2135	0.6369	59.72
VGGFace2	ResNet	0.0220	0.0855	0.4428	97.80
	VGG	0.0295	0.0830	0.4603	97.05

of BA and ASR with respect to the number of pruned channels are shown in Figure 9. We could observe that the ASR of the proposed backdoor attack preserves on CIFAR-10 even if a large portion of channels are pruned. As for the VGGFace2 dataset, the ASR is reduced below 80% when the number of pruned channels is larger than 400. However, the BA is significantly decreased as its sacrifice. These results verify the resistance of our SIBA to FP.

**The Resistance to Scale-Up.** As a representative black-box detection of poisoned testing samples with predicted labels, Scale-Up [39] discovered the phenomenon that the poisoned samples had the scaled prediction consistency when the pixel values were amplified and proposed to distinguish the poisoned samples by counting the predictions of scaled images. In our experiments, we use a scaling set with size 5 and set

the threshold as 0.8. We report true positive rate (TPR), false positive rate (FPR), area under the receiver operating characteristic (AUROC), and ASR in Table IV. As shown in the table, although Scale-Up can decrease the effectiveness of SIBA to some extent, the detection performance is far from satisfactory since the average ASR is still  $> 70\%$ . In other words, our SIBA is resistant to the Scale-Up to a large extent.

**The Resistance to SentiNet.** As a representative white-box detection of poisoned testing samples, SentiNet [37] relies on model interpretability techniques to locate potential trigger regions. Grad-CAM uses the gradient with respect to the model’s final layer and calculates the salience map of the input region to reflect the positive importance of the input image. In our experiments, we visualize the salience maps of some poisoned samples on CIFAR-10 and VGGFace2 datasets. As shown in Figure 10, the salience maps could not provide useful information to detect the trigger. Its failure is mostly because the trigger of SIBA is not a small-sized patch.

**The Resistance to Neural Cleanse (NC).** As a representative model-level detection method, NC [29] first reverses possible triggers of the suspicious model and collects the  $L_1$  values of

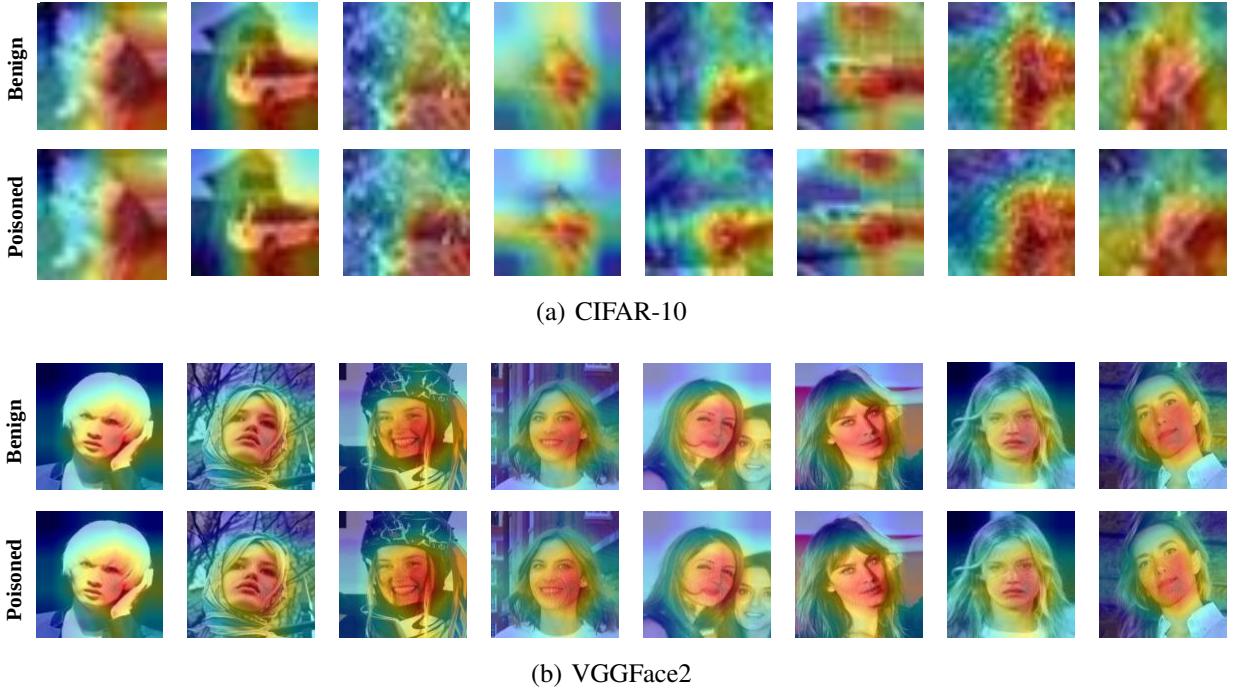


Fig. 10: The resistance of our SIBA to SentiNet.

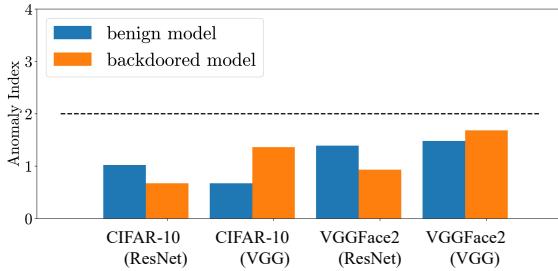


Fig. 11: The resistance of our SIBA to neural cleanse (NC).

the optimized trigger associated with each target label. Then, NC calculates the median absolute deviation of the group and the anomaly index of each label. If the anomaly index is larger than the threshold, the model is regarded as backdoored. In our experiments, the threshold is 2 as suggested in its original paper. We use the Adam optimizer in which the learning rate is 0.1. The coefficient of the regularization term is 0.001 and the number of training epochs is 50. As shown in Figure 11, the NC is ineffective to detect our backdoored model since the anomaly index of the proposed method is always lower than the threshold value. This failure is mostly because our SIBA only needs to manipulate a small number of pixels such that the optimization of NC cannot catch our trigger location.

**The Resistance to Meta Neural Trojan Detection (MNTD).** The MNTD [40] is another famous model-level detection method. It trains a meta-classifier to determine whether a target model is backdoored. In our experiments, we set the query number 10 and use Adam optimizer [70] with a 0.001 learning rate to train the meta classifier. Besides, we train 20 candidate models (10 SIBA-backdoored models and 10 clean models) and then calculate the trojan scores based on its meta classifier. The result (AUROC=0.57) indicates that SIBA can escape from

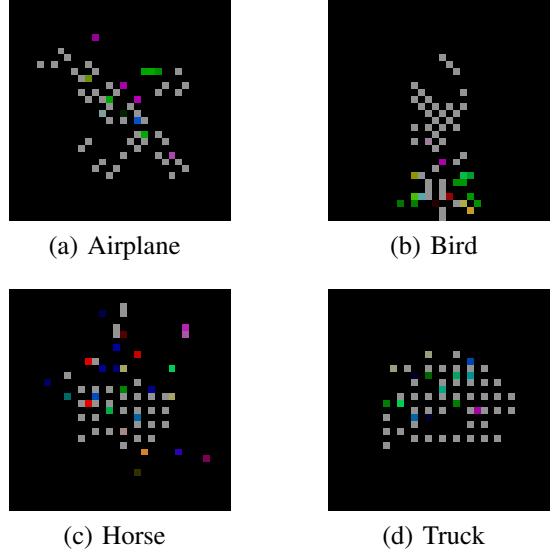


Fig. 12: The normalized SIBA triggers generated with different target classes on the CIFAR-10 dataset.

MNTD. It is mostly because MNTD cannot capture sufficient trigger-related features due to our trigger sparseness. We will study it further in our future works.

### E. Discussion

1) *A Closer Look to the Effectiveness of our SIBA and its Connection to DNN Interpretability:* To understand the effectiveness of our SIBA, we also visualize the (normalized) trigger patterns when different target labels are adopted. As shown in Figure 12, the SIBA trigger is always located in the

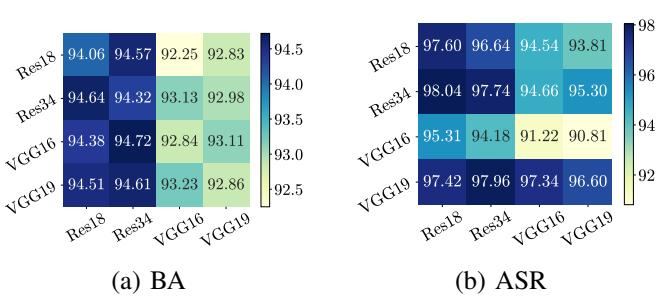


Fig. 13: The performance of our SIBA with different surrogate and victim structures on the CIFAR-10 dataset. **Row:** surrogate models; **Column:** victim models.

main body of the object from the target class (such as the airplane’s fuselage and wings in Figure 12). In other words, the generated sparse trigger pattern is closely related to the concept of the target label. This phenomenon also (partially) explains the prediction and learning mechanisms of DNNs. Specifically, from the prediction perspective, it indicates that the misclassification of DNNs towards poisoned samples from their ground-truth label to the target one might be attributed to perturbations in regions associated with the target class. From the learning perspective, it implies that DNNs learn the trigger pattern as the representative of samples from the target class since we only re-assign the label of selected samples as the target label when generating the trigger pattern. It suggests that our attack may be a viable path toward understanding the learning and prediction principles of DNNs. Our results build an interesting connection between backdoor attacks and DNN interpretability that has not been shown before.

*2) Attack Transferability with Different Model Structures:* As described in Section III-B, we need a pre-trained benign model to generate our SIBA trigger pattern. The experiments in Section IV-A are conducted based on the assumption that the surrogate model and victim model have the same model structure, which may not be feasible in practice since the adversaries have no information of the structure that victim users may use. In this part, we explore the transferability of our SIBA: ‘How effective is SIBA when the surrogate model is different from the victim model?’. We select four network architectures: ResNet18, ResNet34, VGG16, and VGG19 on CIFAR-10 for discussions. Other settings are the same as those used in Section IV-B. As shown in Figure 13, our SIBA achieves consistently excellent attack performance under different settings, although the performance may have some fluctuations due to different model capacities. These results indicate that our SIBA method does not require knowing any information of victim users and therefore can serve as an effective poison-only backdoor attack.

*3) The Extension to All-to-all Setting:* The experiments in Section IV-B adopt the all-to-one setting, where all poisoned samples are expected to be classified as the same target class. In this part, we extend our SIBA to the all-to-all setting, in which the target class depends on the ground truth class of the poisoned sample. Specifically, we adopt the most classical

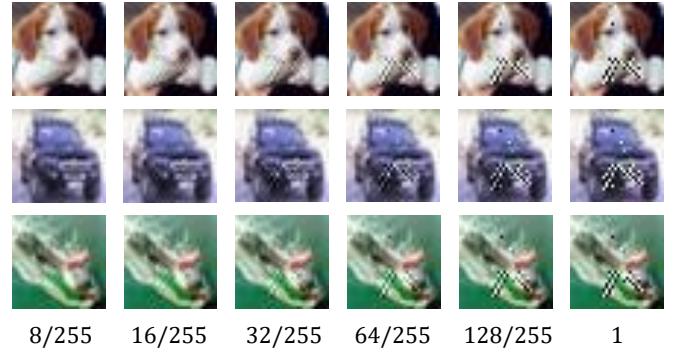


Fig. 14: The example of poisoned samples with amplified triggers on the CIFAR-10 dataset.

TABLE V: Results with limited data on the CIFAR-10 dataset.

Metric → Data Percentage↓	BA (Surrogate)	BA (Victim)	ASR (Victim)
5%	57.55%	94.44%	65.44%
10%	70.65%	94.25%	97.10%
15%	79.32%	94.11%	95.49%
20%	84.14%	94.83%	97.56%
100%	94.67%	94.06%	97.60%

transformation function ‘ $c(y) = (y + 1) \bmod C$ ’ in this paper, following the settings of existing papers [4, 71]. In this case, the problem formulation of our SIBA is as follows:

$$\min_t \sum_{(\mathbf{x}, y) \in \mathcal{D}_v} \mathcal{L}(f_b(\mathbf{x} + \mathbf{t}), c(y)) \quad (6)$$

s.t.  $\|\mathbf{t}\|_0 \leq k$ ,  $\|\mathbf{t}\|_\infty \leq \epsilon$ .

We conduct experiments of the all-to-all SIBA attack on the CIFAR-10 dataset with ResNet18. The poisoning rate is increased to 10% since the all-to-all attacks are more complicated than all-to-one methods. All other settings are the same as those used in Section IV-B. As a result, the BA is 94.61% and the ASR is 93.34%, indicating that our SIBA is feasible to be applied under the all-to-all setting.

*4) SIBA with Limited Training Data:* In the previous sections, we assume that the adversary optimizes the SIBA trigger via the whole training set. However, in real scenarios, it might be infeasible to acquire the whole training set for the adversary to train the surrogate model. We hereby raise the question: ‘How effective is SIBA when the adversary has limited data?’. In this part, we optimize our SIBA trigger based on a subset of the training set in which the data percentage ranges from 5% to 20%. We report the BA of the surrogate model, and the BA and the ASR of the victim model. As shown in Table V, the degraded performance of the surrogate model does not mean the inefficiency of SIBA when only limited training data is adopted. Our SIBA achieves > 90% ASR even when the adversary can only access to 10% training data. These results verify the efficiency of our SIBA.

*5) SIBA with Asymmetric Triggers:* To further boost the attack effectiveness while maintaining attack stealthiness in practical scenarios, we explore the idea of asymmetric triggers [19, 15] that maintains the original trigger during the training process but amplifies it during the inference time. Specifically,

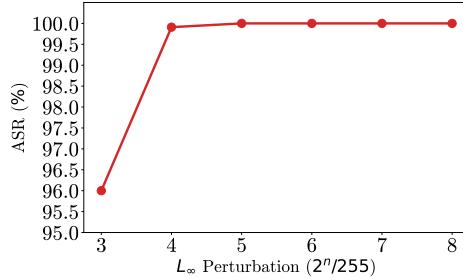


Fig. 15: Results of SIBA with amplified triggers on CIFAR-10.

TABLE VI: Comparison with the baseline attacks under clean-label settings on the CIFAR-10 dataset.

Method↓	Metric↓, Poisoning Rate→	1%	2%	3%	4%	5%
Random	BA (%)	94.51	94.42	94.56	94.10	93.87
	ASR (%)	1.51	4.91	45.58	68.89	79.81
Sparse	BA (%)	94.48	94.80	94.26	94.34	94.06
	ASR (%)	24.76	46.49	59.87	61.22	73.47
SIBA	BA (%)	94.44	94.22	94.45	94.40	94.17
	ASR (%)	62.35	75.57	76.89	86.30	91.73

we construct the test poisoned sample with the following formula:  $\mathbf{x}_i + \epsilon \cdot \text{sign}(\mathbf{t}_i)$ ,  $i = 1, 2, \dots, d$ , where  $\epsilon$  controls the visibility. In our experiments, we set the number of maximum perturbed pixels as 50. Other training details are consistent with those in Section IV-A. We illustrate the poisoned samples in Figure 14 and depict the ASR curves in Figure 15, from which we find that amplified triggers not only outperform the original triggers but also could be implemented as backdoor patches [72, 73] in physical world.

6) *Effectiveness of SIBA under Clean-label Settings:* To evaluate the effectiveness of SIBA under clean-label settings, we conduct experiments on the CIFAR-10 dataset. It is known that clean-label attacks are more challenging than poisoned-label ones and we thus set  $k = 200$ ,  $\epsilon = 16/255$ . The results are summarized in Table VI, from which we could find that our SIBA achieves  $> 90\%$  ASR with 5% poisoning rate and outperforms the other baselines with a notable margin. However, we have to acknowledge that when the sparsity is further reduced (*e.g.*,  $k = 100$ ), clean-label settings may not achieve high ASRs like poisoned-label ones. We will explore more advanced optimization techniques to push the limit of sparsity under clean-label settings in our future work.

## V. CONCLUSION

In this paper, we proposed a novel backdoor attack, *i.e.*, sparse and invisible backdoor attack (SIBA), to achieve attack effectiveness and attack stealthiness simultaneously. Our SIBA method only needs to modify a few pixels of the original images to generate poisoned samples and is human-imperceptible due to the low modification magnitude. To achieve it, we formulated the trigger generation as a bi-level optimization problem with sparsity and invisibility constraints and proposed an effective method to solve it. We conducted extensive experiments on benchmark datasets, verifying the effectiveness, the resistance to potential defenses, and the flexibility under different settings

of our attack. We hope our method can provide a new angle and deeper understanding of backdoor mechanisms, to facilitate the design of more secure and robust DNNs.

## ACKNOWLEDGMENTS

The work is supported in part by the National Natural Science Foundation of China (U20A20178, 62171248, U20B2049, and U21B2018), Shenzhen Science and Technology Program (JCYJ20220818101012025), and the PCNL KEY project (PCL2023AS6-1). This work was mostly done when Yiming Li was a Research Professor at Zhejiang University. He is currently a Research Fellow at Nanyang Technological University.

## APPENDIX

### A. Proof of Lemma 1

**Lemma 1.** Assuming  $\alpha = 0$  in Equation 3 and the initial value of  $\mathbf{t}_i$  is 0, Problem 4 has the analytical solution as follows:

$$\mathbf{t}_{i+1,j} = \begin{cases} \mathbf{v}_{i,j} & \text{if } j \in C' \\ 0 & \text{if } j \notin C' \end{cases} \quad (7)$$

where  $C'$  represents the subscript group which has the largest  $k$  element of  $|\nabla_{\mathbf{t}} h(\mathbf{t}_i)|$ .

*Proof.* We denote  $C$  as the  $k$ -dimension subset of  $\mathbf{v}_i$  such that  $\mathbf{u}_j = \mathbf{v}_{i,j}$  if  $j \in C$  and  $\mathbf{u}_j = 0$  if  $j \notin C$  and assume that the initial value of  $\mathbf{t}_i$  is 0. Then, the objective in the equation (4) could be derived as follows:

$$\begin{aligned} \|\mathbf{s}_i - \mathbf{u}\|_2^2 &= \sum_{j \in C} (\mathbf{s}_{i,j} - \mathbf{v}_{i,j})^2 + \sum_{j \notin C} \mathbf{s}_{i,j}^2 \\ &= \sum_{j \in C} (\alpha \cdot (\nabla_{\mathbf{t}} h(\mathbf{t}_i))_j - \epsilon \cdot \text{sign}(\nabla_{\mathbf{t}} h(\mathbf{t}_i))_j)^2 + \\ &\quad \sum_{j \notin C} (\alpha \cdot \nabla_{\mathbf{t}} h(\mathbf{t}_i))_j^2 \\ &= \sum_{j \in C} ((\alpha \cdot |(\nabla_{\mathbf{t}} h(\mathbf{t}_i))_j| - \epsilon) \cdot \text{sign}(\nabla_{\mathbf{t}} h(\mathbf{t}_i))_j)^2 + \\ &\quad \sum_{j \notin C} (\alpha \cdot \nabla_{\mathbf{t}} h(\mathbf{t}_i))_j^2 \\ &= \sum_{j \in C} (\alpha \cdot |(\nabla_{\mathbf{t}} h(\mathbf{t}_i))_j| - \epsilon)^2 + \\ &\quad \sum_{j \notin C} (\alpha \cdot \nabla_{\mathbf{t}} h(\mathbf{t}_i))_j^2 \\ &= \sum_{j \in C} ((\alpha \cdot \nabla_{\mathbf{t}} h(\mathbf{t}_i))_j)^2 - \sum_{j \in C} 2\alpha\epsilon \cdot |\nabla_{\mathbf{t}} h(\mathbf{t}_i)_j| + \\ &\quad \sum_{j \in C} \epsilon^2 + \sum_{j \notin C} ((\alpha \cdot \nabla_{\mathbf{t}} h(\mathbf{t}_i))_j)^2 \\ &= \|\alpha \cdot \nabla_{\mathbf{t}} h(\mathbf{t}_i)\|_2^2 + k\epsilon^2 - 2\alpha\epsilon \sum_{j \in C} |\nabla_{\mathbf{t}} h(\mathbf{t}_i)_j| \end{aligned} \quad (8)$$

Observing the equation (8), the first two terms are constants and the equation is minimized when the last term contains the largest  $k$  element of  $|\nabla_{\mathbf{t}} h(\mathbf{t}_i)|$ .  $\square$

## REFERENCES

- [1] W. Liu, Z. Li, and X. Tang, “Spatio-temporal embedding for statistical face recognition from video,” in *ECCV*, 2006.
- [2] Z. Li, D. Gong, Q. Li, D. Tao, and X. Li, “Mutual component analysis for heterogeneous face recognition,” *ACM Transactions on Intelligent Systems and Technology*, 2016.
- [3] M. Luo, H. Wu, H. Huang, W. He, and R. He, “Memory-modulated transformer network for heterogeneous face recognition,” *IEEE Transactions on Information Forensics and Security*, 2022.
- [4] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, “Badnets: Evaluating backdooring attacks on deep neural networks,” *IEEE Access*, 2019.
- [5] Y. Wang, E. Sarkar, W. Li, M. Maniatakos, and S. E. Jabari, “Stop-and-go: Exploring backdoor attacks on deep reinforcement learning-based traffic congestion control systems,” *IEEE Transactions on Information Forensics and Security*, 2021.
- [6] X. Gong, Y. Chen, Q. Wang, H. Huang, L. Meng, C. Shen, and Q. Zhang, “Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment,” *IEEE Journal on Selected Areas in Communications*, 2021.
- [7] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, “Backdoor learning: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [8] J. Dong, Q. Han, Y. Li, T. Zhang, Y. Li, Z. Lai, C. Zhang, and S.-T. Xia, “One-bit flip is all you need: When bit-flip attack meets model training,” in *ICCV*, 2023.
- [9] X. Gong, Z. Wang, Y. Chen, M. Xue, Q. Wang, and C. Shen, “Kaleidoscope: Physical backdoor attacks against deep neural networks with rgb filters,” *IEEE Transactions on Dependable and Secure Computing*, 2023.
- [10] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia, “Adversarial machine learning-industry perspectives,” in *IEEE S&P Workshop*, 2020.
- [11] [Online]. Available: <https://news.uchicago.edu/story/computer-scientists-design-way-close-backdoors-ai-based-security-systems>
- [12] A. Nguyen and A. Tran, “Wanet-imperceptible warping-based backdoor attack,” in *ICLR*, 2021.
- [13] S. Cheng, Y. Liu, S. Ma, and X. Zhang, “Deep feature space trojan attack of neural networks by controlled detoxification,” in *AAAI*, 2021.
- [14] Y. Li, H. Zhong, X. Ma, Y. Jiang, and S.-T. Xia, “Few-shot backdoor attacks on visual object tracking,” in *ICLR*, 2022.
- [15] X. Qi, T. Xie, Y. Li, S. Mahloujifar, and P. Mittal, “Revisiting the assumption of latent separability for backdoor defenses,” in *ICLR*, 2023.
- [16] M. Barni, K. Kallas, and B. Tondi, “A new backdoor attack in cnns by training set corruption without label poisoning,” in *ICIP*, 2019.
- [17] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, “Clean-label backdoor attacks on video recognition models,” in *CVPR*, 2020.
- [18] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, “Invisible backdoor attack with sample-specific triggers,” in *ICCV*, 2021.
- [19] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning,” *arXiv preprint arXiv:1712.05526*, 2017.
- [20] Y. Li, T. Zhai, Y. Jiang, Z. Li, and S.-T. Xia, “Backdoor attack in the physical world,” in *ICLR Workshop*, 2021.
- [21] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, and B. Li, “Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection,” in *NeurIPS*, 2022.
- [22] C. Luo, Y. Li, Y. Jiang, and S.-T. Xia, “Untargeted backdoor attack against object detection,” in *ICASSP*, 2023.
- [23] W. Chen, D. Song, and B. Li, “Trojdiff: Trojan attacks on diffusion models with diverse targets,” in *CVPR*, 2023.
- [24] Y. Li, M. Zhu, X. Yang, Y. Jiang, T. Wei, and S.-T. Xia, “Black-box dataset ownership verification via backdoor watermarking,” *IEEE Transactions on Information Forensics and Security*, 2023.
- [25] J. Guo, Y. Li, L. Wang, S.-T. Xia, H. Huang, C. Liu, and B. Li, “Domain watermark: Effective and harmless dataset copyright protection is closed at hand,” in *NeurIPS*, 2023.
- [26] Y.-S. Lin, W.-C. Lee, and Z. B. Celik, “What do you see? evaluation of explainable artificial intelligence (xai) interpretability through neural backdoors,” in *KDD*, 2021.
- [27] M. Ya, Y. Li, T. Dai, B. Wang, Y. Jiang, and S.-T. Xia, “Towards faithful xai evaluation via generalization-limited backdoor watermark,” in *ICLR*, 2024.
- [28] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, “Detecting backdoor attacks on deep neural networks by activation clustering,” *arXiv preprint arXiv:1811.03728*, 2018.
- [29] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, “Neural cleanse: Identifying and mitigating backdoor attacks in neural networks,” in *IEEE S&P*, 2019.
- [30] J. Hayase, W. Kong, R. Somani, and S. Oh, “Spectre: Defending against backdoor attacks using robust statistics,” in *ICML*, 2021.
- [31] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, “Anti-backdoor learning: Training clean models on poisoned data,” in *NeurIPS*, 2021.
- [32] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, “Backdoor defense via decoupling the training process,” in *ICLR*, 2022.
- [33] R. Tang, J. Yuan, Y. Li, Z. Liu, R. Chen, and X. Hu, “Setting the trap: Capturing and defeating backdoor threats in plms through honeypots,” in *NeurIPS*, 2023.
- [34] K. Liu, B. Dolan-Gavitt, and S. Garg, “Fine-pruning: Defending against backdooring attacks on deep neural networks,” in *RAID*, 2018.
- [35] Y. Zeng, S. Chen, W. Park, Z. M. Mao, M. Jin, and R. Jia, “Adversarial unlearning of backdoors via implicit hypergradient,” in *ICLR*, 2022.
- [36] B. Li, Y. Cai, H. Li, F. Xue, Z. Li, and Y. Li, “Near-

- est is not dearest: Towards practical defense against quantization-conditioned backdoor attacks,” in *CVPR*, 2024.
- [37] E. Chou, F. Tramer, and G. Pellegrino, “Sentinet: Detecting localized universal attacks against deep learning systems,” in *IEEE S&P Workshop*, 2020.
- [38] Y. Gao, Y. Kim, B. G. Doan, Z. Zhang, G. Zhang, S. Nepal, D. C. Ranasinghe, and H. Kim, “Design and evaluation of a multi-domain trojan detection method on deep neural networks,” *IEEE Transactions on Dependable and Secure Computing*, 2021.
- [39] J. Guo, Y. Li, X. Chen, H. Guo, L. Sun, and C. Liu, “Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency,” in *ICLR*, 2023.
- [40] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, “Detecting ai trojans using meta neural analysis,” in *IEEE S&P*, 2021.
- [41] X. Xu, K. Huang, Y. Li, Z. Qin, and K. Ren, “Towards reliable and efficient backdoor trigger inversion via decoupling benign features,” in *ICLR*, 2024.
- [42] Y. Fan, B. Wu, T. Li, Y. Zhang, M. Li, Z. Li, and Y. Yang, “Sparse adversarial attack via perturbation factorization,” in *ECCV*, 2020.
- [43] M. Zhu, T. Chen, and Z. Wang, “Sparse and imperceptible adversarial attack via a homotopy algorithm,” in *ICML*, 2021.
- [44] X. Zhou, Y. Lin, W. Zhang, and T. Zhang, “Sparse invariant risk minimization,” in *ICML*, 2022.
- [45] B. Li, Y. Shen, J. Yang, Y. Wang, J. Ren, T. Che, J. Zhang, and Z. Liu, “Sparse mixture-of-experts are domain generalizable learners,” in *ICLR*, 2023.
- [46] C. Févotte and S. J. Godsill, “Sparse linear regression in unions of bases via bayesian variable selection,” *SPL*, 2006.
- [47] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, 2006.
- [48] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, “Data compression and harmonic analysis,” *IEEE Transactions on Information Theory*, 1998.
- [49] T. Li, B. Wu, Y. Yang, Y. Fan, Y. Zhang, and W. Liu, “Compressing convolutional neural networks via factorized convolutional filters,” in *CVPR*, 2019.
- [50] H. Yang, S. Gui, Y. Zhu, and J. Liu, “Automatic neural network compression by sparsity-quantization joint learning: A constrained optimization-based approach,” in *CVPR*, 2020.
- [51] F. Croce and M. Hein, “Sparse and imperceptible adversarial attacks,” in *ICCV*, 2019.
- [52] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, 2005.
- [53] Z. Xu, X. Chang, F. Xu, and H. Zhang, “ $l_{\{1/2\}}$  regularization: A thresholding representation theory and a fast solver,” *IEEE Transactions on Neural Networks and Learning Systems*, 2012.
- [54] T. Zhang, “Analysis of multi-stage convex relaxation for sparse regularization.” *Journal of Machine Learning Research*, 2010.
- [55] A. Beck and Y. C. Eldar, “Sparsity constrained nonlinear optimization: Optimality conditions and algorithms,” *SIAM Journal on Optimization*, 2013.
- [56] P. Jain, A. Tewari, and P. Kar, “On iterative hard thresholding methods for high-dimensional m-estimation,” in *NeurIPS*, 2014.
- [57] Z. Lu, “Iterative hard thresholding methods for 1 0 regularized convex cone programming,” *Mathematical Programming*, 2014.
- [58] X. Dong, D. Chen, J. Bao, C. Qin, L. Yuan, W. Zhang, N. Yu, and D. Chen, “Greedyfool: Distortion-aware sparse adversarial attack,” in *NeurIPS*, 2020.
- [59] B. Biggio and F. Roli, “Wild patterns: Ten years after the rise of adversarial machine learning,” *Pattern Recognition*, vol. 84, pp. 317–331, 2018.
- [60] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, “Adversarial examples for semantic segmentation and object detection,” in *ICCV*, 2017.
- [61] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu, “Efficient decision-based black-box adversarial attacks on face recognition,” in *CVPR*, 2019.
- [62] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in *ICLR*, 2018.
- [63] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, “Februus: Input purification defense against trojan attacks on deep neural network systems,” in *ACSAC*, 2020.
- [64] J. Dumford and W. Scheirer, “Backdooring convolutional neural networks via targeted weight perturbations,” in *IJCB*, 2020.
- [65] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, “Backdoor attacks against deep learning systems in the physical world,” in *CVPR*, 2021.
- [66] Y. Li, M. Ya, Y. Bai, Y. Jiang, and S.-T. Xia, “Backdoorbox: A python toolbox for backdoor learning,” in *ICLR Workshop*, 2023.
- [67] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.
- [68] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [69] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.
- [70] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [71] K. D. Doan, Y. Lao, and P. Li, “Marksman backdoor: Backdoor attacks with arbitrary target class,” *NeurIPS*, 2022.
- [72] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, “Adversarial patch,” *arXiv preprint arXiv:1712.09665*, 2017.
- [73] A. Liu, J. Wang, X. Liu, B. Cao, C. Zhang, and H. Yu, “Bias-based universal adversarial patch attack for automatic check-out,” in *ECCV*, 2020.



**Dr. Yinghua Gao** received his Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University in 2024 and B.S. degree from the Department of Mathematics, Nankai University in 2018. His research interests are primarily in trustworthy machine learning.



**Dr. Shu-Tao Xia** received the B.S. degree in mathematics and the Ph.D. degree in applied mathematics from Nankai University, Tianjin, China, in 1992 and 1997, respectively. Since January 2004, he has been with the Tsinghua Shenzhen International Graduate School of Tsinghua University, Guangdong, China, where he is currently a full professor. From September 1997 to March 1998 and from August to September 1998, he visited the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include coding and information theory, machine learning, and deep learning. His papers have been published in multiple top-tier journals and conferences, such as IEEE TPAMI, IEEE TIFS, IEEE TDSC, CVPR, ICLR, NeurIPS.



**Dr. Yiming Li** is currently a Research Fellow at Nanyang Technological University. Before that, he was a Research Professor in the State Key Laboratory of Blockchain and Data Security at Zhejiang University and also in HIC-ZJU. He received his Ph.D. degree with honors in Computer Science and Technology from Tsinghua University in 2023 and his B.S. degree with honors in Mathematics from Ningbo University in 2018. His research interests are in the domain of Trustworthy ML and Responsible AI, especially backdoor learning and AI copyright

protection. His research has been published in multiple top-tier conferences and journals, such as ICLR, NeurIPS, and IEEE TIFS. He served as the Area Chair of ACM MM, the Senior Program Committee Member of AAAI, and the Reviewer of IEEE TPAMI, IEEE TIFS, IEEE TDSC, etc. His research has been featured by major media outlets, such as IEEE Spectrum. He was the recipient of the Best Paper Award at PAKDD in 2023 and the Rising Star Award at WAIC in 2023.



**Dr. Xueluan Gong** received her B.S. degree in Computer Science and Electronic Engineering from Hunan University in 2018. She received her Ph.D. degree in Computer Science from Wuhan University in 2023. Her research interests include AI security and information security.



**Dr. Qian Wang** (Fellow, IEEE) is a Professor in the School of Cyber Science and Engineering at Wuhan University, China. He was selected into the National High-level Young Talents Program of China, and listed among the World's Top 2% Scientists by Stanford University. He also received the National Science Fund for Excellent Young Scholars of China in 2018. He has long been engaged in the research of cyberspace security, with focus on AI security, data outsourcing security and privacy, wireless systems security, and applied cryptography. He was a recipient of the 2018 IEEE TCSC Award for Excellence in Scalable Computing (early career researcher) and the 2016 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He has published 200+ papers, with 120+ publications in top-tier international conferences, including USENIX NSDI, ACM CCS, USENIX Security, NDSS, ACM MobiCom, ICML, etc., with 20000+ Google Scholar citations. He is also a co-recipient of 8 Best Paper and Best Student Paper Awards from prestigious conferences, including ICDCS, IEEE ICNP, etc. In 2021, his PhD student was selected under Huawei's 'Top Minds' Recruitment Program. He serves as Associate Editors for IEEE Transactions on Dependable and Secure Computing (TDSC) and IEEE Transactions on Information Forensics and Security (TIFS).



**Dr. Zhifeng Li** is currently a top-tier principal research scientist at Tencent Data Platform. He received the Ph.D. degree from the Chinese University of Hong Kong in 2006. After that, He was a postdoctoral fellow at the Chinese University of Hong Kong and Michigan State University for several years. Before joining Tencent, he was a full professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include deep learning, computer vision and pattern recognition, and face detection and recognition. He is currently serving on the Editorial Boards of Pattern Recognition, IEEE Transactions on Circuits and Systems for Video Technology, and Neurocomputing. He is a fellow of the British Computer Society (FBCS).