Need for Speed: Taming Backdoor Attacks with Speed and Precision

Zhuo Ma[1], Yilong Yang[1,‡], Yang Liu[1,‡], Tong Yang[2], Xinjing Liu[1], Teng Li[1], Zhan Qin[3]
[1] Xidian University, [2] Peking University, [3] Zhejiang University

***Abstract*—Modern deep neural network models (DNNs) require extensive data for optimal performance, prompting reliance on multiple entities for the acquisition of training datasets. One prominent security threat is backdoor attacks where the adversary party poisons a small subset of training datasets to implant a backdoor into the model, leading to misclassifications during runtime for triggered samples. To mitigate the attack, many defense methods have been proposed, such as detecting and removing poisoned samples or rectifying trojaned model weights in victim DNNs. However, existing approaches suffer from notable inefficiency as they are faced with large-scale training datasets, consequently rendering these defenses impractical in the real world. In this paper, we propose a lightweight backdoor identification and removal scheme, called REBACK. In this scheme, REBACK first extracts a subset of suspicious and benign samples, and then, proceeds with a "averaging and differencing" based method to identify target label(s). Next, leveraging the identification results, REBACK invokes a novel reverse engineering method to recover the exact trigger using only basic arithmetic atoms. Our experiments demonstrate that, for ImageNet with 750 labels, REBACK can defend against backdoor attacks in around 2 hours, showcasing a speed improvement of $18.5\times$ to $214\times$ compared to existing methods. For backdoor removal, the attack success rate can be decreased to 0.05% owing to 99% cosine similarity of the reversed triggers. The code is online available.**

1. Introduction

In recent years, backdoor attacks become a prominent threat to Deep Neural Network models (DNNs) [1] and received substantial attention in the security domain [2]–[4]. In essence, backdoors are training attacks. These attacks involve poisoning a small portion of the training dataset with pre-defined hidden patterns, known as triggers. DNNs trained with such poisoned training data are consequently backdoored. When provided with benign inputs, these backdoored DNNs behave correctly, but they produce attacker-specified outputs when exposed to triggered inputs [5]–[7].

Many defense methods have been proposed to mitigate the threat of backdoor attacks when training a clean model on an untrusted dataset. They can be broadly classified into dataset-level defences and model-level defences based on the processing objectives. Dataset-level defenses focus on purifying untrusted dataset by eliminating poisoned samples.

‡ *Corresponding Authors: Yilong Yang and Yang Liu.*

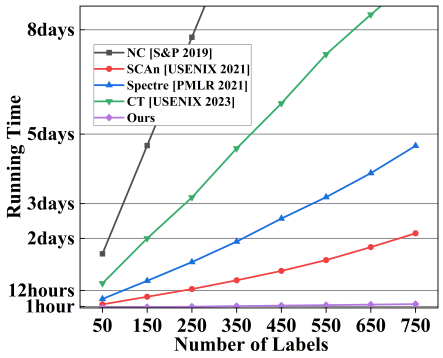


Figure 1: Running time of different backdoor defenses on ImageNet. Our scheme only needs 2.4 hours to tame backdoor attacks when the number of labels is 750, with a $18.5\times$ to $214\times$ improvement over other defenses.

Depending on the eliminating stage, they can be further categorised into before-training schemes (e.g., SCAn [8], Spectre [9], CT [10]) and during-training schemes (e.g., ABL [11], DBD [12], ASD [13]). Model-level defenses (e.g., NC [14], ABS [15], K-arm [16]) aim to rectify polluted weights in victim DNNs associated with trigger patterns.

Nevertheless, existing studies overlook a crucial aspect: *efficiency*. To exemplify this, we conducted runtime tests on four representative approaches, namely SCAn, Spectre, CT, and NC, across varying numbers of labels. The results are shown in Figure 1. Here, the larger the number of labels, the larger the untrusted training dataset is. As we can see, these defenses struggle to meet the real-world timeliness requirements as dataset size increases. For example, CT needs 9 days to defend the trojaned model with 750 labels. In turn, such inefficiency renders these defenses impractical.

Why do current backdoor defenses suffer from dramatically low efficiency? *1). Dataset-level defense:* Before-training schemes must perform computationally expensive mathematical transformations on each sample, such as computing covariance metrics or whitening. During-training schemes typically scan the entire dataset to split poisoned samples at each training epoch, often accompanied by computationally intensive training procedures. *2). Model-level defense:* A mainstream idea is first to obtain the triggers through reverse engineering, and then, use them to cleanse the models. However, current trigger reverse process often requires iterative complex calculation of gradients to guide the optimization of triggers. Moreover, the reverse process must be repeated for all target labels, incurring unavoidable computational costs due to the need for label determina-

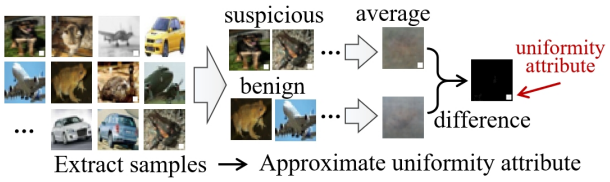


Figure 2: An intuitive example of "uniformity attribute" on BadNets attack and CIFAR10 dataset.

tion [14]. More detailed complexity analysis is given in Section 4.1.

Our Approach. This paper proposes a novel reversing engineering-based dataset-level defense against backdoor attacks (REBACK), designed to tame backdoor attacks while addressing critical efficiency concerns. Unlike existing defenses that process the entire dataset, REBACK optimizes efficiency by identifying target label(s) only on a small subset of samples. Subsequently, REBACK employs a novel reverse engineering method from a distinct data perspective to accurately reveal triggers. The following details our breakthroughs on each step.

Step 1: Extract a small portion of suspicious and benign samples from the untrusted dataset. Previous studies [2], [11], [17] indicate that poisoned samples exhibit greater learnability and lower entropy compared to benign samples. Following this insight, REBACK first trains a model for *a few epochs* on the untrusted dataset. Then, REBACK extracts a small portion of *suspicious* samples with low entropy values for each label. While the samples with relatively higher entropy are marked as "*benign*".

Step 2: Identify the backdoor attack target label(s) using suspicious and benign samples. Subsequently, we observe that suspicious samples have an inherent characteristic: a high degree of uniformity at the trigger position, as illustrated in Figure 2. We approximate the uniformity attribute by simply "averaging and differencing" on suspicious and benign samples (detailed in Algorithm 2). REBACK then applies this approximated uniformity attribute to benign samples and identifies target label(s) by detecting their label transfer rates. In cases where no such labels emerge, the untrusted dataset is deemed clean. Note that the approximated attribute is only used to identify target label(s).

Step 3: Reverse different patterns of triggers. Upon identifying the target label(s), REBACK employs innovative trigger reverse engineering to reconstruct precise triggers for subsequent backdoor removal. The core idea centers on understanding the formulation of poisoned samples. At a high level, poisoned samples $\boldsymbol{x}_p$ can be formulated as $\boldsymbol{x}_p = (1-\boldsymbol{m})\cdot\boldsymbol{x}_b+\boldsymbol{m}\cdot\boldsymbol{\Delta}$, where $\boldsymbol{x}_b$ is the raw samples, $\boldsymbol{\Delta}$ is the trigger with the same dimension as $\boldsymbol{x}_b$, and $\boldsymbol{m}$ denotes the mask [14]. Existing mainstream backdoor triggers $\boldsymbol{\Delta}$ can be categorized into two patterns, namely replace pattern (e.g., BadNets [18]) and blend pattern (e.g., Blend [19]). For replace pattern, the entries of $\boldsymbol{m}$ are set to 1 at the trigger position, and the others are set to 0. For blend pattern, the entries of $\boldsymbol{m}$ can be any arbitrary values, determined by the attacker. First, REBACK automatically distinguish trigger types without prior knowledge of triggers (see Section 5.2).

- *Replace pattern.* The uniformity attribute across the poisoned samples indicates that all poisoned samples have the same values (i.e., $\boldsymbol{\Delta}$) at the position where $\boldsymbol{m} = 1$. This is an outward signature that the backdoor attack will work. In other words, the distance between the poisoned samples is 0 at these positions. From the perspective of the defender, it is very dubious if there exist some samples whose distances are all 0 at the same position in the suspicious dataset. Therefore, we locate triggers by testing for the presence of the same pixels between the suspicious samples, called the distance test (Algorithm 3).
- *Blend pattern.* The prominent challenge is that with unknown $\boldsymbol{m}$, the defender cannot remove the mixing of $\boldsymbol{\Delta}$ from $\boldsymbol{x}_p$. The following observation motivates us to embrace this challenge. Consider an ideal condition where there are two raw pixels ($\boldsymbol{x}_0$ and $\boldsymbol{x}_1$), whose entries are set to 0 and 1, respectively. Correspondingly, the values after backdoor injection are $\boldsymbol{x}'_0 = \boldsymbol{m}\cdot\boldsymbol{\Delta}$ and $\boldsymbol{x}'_1 = (1-\boldsymbol{m})+\boldsymbol{m}\cdot\boldsymbol{\Delta}$. In this setup, the defender can get approximated $\boldsymbol{m}$ by solving the above equations: $\boldsymbol{m} \approx 1-(\boldsymbol{x}'_1-\boldsymbol{x}'_0)$. And then, obtain $\boldsymbol{\Delta} = \boldsymbol{x}'_0 \div \boldsymbol{m}$. The general cases are detailed in Section 5.2.

As the triggers are reversed, we can easily utilize them to cleanse a trojaned model with machine unlearning [7] or fine-pruning [20]. Above all, since REBACK only needs a quick shot on all samples and focuses on a very small portion of samples to reverse trigger, its overall complexity is much lower than existing work (see Section 4.1).

Contributions. The following concludes our contributions.

- We develop a lightweight scheme, called REBACK, to tame backdoor attacks with precisely reversed triggers. Even for ImageNet dataset with 750 labels and 750000 samples, the defense of REBACK can be completed in only about 2 hours, showcasing an efficiency improvement of 18.5× to 214×.
- REBACK can quickly identify the type of backdoor without any prior knowledge and involves two trigger reverse engineering methods using only basic arithmetic atoms. Even for ImageNet dataset, REBACK can complete trigger reversing in 0.01 seconds, while some existing work need days to accomplish.
- We implement and validate REBACK on six datasets and a variety of state-of-the-art attacks. Extensive experiments show that with REBACK, the attack success rates of state-of-the-art attacks can be decreased to 0.05%, owing to the 99% cosine similarity between reversed triggers and original triggers.

2. Preliminary & Background

2.1. Backdoor Attacks

Overview. Backdoor attacks are highly threatening against DNNs trained with publicly collected data. By inserting only a small part of adversarially crafted data into the training dataset, the backdoor attacker can induce the model

to learn a specific trigger pattern. The backdoored DNN is activated to perform attack-specified behavior as given triggers as inputs while exhibiting normally on clean inputs. Uniformly, we can formalize a typical backdoor injection into the following format [14]:

$$\boldsymbol{x}_p = (1 - \boldsymbol{m}) \cdot \boldsymbol{x}_b + \boldsymbol{m} \cdot \boldsymbol{\Delta}. \tag{1}$$

Suppose $\boldsymbol{x}_b$ is 3D pixel matrix for a benign sample and $\boldsymbol{x}_p$ is the corresponding poisoned sample. $\boldsymbol{\Delta}$, the backdoor trigger, is also a 3D matrix with same dimension and input space as $\boldsymbol{x}_b$. $\boldsymbol{m}$ is 2D matrix, called mask, indicating how much the trigger overwrites $\boldsymbol{x}_b$. Normally, different channels of $\boldsymbol{x}_b$ share the same mask[1]. The input space of $\boldsymbol{m}$ depends on the trigger patterns. Referring to [4], [19], there are two patterns of triggers commonly discussed, namely replace pattern and blend pattern:

Replace Pattern. This trigger pattern sets $\boldsymbol{m}$ to 1 at the trigger positions while setting $\boldsymbol{m}$ to 0 at other positions. Mark the size of 3D input space as (C, W, H). If $\boldsymbol{m}^{i,j} = 1$, the input masked with a replace-pattern trigger can be formalized as:

$$\boldsymbol{x}_p^{c,i,j} = \boldsymbol{\Delta}^{c,i,j}, \tag{2}$$

where $c \in [C]$, $i \in [W]$ and $j \in [H]$ are position indexes. Otherwise, there exists $\boldsymbol{x}_p^{c,i,j} = \boldsymbol{x}_b^{c,i,j}$.

Blend Pattern. This trigger pattern can set $\boldsymbol{m}$ to any values between 0 and 1. In particular, blend-pattern triggers can be masked on the whole image or only specific part positions. For generality, we formalize the data masked with blend pattern trigger as:

$$\boldsymbol{x}_p^{c,i,j} = (1 - \boldsymbol{m}^{i,j}) \cdot \boldsymbol{x}_b^{c,i,j} + \boldsymbol{m}^{i,j} \cdot \boldsymbol{\Delta}^{c,i,j}. \tag{3}$$

Otherwise, there still exists $\boldsymbol{x}_p^{c,i,j} = \boldsymbol{x}_b^{c,i,j}$. From the above formulation, the pixels corresponding to a mask of 0 are unmodified. If part of the entries of $\boldsymbol{m}$ is 0, we called this type part-blend attacks in this paper.

2.2. Related Work of Backdoor Defenses

To obtain a clean model on an untrusted dataset, existing mainstream backdoor defenses are divided into three major lines: 1) dataset-level, 2) model-level, and 3) other defenses.

Dataset-level. This work aims to detect poisoned samples in the dataset upon latent separation characteristics before or during training process. *Before Training*: Tran et al. [21] employ a method where latent representations are projected onto the primary principal component analysis direction, revealing a bimodal distribution where poisoned and benign samples form distinct modes. Then, Chen et al. [22] distinguish between poison and benign clusters in the latent space by means of the K-means cluster. In recent developments, Tang et al. [8] and Hayase et al. [9] enhance the aforementioned cluster analysis by introducing the use of statistics derived from the benign distribution. Additionally, the work contemporaneous with ours, Qi et a al. [10] actively decouples benign correlation while exposing backdoor patterns to detection by confusion training. *During Training*: Li et al. [11] maintain a polluted pool by evaluating the speed of sample fitting state, which are unlearning during training. However, the polluted pool is fixed during the whole training process. To mitigate this, Huang et al. [12] initializes two dynamically updated sample pools and the model is fine-tuned by semi-supervised learning. To more accurately distinguish poisoned samples, Gao et al. [13] introduce meta-learning-inspired split to locate clean hard samples which that are hard distinguished by loss-guided split methods.

Model-level. This work aims to directly adjust the trojaned model by analyzing its behavior to remove backdoor attacks, including trigger reverse engineering [14], [16] and neuron stimulation [15]. The pioneering work of trigger reverse engineering, NC [14], advocates reversing a trigger for each label and subsequently applying outlier detection to identify the final triggers. Shen et al. [16] enhance NC by integrating Multi-Arm Bandit techniques to reduce the complexity. Inspired by the electrical brain stimulation technique used to analyze the human brain neurons, Liu et al. [15] scan a DNN model to determine compromised neurons and map them back to input space. For backdoor removal, Liu et al. [7] and Liu et al. [20] introduce machine unlearning and fin e-pruning to mitigate the compromised neurons.

Other. In this paper, we only talk about the context of obtaining a clean model on the untrusted training datasets. Thus, we omit the analysis of other tracks of backdoor defenses [23]–[25].

3. Threat Model

In this section, we describe the standard threat model of backdoor attacks and defenses [8], [10], [14].

Attacker's Capability. The attacker is a data provider and has full access to the training set of the victim model. The attacker can inject a small portion of poisoned samples into the training dataset. They can freely exploit trigger patterns but cannot interfere with the training procedure.

Defender's Capability. The defender is a model trainer and may collect training datasets from untrusted data providers. The defender has no knowledge about whether the training dataset is backdoored, which attack method is used by the attacker, and which samples are poisoned. After obtaining the poisoned training dataset, the defender has full autonomy to inspect it. This encompasses the capability to access, examine, and process the dataset, as well as independent model training. This approach diverges from previous research, such as [14], where the training process is delegated to a potentially malicious external entity. Moreover, we do not mandate that the defender must hold benign validation data [26]. Our assumption is therefore practical and meaningful, aligning with the scenarios considered in existing backdoor defenses [8]–[10].

Goals. Our defensive effort includes four specific goals:

1. REBACK is robust on different masks across channels (Figure 9).

TABLE 1: Overall time complexity of existing defenses.

	REBACK $O(\mathcal{N}+\mathcal{F}), \mathcal{F} \ll \mathcal{P}$		
Dataset -level	SCAn [8] $O(\mathcal{K}+\mathcal{N}\mathcal{C}_1+\mathcal{P})$	Spectre [9] $O(\mathcal{K}+\mathcal{N}\mathcal{C}_2+\mathcal{P})$	CT [10] $O(2\mathcal{N}\mathcal{C}_3+\mathcal{P})$
	ABL [11] $O(\mathcal{N}\mathcal{E}_e\mathcal{C}_4+\mathcal{P})$	DBD [12] $O(\mathcal{N}\mathcal{E}\mathcal{C}_5+\mathcal{P}_{sl})$	ASD [13] $O(\mathcal{N}\mathcal{E}\mathcal{C}_6+\mathcal{P}_{sm})$
Model -level	NC [14] $O(\mathcal{N}_o\mathcal{I}\mathcal{L}+\mathcal{F})$	K-arm [16] $O(\mathcal{N}_o\mathcal{I}log\mathcal{L}+\mathcal{F})$	ABS [15] $O(\mathcal{M}^{\mathcal{Q}}+\mathcal{F})$

- *Detecting the existence of backdoor attacks.* The objective is to determine the presence of backdoor attacks within a given dataset and identify label(s) targeted by the attack.
- *Reversing backdoor triggers.* The defender aims to reverse engineer triggers within the untrusted dataset to uncover potential backdoor mechanisms.
- *Mitigating backdoor attacks.* The defender's goal is to neutralize the impact of backdoor attacks during the model training on the untrusted dataset.
- *Defense efficiency.* Foremost, the defender strives to complete the defense process within a short timeframe, making it practical for real-world applications, where a substantial number of samples need to be inspected.

4. Limitation of Prior Work

In this section, we focus on two limitations of prior work: 1) high computation complexity (see Section 4.1) and 2) the imprecise localization of compromised neurons due to imprecise reversed triggers (see Section 4.2).

4.1. High Computation Complexity

Here, we elaborate our analysis about time complexity of nine representative backdoor defense approaches, including *dataset-level* defenses: SCAn, Spectre, CT, ABL, DBD, ASD, and *model-level* defenses: NC, K-arm, and ABS. The complexity results are also summarized in Table 1.

Dataset-level Defense. Suppose the number of labels and samples is $\mathcal{L}$ and $\mathcal{N}$. Since SCAn and Spectre need to extract the latent feature representation of the entire dataset and retrain the final model, we add $O(\mathcal{K}+\mathcal{P})$ complexity for them. $\mathcal{C}_i$ denotes the complexity of the operations that need to be done on each sample.

- *SCAn* estimates mixture model parameters for each label in the training dataset and anomaly likelihood is calculated. Its workflow involves estimating the covariance metrics, computing the identity vector for all labels, estimating subgroup parameters using an iterative method on all samples, and performing the likelihood ratio test using the mixture model. The complexity is $O(\mathcal{K}+\mathcal{N}\mathcal{C}_1+\mathcal{P})$.
- Like SCAn, *Spectre* consists of three operations on all samples: dimensionality reduction, whitening, and calculation of QUE values. The complexity is $O(\mathcal{K}+\mathcal{N}\mathcal{C}_2+\mathcal{P})$.
- *CT* amplifies latent representations through a training procedure that inhibits the learning of benign samples by randomly mislabeling them. The primary computational cost comes from multiple pretraining procedures on all samples, approximately training all samples twice. The complexity is $O(2\mathcal{N}\mathcal{C}_3+\mathcal{P})$.
- *ABL* isolates some samples with the lowest losses in early epochs $\mathcal{E}_e$, then unlearn isolated samples during the last few training epochs. The complexity is $O(\mathcal{N}\mathcal{E}_e\mathcal{C}_4+\mathcal{P})$.
- *DBD* trains the model via self-supervised learning based on label-removed training samples. Suppose the training epoch is $\mathcal{E}$, the complexity is $O(\mathcal{N}\mathcal{E}\mathcal{C}_5+\mathcal{P}_{sl})$, where $\mathcal{P}_{sl}$ is the complexity of self-supervised learning.
- *ASD* updates poisoned samples at each epoch with Loss-guided Split or Meta-Split. The complexity is $O(\mathcal{N}\mathcal{E}\mathcal{C}_6+\mathcal{P}_{sm})$, where $\mathcal{P}_{sm}$ is the complexity of semi-supervised learning.

Model-level Defense. Let the number of samples used in optimization be $\mathcal{N}_o$ and $\mathcal{I}$ be the time to interact with the trojaned model over one sample. Since all model-level defenses need to cleanse the trojaned model, we add $O(\mathcal{F})$ complexity.

- *NC* performs a multi-objective optimization procedure on all labels to obtain the L1 norm values of each potential trigger. The complexity is $O(\mathcal{N}_o\mathcal{I}\mathcal{L}+\mathcal{F})$.
- *K-arm* reduces NC's computation complexity by introducing Multi-Arm Bandit bandit problem. At each epoch, the probability of choosing the label with a smaller L1 norm to continue optimization is higher, until the threshold is reached. Theoretically, the complexity is $O(\mathcal{N}_o\mathcal{I}log\mathcal{L}+\mathcal{F})$ [16].
- *ABS* intercepts and changes the neuron activation values on benign samples, and then observes whether this manipulation consistently leads to misclassification. If so, the corresponding neurons are marked as compromised and are used to reverse triggers through a similar optimization process like NC. Assume each layer has $\mathcal{M}$ neurons. The final complexity is $O(\mathcal{M}^{\mathcal{Q}}+\mathcal{F})$, where $\mathcal{Q}$ is the number of layers.

Summary. From the complexity analysis results, we can observe that two main problems cause the inefficiency of existing schemes: 1) working on the entire dataset; 2) requiring a complicated iterative optimization process. As such, REBACK is motivated to avoid these two problems by exploring a new backdoor defense diagram.

4.2. Imprecise Trigger Reverse

Trigger reverse has been a hard task in the backdoor defense field. A prominent limitation of existing trigger reverse methods [14], [16], [27] is that they still follow an "exhaustive search-like" idea. More precisely, these methods are basically designed to find different optimization ways to search the entire input space and determine a specific input that works like a trigger. Clearly, the hardness of such an idea is positive to the size of triggers. As the trigger size is larger or even covers the whole image, the existing trigger reverse methods can hardly achieve a good performance. What's worse, state-of-the-art backdoor attacks [16], [27]

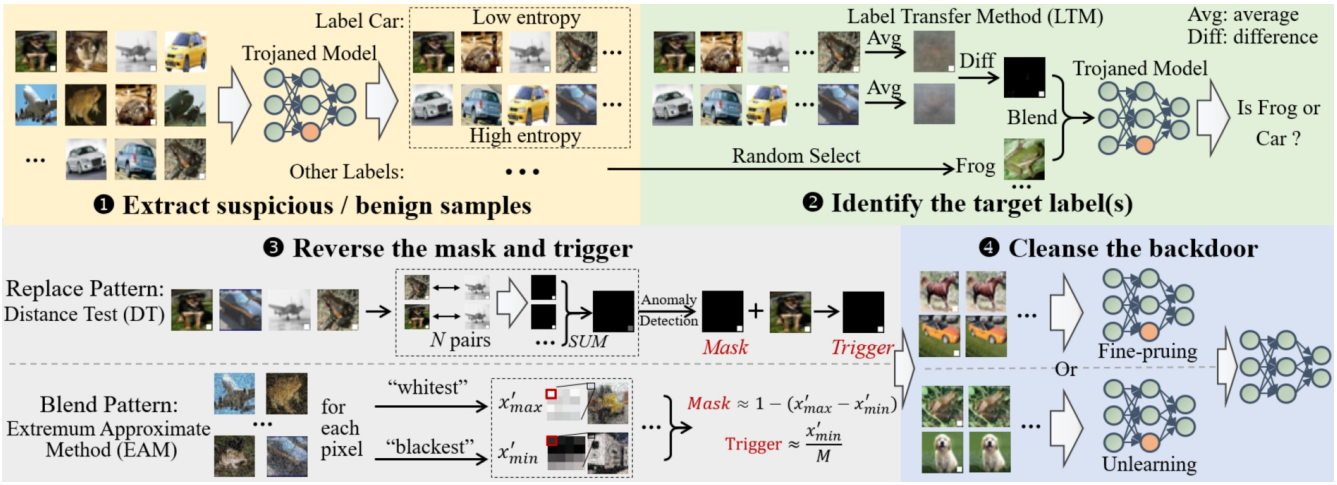


Figure 3: Workflow of REBACK. Step ❶: extract suspicious and benign samples from the entire dataset; Step ❷: identify the suspicious target label(s) based on these samples; Step ❸: reverse the mask and trigger against replace and blend patterns; Step ❹: cleanse the trojaned model using the reversed trigger.

are still moving towards this direction, although having a flexible trigger selection strategy. Especially, most of the above works only evaluate their effectiveness via the ASR with the reversed trigger, while ignoring a key metric, that is, the similarity between the reversed and real triggers. The fact is due to the strong robustness of backdoor attacks, an imprecisely reversed trigger can still be used to launch a successful attack but easily misleads the defender to wrongly cleanse benign neurons. Our experiments in Table 10 also validate such a statement.

5. Approach

5.1. Defense Overview

At a high level, REBACK comprises four steps to defend against backdoor attacks as shown in Figure 3.

- Step ❶: REBACK extracts a small number of suspicious and benign samples from the entire dataset. The key knowledge employed is that poisoned (suspicious) samples have significantly lower entropy than benign samples.
- Step ❷: REBACK identifies the target label(s) of the backdoor attack through the Label Transfer Method (LTM, see Algorithm 2). For target labels, *uniformity attributes* approximated from suspicious and benign samples are likely to cause benign samples to be misclassified as the target label. In contrast, uniformity attributes in clean labels would hardly not. Utilizing this observation, LTM identifies target label(s) by assessing the label transfer ratio between the target and clean labels.
- Step ❸: REBACK automatically reverses the trigger in the replace and blend patterns using Distance Test (DT, see Algorithm 3) and Extremum Approximate Method (EAM, see Algorithm 4), respectively. Importantly, REBACK operates without prior knowledge of the triggers employed by attackers.
- Step ❹: After obtaining the reversed trigger, REBACK cleanses the trojaned model by combining fine-pruning [20] or machine unlearning [7] approaches.

In the following section, we discuss the technical details of REBACK.

5.2. Technical Details of REBACK

Step ❶ Extract suspicious and benign samples. Typically, an effective backdoor attack with high ASR builds an efficient activation pattern (i.e., shortcut) of the trigger on the trojaned model [28]. This shortcut enables the trojaned model to confidently misclassify the poisoned samples to the target label, resulting in a lower entropy level than that of benign samples. Inspired by this, REBACK first extracts K samples with the lowest entropy values for each label, constituting the *suspicious dataset*. Similarly, K samples with the highest entropy values are extracted as the *benign dataset*. Suppose that the suspicious and benign dataset with label L (the ground label) are $D_s^L = (X_s^L, Y_s^L)$ and $D_b^L = (X_b^L, Y_b^L)$. The entire extracted dataset for all labels are represented as $D_s = \bigcup_1^L D_s^i$ and $D_b = \bigcup_1^L D_b^i$.

Step ❷ Identify the target label(s). Unlike existing backdoor defense methods, REBACK first identifies the target label(s) by a lightweight scheme, i.e., the label transfer method (Algorithm 2). We observe a high degree of uniformity across the poisoned samples in the trigger position, including replace and blend patterns. This is because the backdoor injection does the same operation on the fixed positions of all poisoned samples. Based on the above idea, we approximate the uniformity attributes (denoted as Θ) through "averaging and differencing" operations. if a label is backdoored, its Θ will change the classification of other benign samples to the target label with high probability. Conversely, the clean label will not.

Algorithm 1 Trigger Reverse Engineering of REBACK.

Input: dataset $\mathcal{D}$, threshold ε, ϵ and Υ, sample number N.
Output: reversed mask $\hat{\boldsymbol{m}}$ and trigger $\hat{\Delta}$.

Train a model $\mathcal{M}$ with $\mathcal{D}$.
for $L = 1$ to $\mathcal{L}$ **do**
  Extract K samples with the lowest/highest entropy to form suspicious/benign dataset D_s^L and D_b^L.
  If *LTM*$(D_s^L, D_b^L, \mathcal{M}, L, \Upsilon)$ is *True*, put L in suspicious label set Ω.
end for
for $L \in \Omega$ **do**
  if $MAX(DT(D_s^L, N)) > \epsilon$ **then** ▷ Replace Pattern
    $\hat{\boldsymbol{m}} = SDOD(DT(D_s^L, N), \varepsilon)$.
    $\hat{\Delta} = \boldsymbol{m} \odot \mathcal{S}$, where $\mathcal{S} \in D_s^L$.
  else ▷ Blend Pattern
    for $(i, j, c) \in ([C], [W], [H])$ **do**
      Find the maximum and minimum values $x_{max}^{c,i,j}$ and $x_{min}^{c,i,j}$ in D_s^L, the maximum and minimum values $p_{max}^{c,i,j}$ and $p_{min}^{c,i,j}$ in D_b^L.
      $\hat{\boldsymbol{m}}^{c,i,j} \approx 1 - (\frac{\boldsymbol{x}_{max}^{c,i,j} - \boldsymbol{x}_{min}^{c,i,j}}{\boldsymbol{p}_{max}^{c,i,j} - \boldsymbol{p}_{min}^{c,i,j}})$
    end for
    For $(c, i, j) \in ([C], [W], [H])$
      $\hat{\Delta}^{c,i,j} \approx \frac{\boldsymbol{x}_{min}^{c,i,j} - (1 - \hat{\boldsymbol{m}}^{c,i,j})\boldsymbol{p}_{min}^{c,i,j}}{\hat{\boldsymbol{m}}^{c,i,j}}$.
  end if
end for
Return reversed mask $\hat{\boldsymbol{m}}$ and trigger $\hat{\boldsymbol{\Delta}}$.

Specifically, for label L, LTM randomly selects[2] two other labels (L_1 and L_2) and corresponding benign dataset $D_b^{L_1}$ and $D_b^{L_2}$. Then, LTM extracts uniformity attributes of L and L_2 based on L_1 through "averaging and differencing": $\Theta_s = AVG(D_s^L) - AVG(D_b^{L_1})$ and $\Theta_b = AVG(D_b^{L_2}) - AVG(D_b^{L_1})$. To test label transfer ratio, LTM randomly chooses S sample(s) $X_b^{L_1}$ from $D_b^{L_1}$ and computes $\pi_s = \sum(\mathcal{M}(X_b^{L_1} + \Theta_s) == L)/S$ and $\pi_b = \sum(\mathcal{M}(X_b^{L_1} + \Theta_b) == L)/S$. π_s and π_b represent the label transfer rate of the uniformity attributes of the suspicious dataset and that of the benign dataset. Given a threshold Υ, if $\pi_s/\pi_b > \Upsilon$, LTM reports L as the suspicious target label. Experiments in Table 5 have proven that one sample can achieve 91% accuracy in BadNets. Additional analysis of the feasibility of LTM is detailed in Appendix A.1.

Step ❸ Reverse the mask and trigger. Different from the uniformity attributes, here the precise trigger that can be used to tame the backdoor attacks must be reversed. In this step, we propose Distance Test (DT) and Extremum Approximate Methods (EAM) for replace and blend patterns, respectively.

Discussion. For replace patterns like BadNets, since the trigger inserted into different samples is identical, the entry distribution at the trigger position significantly differs from other entries. Therefore, this inspires us that detecting the anomaly points can effectively identify replace pattern triggers. An intuitive idea is to directly perform anomaly detection on the uniformity attributes to locate triggers, which we call the Uniformity Attribute Locating method (UAL). Related experiments, see Appendix A.3, indicate that UAL has optimal results on existing replace pattern attacks than state-of-the-art reverse engineering methods. However, due to the diversity of dataset distribution, the anomaly detection threshold has a large impact on the performance. Worse, if the attacker intentionally uses the average of poisoned samples as the trigger, the UAL has little effect. However, we observe that UAL is robust in detecting the trigger position of *part-blend pattern* attacks, detailed as *Case 1* in Blend pattern.

Algorithm 2 Label Transfer Method (LTM).

Input: suspicious/benign dataset D_s^L/D_b^L, model $\mathcal{M}$, label L, threshold Υ.
Output: *True* or *False*.

Randomly select two labels L_1 and L_2 different from L , and their benign dataset denoted as $D_b^{L_1}$ and $D_b^{L_2}$.
Approximate uniformity attributes through "averaging and differencing": $\Theta_s = AVG(D_s^L) - AVG(D_b^{L_1})$ and $\Theta_b = AVG(D_b^{L_2}) - AVG(D_b^{L_1})$.
Randomly select S samples $X_b^{L_1}$ from $D_b^{L_1}$, and calculate the proportion $\pi_s = \sum(\mathcal{M}(X_b^{L_1} + \Theta_s) == L)/S$ and $\pi_b = \sum(X_b^{L_1} + \Theta_b) == L)$.
If $\pi_s/\pi_b > \Upsilon$ **Return** *True* else **Return** *False*

Algorithm 3 Distance Test (DT).

Input: suspicious dataset D_s^L, sample number N.
Output: ratio P.

Randomly select N unique pairs of samples from D_s^L.
Compute mask M^r indicating positions where the distance between samples in the $r{-}th$ pair is zero, $r \in [N]$.
Compute proportion $P = \sum_{i=1}^{N}(M^i)/N$.
Return ratio P.

- Replace pattern - Distance Test: Algorithm 3 shows the procedure of distance test. This algorithm accepts a suspect dataset D_s^L of size K and a specified sampling number N, where N takes the value range $(0, C_K^2]$. First, N distinct sample pairs are randomly selected. Subsequently, DT computes N mask positions M^r that express the pixel's distance between $r - th$ sample pair as zero[3]. $P = \sum_{i=1}^{N}(M^i)/N$ thus represents the ratio of selected sample pairs that is zero at a given pixel position. If L is backdoored, the entry of P at the trigger position is exceptionally larger than other positions. On this basis, we use anomaly detection (Algorithm 4) to finally locate the trigger position (i.e., $\boldsymbol{m}$). Although a small number of

2. It is reasonable to assume that there exist clean targets in the dataset, as we find that when the number of attack targets is too high, the backdoor effect decreases dramatically. The experiments of Multiple Triggers for Multiple Target Labels (MTMT) attacks prove that REBACK is robust.

3. Or less than a smaller value if the attacker can adaptively perturb the trigger across different samples.

benign samples reduce the value at the trigger position, the values also remain anomalous (see Figure 6).

How to distinguish replace and blend patterns? As discussed above, high zero distance ratios are essentially undetectable in clean datasets. So do the poisoned dataset with blend attack: Suppose the clean dataset X_c and corresponding poisoned dataset X_p, where $X_p = (1-\boldsymbol{m})\cdot X_c + \boldsymbol{m}\cdot\boldsymbol{\Delta}$. The distance between X_p is equal to that between X_c. Hence, $DT(X_c, N) \approx DT(X_p, N)$. Given a threshold ϵ, if the maximum ratio returned by DT is larger than ϵ, REBACK considers the backdoor attack as blend pattern.

Algorithm 4 Standard Deviation Outlier Detection (SDOD).

Input: data ϖ, threshold ε.
Output: mask M.

1: Compute the mean and standard deviation: $\mu = mean(\varpi)$ and $\sigma = std(\varpi)$.
2: Get mask $M_i = 1$ if $ABS(M_i - \mu) > \varepsilon * \sigma$ else 0.
3: **Return** mask M.

- Blend pattern - Extremum Approximate Method: As discussed in Section 2, the blend pattern injection process in backdoor attacks can be formalized as:

$$\boldsymbol{x}_p^{c,i,j} = (1-\boldsymbol{m}^{i,j})\cdot\boldsymbol{x}_b^{c,i,j} + \boldsymbol{m}^{i,j}\cdot\boldsymbol{\Delta}^{c,i,j}. \quad (4)$$

Take the example of an image dataset with pixel values ranging from 0 to 1, where 0 means black and 1 means white. Note that we illustrate EAM of REBACK under the situation where all pixels share same mask value and different channels share same trigger value. When pixel at i, j position is black, the corresponding pixel of the poisoned sample is $\boldsymbol{x}_{black}^{i,j} = (1-\boldsymbol{m}^{i,j})\cdot 0 + \boldsymbol{m}^{i,j}\cdot\boldsymbol{\Delta}^{i,j} = \boldsymbol{m}^{i,j}\cdot\boldsymbol{\Delta}^{i,j}$. Similarly, when the pixel is white, the corresponding poisoned pixel is computed by $\boldsymbol{x}_{white}^{i,j} = (1-\boldsymbol{m}^{i,j}) + \boldsymbol{m}^{i,j}\cdot\boldsymbol{\Delta}^{i,j}$. Thus, associating the above two equations, we can approximately get: $\boldsymbol{m}^{i,j} \approx 1 - (\boldsymbol{x}_{white}^{i,j} - \boldsymbol{x}_{black}^{i,j})$.
In general, REBACK finds the blackest ($\boldsymbol{x}_{min}^{i,j}$) and whitest values ($\boldsymbol{x}_{max}^{i,j}$) for each pixel from the suspicious dataset D_p^L, getting τ m, where τ is the number of pixels. In addition, to exclude the effect of extreme pixels, we provide Histogram Peak Detection (HPD) to find optimal $\hat{\boldsymbol{m}}$, detailed in Algorithm 5. to remove the range restriction for different pixel spaces, the, for each pixel, the defender can search D_b to determine the input spaces $[\boldsymbol{p}_{min}, \boldsymbol{p}_{max}]$, and then, compute m via the following equation:

$$\boldsymbol{m}^{i,j} \approx 1 - \left(\frac{\boldsymbol{x}_{max}^{i,j} - \boldsymbol{x}_{min}^{i,j}}{\boldsymbol{p}_{max} - \boldsymbol{p}_{min}}\right). \quad (5)$$

Finally, REBACK obtains the trigger values over the whole image through

$$\hat{\Delta}^{i,j} \approx \frac{(\boldsymbol{x}_{min}^{i,j} - (1-\hat{\boldsymbol{m}}))\cdot\boldsymbol{p}_{min}}{\hat{\boldsymbol{m}}}, \quad (6)$$

where $\hat{\boldsymbol{m}} = HPD(\boldsymbol{m}^{i,j}), (i,j) \in ([W],[H])$. In the following, we discuss two special cases in blend pattern.

Case 1: Part-blend pattern (injecting triggers on only a portion of an image). The effectiveness of EAM for part-blend pattern lightly decreases because $\boldsymbol{m}$ derived on pixels without trigger injection is meaningless. Thus, the trigger injection position needs to be determined first. As discussed in UAL, we find that the pixel values on the trigger injection position remain anomalous. REBACK recovers the mask, i.e., the trigger position, by UAL and on this basis recovers the trigger values by EAM. Note that HPD is needed to find optimal $\hat{\boldsymbol{m}}$.
Case 2: Whole-Image pattern (Different $\boldsymbol{m}$ among channels and pixels). We consider that there is another enhanced blend attack, where each pixel on different channels corresponds to completely different masks and triggers. In this case, Equation 3 becomes

$$\boldsymbol{x}_p^{c,i,j} = (1-\boldsymbol{m}^{c,i,j})\cdot\boldsymbol{x}_b^{c,i,j} + \boldsymbol{m}^{c,i,j}\cdot\boldsymbol{\Delta}^{c,i,j}. \quad (7)$$

In this regard, the difference with standard EAM is that the step of finding optimal $\hat{\boldsymbol{m}}$ on the whole image is canceled. From the experimental results in Figure 9, EAM is also robust under this enhanced attack.

Step ❹ Cleanse the backdoor. After obtaining the reversed trigger, we can easily leverage it with some existing schemes[4]. Prior work [7], [14] shows that by combining with machine unlearning or fine-pruning, the reversed trigger can be used to cleanse the trojaned model. Note that as a specific trigger is given, only little effort is required to implement these additional backdoor cleansing methods. Thus, the last step will not degrade the efficiency of REBACK significantly.

5.3. Efficiency Complexity of REBACK

Given a training dataset, REBACK first obtains the entropy of all samples, the complexity of which is $O(\mathcal{N})$. To identify target label(s), LTM needs to extract S samples on each label. To reverse trigger, DT and EAM only need to conduct atomic operations, such as addition, multiplication, etc., on K samples. Since $S, \mathcal{L}, K$ are all much less than $\mathcal{N}$, i.e., $S, \mathcal{L}, K \ll \mathcal{N}$, we can omit the complexity of the above three algorithms in REBACK. Denote the complexity of backdoor cleansing with the reversed trigger as $O(\mathcal{F})$. Overall, the complexity of REBACK is $O(\mathcal{N} + \mathcal{F})$. According to [14], [15], [20], there exists $\mathcal{F} \ll \mathcal{P}$. Therefore, combining with the analysis in Section 4.1, we say that the complexity of REBACK is much less than prior work.

6. Experiment

6.1. Experimental Setup

Attack Configurations. We use six benchmark datasets in our experiments, including CIFAR10 [31], CIFAR100 [31],

4. Besides, for replace patterns, by evaluating the similarity between the reversed trigger and each sample, we can directly mark and remove the poisoned samples (see Appendix A.5).

TABLE 2: Detailed default information about dataset, attack, and other parameters.

Attack	BadNets [18]	Checkerboard [18]	TrojanNN [29]	TrojanNNb [29]	SIG [30]	Blend Attack [19]
Dataset	CIFAR10 [31]	PUBFIG [32]	GTSRB [33]	CIFAR100 [31]	TinyImage [34]	ImageNet [34]
Labels/Sizes	10/60000	83/13838	13/5168	100/60000	200/200000	16/16000
Parameter	4*4	32*32	32*32	0.4 + 9*9	0.2	0.2
Sample Size	3*32*32	3*100*100	3*64*64	3*32*32	3*64*64	3*224*224
Model	ResNet18	ResNet18	ResNet18	ResNet18	ResNet18	ResNet50

TABLE 3: Running time of different backdoor defenses under different stages. Other defenses are detailed in Table 15.

Running times (s)	Pre-train model	Dataset-level / Model-level defenses								Cleanse the Model		
		NC	ABS	SCAn	CT	ABL	ASD	REBACK Step ❶❷	REBACK Step ❸	NC Finetune	ABS, REBACK Fine-pruning	SCAn, CT Retrain
BNets	170.32	459.14	597.93	633.19	2017.54	1614.41	7857.39	**2.42**	**0.01**	8.44	47.54	170.32
ChB	49.36	11461.47	561.65	5870.44	4100.55	428.16	9932.17	**2.64**	**0.01**	2.86	103.71	49.36
TrNN	19.55	801.97	600.35	1922.45	2193.90	114.98	3515.48	**0.34**	**0.01**	1.17	19.56	19.55
TrNNb	175.34	4350.98	630.07	4612.78	2228.99	1714.69	8138.72	**9.26**	**0.01**	8.97	67.15	175.34
SIG	403.29	13175.45	2880.19	11948.95	5136.84	3797.19	15096.87	**45.26**	**0.02**	18.41	398.37	403.29
Blend	2279.76	42907.85	6388.93	2499.85	21948.48	7237.67	51352.39	**64.74**	**0.02**	135.99	628.82	2279.76

GTSRB [33], PubFig [32], TinyImage [34], and ImageNet [34]. We consider six backdoor attacks: BadNets (BNets) [18], Checkerboard (ChB) [18], TrojanNN (TrNN) [29], Blend+TrojanNN (TrNNb) [29], Sinusoidal signal attack (SIG) [30] and Blend attack (Blend) [19]. The former three belong to the replace pattern, whereas the others are the blend pattern attacks. As typical model structures, ResNet18 [35] and ResNet50 (for ImageNet) are used as the base model. The detailed settings are summarized in Table 2. More analysis about clean label and dynamic backdoor attacks are detailed in Appendix A.4.

Defense Configurations. We compare REBACK[5] with ten state-of-the-art backdoor defense schemes, including 1) dataset-level defenses: SCAn[6] [8], Spectre[7] [9], CT[8] [10], ABL[9] [11], DBD[10] [12], ASD[11] [13], and 2) model-level defenses: NC[12] [14], ABS[13] [15], K-arm[14] [16], DTI [17]. DTI generates "patches" (as oppositely as triggers) using the same optimization as NC. Hence, we involve it in some experiments. For NC, we set the optimization iteration as 30 epochs and used 10% of training samples. For ABS, we find that the variation of its hyper-parameter max_mask_size has a large effect on the results. Thus, we follow the suggestions of its authors to test many hyper-parameter settings and show the optimal results under our experimental conditions. For CT, we find that a pretraining epoch of 10 is sufficient, which is 100 by default. For REBACK, by default, we set K in Step ❶ to 50 and 100 for replace and blend patterns.

5. Our source code is available at https://github.com/Echotoken/ReBack
6. https://github.com/TDteach/Demon-in-the-Variant
7. https://github.com/SewoongLab/spectre-defense
8. https://github.com/Unispac/Fight-Poison-With-Poison
9. https://github.com/bboylyg/ABL
10. https://github.com/SCLBD/DBD
11. https://github.com/KuofengGao/ASD
12. https://github.com/bolunwang/backdoor
13. https://github.com/naiyeleo/ABS
14. https://github.com/PurduePAML/K-ARM_Backdoor_Optimization

Evaluation Metrics. Similar to other work [10], [15], we adopt two commonly used metrics for evaluating the performance: Attack Success Rate (*ASR*) which measures the rate of classifying poisoned samples to the target label, and model ACCuracy (*ACC*) of benign samples. ASR ignores the misclassification accuracy of the clean model on the clean test dataset. For trigger reverse, we introduce three metrics to measure the fidelity of the reversed masks and triggers: *F1 score* for masks, and Mean Squared Error (*MSE*), Cosine Similarity (*CoSi*) for triggers. Detailed definitions of these metrics are given in Appendix A.2.

Other Settings. Most experiments are conducted on a Linux GPU server with an Intel(R) Core(TM) i7-5930K CPU, an Nvidia GeForce RTX 3090 GPU, and 80 GB RAM. For ImageNet, we utilize 2 GPUs to ensure enough computation resources for execution. All the experimental results are averaged over five trials. The optimizer is Adam with learning rate of 0.1, batch size of 128.

6.2. Efficiency Comparison.

Table 3 indicates the running times of six backdoor defenses. All before-training detaset-level need to extract latent representations from pre-trained model. So do model-level defenses and REBACK to rectify polluted weights in victim DNNs. An observed trend is a positive correlation between the running time of defenses and both the dataset size and model complexity, which is consistent with our complexity analysis in Section 4.1. The bright spot is that REBACK can reverse the mask and triggers in about 2 seconds for BNets, ChB and TrNN, which is 37× to 6267× faster than the rest of the reversed defenses. This is because REBACK only needs one forward prediction of all samples in Step ❶, eliminating the requirement for iterative optimization in identifying attack target labels. Conversely, in the case of SIG and Blend pattern attacks, the execution time is slowed

TABLE 4: Comparison of performance in ASR and ACC after retraining or fine-pruning, unlearning with different reversed triggers in the format of "mean (standard deviation)". The best performance of backdoor defenses is bolded. Other defenses are detailed in Table 11.

Defense (%)	SCAn		Spectre		CT		NC Fine-tune		REBACK				Original	
									Fine-pruning		Unlearning			
	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
BNets	1.67(0.92)	81.5(0.96)	0.91(0.31)	81.9(0.39)	0.52(0.37)	81.7(0.37)	1.08(0.76)	79.6(1.94)	**0.05(0.08)**	80.6(1.61)	1.79(0.56)	**82.8(0.76)**	99.09	82.75
ChB	68.3(25.1)	79.7(1.14)	35.1(13.2)	81.4(0.72)	0.08(0.05)	81.4(0.83)	8.79(2.57)	80.5(1.87)	**0.04(0.04)**	80.9(1.50)	3.97(1.80)	**82.1(0.51)**	99.34	81.71
TrNN	0.36(0.37)	97.8(0.51)	**0.06(0.08)**	97.1(1.31)	**0.06(0.06)**	98.6(0.9)	2.75(2.54)	98.6(0.71)	**0.06(0.04)**	97.9(0.56)	4.16(1.71)	**98.8(0.37)**	100	98.6
TrNNb	76.4(21.1)	51.8(1.47)	6.39(3.66)	52.3(0.75)	**0.08(0.02)**	52.6(0.74)	5.49(1.62)	52.8(1.57)	0.34(0.32)	52.1(0.84)	1.52(0.87)	**57.1(4.29)**	100	53.72
SIG	75.1(20.9)	92.2(1.37)	1.12(0.83)	93.3(1.70)	0.11(0.06)	94.1(1.27)	26.7(16.5)	92.3(1.52)	**0.02(0.10)**	92.6(1.11)	2.67(1.09)	**95.3(2.22)**	99.96	94.18
Blend	60.7(32.1)	76.2(2.39)	27.2(10.2)	75.9(3.34)	9.37(4.63)	78.2(0.29)	22.5(12.8)	78.0(1.66)	**0.47(0.47)**	77.5(0.92)	2.78(1.68)	**80.8(1.28)**	99.17	78.33

down due to massive samples and high model complexity. Also, the results show that our reverse engineering method in Step ❸ remains unaffected by variations in model complexity and dataset size. Moreover, we can notice that the efficiency of REBACK is mainly related to the number and size of training samples (also shown in Figure 1).

6.3. Mitigation of Backdoor Attacks

Table 4 shows the comparison of performance in ASR and ACC after cleansing the trojaned model or retaining without detected poisoned samples. The reason why NC is the only model-level defense is because it has the best performance, see Section 6.5. For NC, 10% of benign samples are extracted, keeping their labels unchanged, and the fine-tuned dataset is formed by adding the reversed triggers. Since we possess the reversed trigger, the key idea of Fine-pruning[15] is to prune neurons activated by the reversed trigger. We prune the neurons at the last convolution layer by feeding samples with the reversed triggers, followed by finetuning the pruned model ten epochs with 10% of benign samples[16]. We do our best to tune the pruning ratio of the neurons to achieve the optimal ASR and ACC performance[17], which mostly ranges from 1% to 20%. For backdoor removal method of unlearning[18], we set $\beta/\alpha = 1$ to avoid catastrophic forgetting. ABL and ASD are detailed in Appendix A.3.

The results demonstrate that REBACK can effectively remove the backdoor, reducing the ASR to nearly 0.02% through Fine-pruning. This is attributed to REBACK's precise identification of compromised neurons and the direct elimination of shortcut paths left by the backdoor. However, since these compromised neurons are still more or less related to benign samples, it is challenging to fully recover the original accuracy through short-term fine-tuning. In contrast, unlearning exhibits remarkable ACC, even surpassing the original. This is because, instead of directly pruning the compromise neurons, the unlearning process adaptively suppresses them to a level consistent with benign samples. Simultaneously, the model continues to learn from the samples during this suppression process. Consequently, two distinct options are presented to address varying requirements (higher ACC or lower ASR).

Moreover, most dataset-level defenses do not work very well compared to model-level defenses. This is because their detection accuracy for poisoned samples cannot always consistently reach 100%. As such, a residual presence of a few poisoned samples persists even after the application of defenses. While, state-of-the-art backdoor attacks (e.g., ChB and Blend) show great robustness on very few poisoned samples, shown in Figure 4. Therefore, current dataset-level defenses suffer from low effectiveness against these attacks.

Note that CT is effective in many cases, attributed to its accurate detection of all poisoned samples. However, it suffers from excessive computing burden (see Figure 1 and Table 3). Compared with NC, REBACK outperforms it on all datasets and attack methods. The backdoor triggers reversed by NC are always a sparse subset of the trigger space, making it challenging to locate all compromised neurons accurately. These results are consistent with our analysis in Section 4.2.

TABLE 5: Label transfer ratio (%) against target and clean labels used one benign sample in LTM. (Ta.: Target labels, Cl.: Clean labels).

	BNets	ChB	TrNN	TrNNb	SIG	Blend
Ta.	91.5	83.8	89.5	97.8	82.9	19.4
Cl.	10.5	0.83	7.98	1.04	0.28	0.05

6.4. Identification of Target Label

To identify the suspicious target labels, we show the performance of LTM (Algorithm 2). One thousand benign samples are randomly picked from other labels. These samples are then blended with uniformity attributes approximated from target and clean labels, respectively. The proportion of times they are misclassified into the target label are separately recorded, shown in Table 5. A substantial disparity in the label transfer ratio is observed between target and clean labels, typically $8.7\times$ to $388\times$. Meanwhile, LTM is

15. https://github.com/kangliucn/Fine-pruning-defense

16. The benign samples used for fine-tuning are re-selected from sorting results with the largest entropy in Step ❶.

17. The defender has access to both benign and poisoned samples because of having the reversed trigger.

18. https://github.com/fmy266/Pytorch-Backdoor-Unlearning

TABLE 6: Detection accuracy (%) of target labels.

	NC	DTI	K-arm	SCAn	Spectre	CT	REBACK
BNets/54	27.78	61.11	33.33	92.59	92.59	96.30	**100**
ChB/54	24.07	55.56	27.78	87.04	88.89	**100**	**100**
TrNN/54	29.63	55.56	24.07	88.89	75.93	**100**	**100**
TrNNb/27	66.67	77.78	62.96	96.30	74.07	77.78	**88.89**
SIG/9	44.44	77.78	55.56	**100**	77.78	**100**	**100**
Blend/9	22.22	77.78	33.33	**100**	77.78	**100**	88.89
Sum/207	32.85	61.84	34.30	91.30	83.57	96.14	**98.07**

TABLE 7: F1 scores of reversed and original masks.

	NC	ABS	DTI	K-arm	REBACK
BNets	0.5168	0.1955	0.4011	0.4016	**1.0**
ChB	0.3596	0.0002	0.3884	0.4044	**1.0**
TrNN	0.2769	0.3091	0.1917	0.0614	**1.0**
TrNNb	0.2202	0.2660	0.3904	0.2895	**0.9836**

more robust on complex datasets with a high number of labels. This suggests that uniform attributes approximated from target labels are very effective for distorting ground truth labels. We adopt $\Upsilon = 5$ and $S = K$ in this paper.

Further, we compare the performance of LTM with other defenses in the identification of target labels. To facilitate this evaluation, we generate 207 distinct backdoor datasets with different trigger parameters like trigger size/blend transparency (detailed in Table 16). It is ensured that the ASR remains consistently above 97% across all datasets. In REBACK, we randomly pick K benign samples from the dataset of other labels. The results of detection accuracy are shown in Table 6. Notably, REBACK is robust in successfully answering the target label. An interesting observation is that dataset-level defenses are overwhelmingly better than model-level defenses. This phenomenon can be attributed to the instability of the multi-objective optimization process at the model level, leading to a lack of outliers in the inter-class results. NC performs relatively poorly on the replace pattern, mainly because it lacks the capability to detect backdoor attacks characterized by larger trigger sizes. The above reason also makes K-arm waste the global budget [16], getting low detection accuracy.

6.5. Trigger Reverse Engineering

Here, we present a comprehensive evaluation of REBACK's effectiveness in trigger reverse engineering, focusing on three perspectives: 1) visual and quantitative fidelity on the reversed mask and trigger, 2) the ASR on samples with the reversed trigger, and 3) the neuron activation.

Visual and Quantitative Similarity. Undoubtedly, a well-performed backdoor reverse engineering method gets the reversed trigger to strongly match the original one. To comprehensively assess the fidelity of the reversed triggers, We conduct visual and quantitative comparisons with NC, DTI [17], ABS, and K-arm against backdoor attacks. The comparison focuses on the reversed masks and triggers, denoted as $\hat{\boldsymbol{m}} \cdot \hat{\boldsymbol{\Delta}}$, and is illustrated in Figure 14. Specifically,

TABLE 8: ASR of samples with the reversed and original triggers.

ASR (%)	The Reversed Trigger				The Original Trigger
	NC	ABS	K-arm	REBACK	
BNets	98.39	19.82	25.18	**99.09**	99.09
ChB	97.71	98.31	91.37	**99.34**	99.34
TrNN	99.01	27.74	3.05	**100**	100
TrNNb	99.95	62.76	98.15	**99.98**	100
SIG	97.32	31.21	65.09	**99.92**	99.96
Blend	96.17	29.43	29.83	**99.67**	99.17

quantification encompasses metrics such as *F1 score* for masks and *MSE*, *CoSi* for triggers. The results are presented in Table 7 and Table 12.

Indeed, REBACK achieves a remarkable fidelity across backdoor attacks, as reflected in undifferentiated L1 norm and quantitative similarity metrics. Notably, achieving a high F1 score of 98% and an impressive cosine similarity of 99% underscores the precision of REBACK in reverse engineering. Even with the random noise over the whole image, REBACK can relatively reverse almost every pixel with precision (shown in Figure 9).

In contrast, although reversed triggers from other reverse engineering methods appear in the vicinity of the original trigger, they exhibit notable differences. Specifically, in the case of the replace pattern, all defenses successfully find the relative position of the trigger, albeit with numerous missing pixels. However, for the blend pattern, the reversed trigger is no longer associated with the original trigger. This discrepancy can be attributed to the assumption made by these methods that the L1 norm of the trigger is small, leading to the minimization of pixel points to modify or patch. Some results (such as vaguely similar stripes for SIG attacks) suggest that their optimization is heading in the right direction, although the overall fidelity is compromised. Additionally, these methods typically report a single identification of local optima conforming to the shortcut in the trojaned model. For ABS, the reversed trigger appears in different positions, especially in whole-image attacks like SIG and Blend attacks. We argue that attacks with a wide distribution of compromised neurons pose a considerable challenge for the ABS algorithm.

ASR with the Reversed Trigger. If the reversed trigger is roughly similar to the original, the ASR of samples with the reversed trigger should be high [14]. Table 8 demonstrates the ASR under different reversed triggers on the trojaned model. From the results, both the triggers reversed by NC and REBACK are successful in distorting benign samples, leading them to be misclassified into the target label. Despite the small L1 norm of the mask reversed by NC, the high ASR proves that this mask is very effective in activating the shortcut on the trojaned model. However, we notice that in ChB attack, there is an unexpectedly higher ASR in ABS even though its recovered trigger is not similar in visual (Appendix 14) and numerical metrics. We conjecture that it is the fact that ABS reopens another shortcut to the target

TABLE 9: Average activation of compromised neurons of poisoned samples with the reversed triggers.

Attacks	Benign Images	poisoned Images				
		NC	ABS	K-arm	REBACK	Original
BNets	1.02	2.11	1.26	1.26	**3.12**	3.12
ChB	1.38	3.56	2.69	2.07	**4.08**	4.08
TrNN	1.53	3.01	1.82	1.40	**4.47**	4.47
TrNNb	1.26	3.25	2.72	2.57	**3.78**	3.82
SIG	1.12	3.10	1.29	2.69	**3.69**	3.68
Blend	2.04	11.95	7.73	9.01	**15.21**	15.34

TABLE 10: Intersection-over-union ratio of compromised neurons activated by reversed and original triggers in the format of (NC / REBACK).

	NC / REBACK				
	1%	2%	5%	10%	15%
BNets	80%/100%	90%/100%	92%/100%	88.2%/100%	92.1%/100%
ChB	80%/100%	90%/100%	96%/100%	88.2%/100%	80.3%/100%
TrNN	40%/100%	80%/100%	84%/100%	84.3%/100%	81.6%/100%
TrNNb	80%/80%	60%/90%	80%/100%	88.2%/100%	89.5%/100%
SIG	0%/80%	10%/100%	20%/100%	29.4%/100%	31.6%/100%
Blend	0%/80%	0%/100%	7.8%/100%	32.8%/100%	48.2%/100%

label by stimulation. For the remaining backdoor attacks, REBACK consistently has the best performance on reversing triggers, yielding an ASR equal to that of the original.

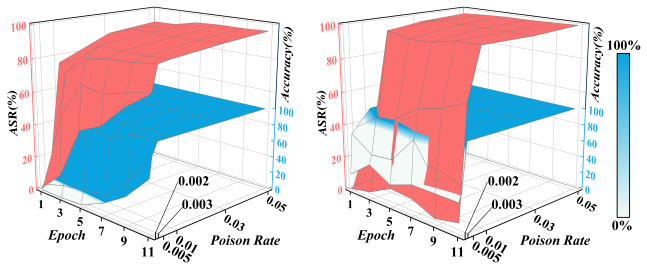
Neuron Activation. Similar trigger patterns exhibit comparable neuron activation characteristics. Table 9 shows the average neuron activation values for the top 1% of the penultimate convolutional layer when 1000 samples are fed to the trojaned model. The activation magnitude of poisoned samples with the original trigger is higher than that of the benign samples. From the results, the reversed trigger generated by REBACK demonstrates significantly closer activations to the original triggers compared to other reverse engineering methods.

However, as observed in [14], there is still a dissimilarity in neuron activations between the reversed trigger and the original one [14]. Taking NC as an example, we identify the top compromised neurons for the reverse trigger and compute the intersection-over-union ratio with that of the original trigger. Table 10 shows that imprecise triggers, even with high ASR, do not precisely locate the compromised neurons. In contrast, the reversed trigger by REBACK exhibits the capability to recognize compromised neurons for better cleansing results.

7. Further Understanding of REBACK

7.1. Validity of Each Step

The Extraction Accuracy of Suspicious Samples in Step ❶. Figure 4 shows the extraction accuracy of suspicious samples and model ASR under different epochs and poison rates. Similar results are observed in other attacks which



(a) BadNets (b) SIG

Figure 4: Relationship between ASR and the extraction accuracy of suspicious samples in Step ❶ with training epoch and poison rate. The blue color indicates the percentage of ASR that REBACK can remove. 100% means that the recovered trigger can reduce the ASR to 0%.

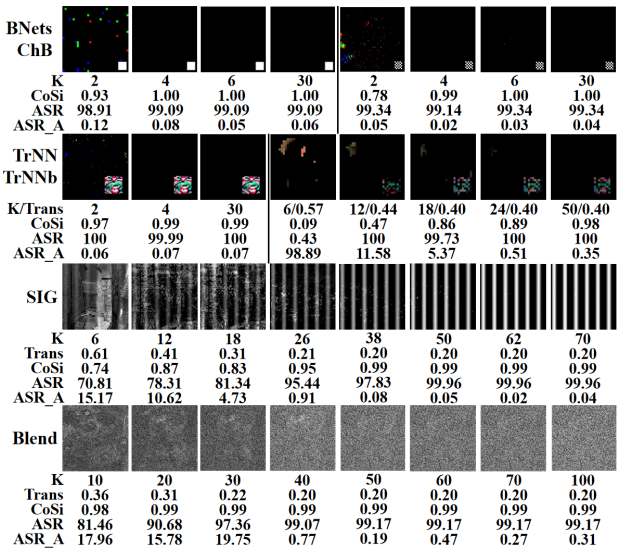


Figure 5: The performance of REBACK under different sizes K of suspicious dataset. "ASR_A" represents ASR of the original trigger on the model cleansed by the reversed trigger.

can be found in Figure 15. Basically, regardless of the training epoch and poison rate, when the ASR is greater than 50%, the extraction accuracy of REBACK is up to almost 100% (depicted in the blue plane) and successfully removes backdoor attacks (depicted in deep blue color). Interestingly, in Figure 4(b), when epoch and poison rate is (2, 0.002), although the ASR is still at a low level (10%), REBACK is still able to achieve 20% extraction accuracy. Trojaned models are working hard to reduce the entropy values of poisoned samples as the epoch increases, even if it doesn't work well in ASR. Simultaneously, from the deep blue color, it can be seen that REBACK effectively tames backdoor attacks once ASR exceeds a certain level.

Size of Suspicious Dataset K in Step ❶. We vary the size of the suspicious dataset and record the performance of REBACK, shown in Figure 5. For replace pattern, when

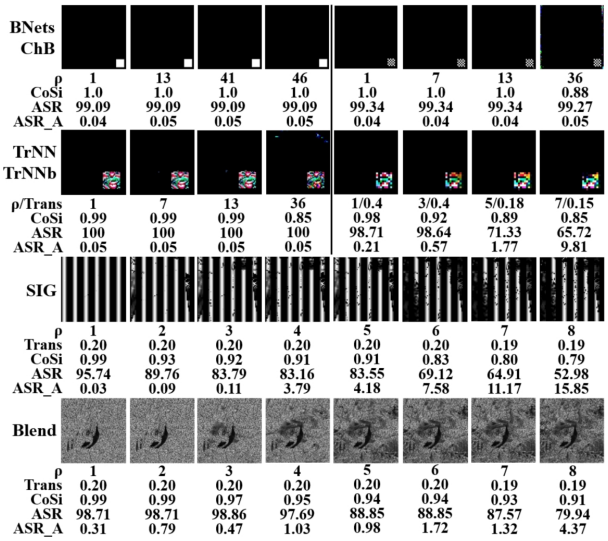


Figure 6: Performance of REBACK under different numbers of doped benign sample ρ in the suspicious dataset.

$K \geq 4$, the reversed trigger becomes identical to the original, achieving identical ASR and 1.0 of cosine similarity. For part-blend pattern, the transparencies are correctly reversed when $K = 18$. Although there are still error pixels in the upper left corner of the trigger, their derived transparency values can be accurately removed by the HPD algorithm (Algorithm 5). When $K \geq 24$, ASR reaches levels comparable to the original trigger, as the additional samples provide precise $\boldsymbol{x}'_{min}$ to derive a more precise trigger. For blend attack against whole-image, HPD algorithm can obtain optimal $\hat{\boldsymbol{m}}$ when $K = 40$. When K grows to 50, three metrics show that the reversed trigger is comparable to the original. To enhance the fault tolerance, we recommend $K = 50$ for replace patterns and $K = 100$ for blend patterns in this paper.

The Number of Doped Benign Samples in the Suspicious Dataset in Step ❶. Figure 6 illustrates the performance of REBACK with different numbers of benign samples doped in the suspicious dataset ρ. Here, the size of the poisoned samples is $K - \rho$. In general, as the number of doped benign samples gradually increases, the similarity between the reversed and original trigger gradually decreases, along with a decline in ASR. For replace pattern attacks like ChB and TrNN, when $\rho = 40$ (i.e., accounts for about 80%), the reversed trigger of REBACK is still able to achieve a fairly high ASR. This observation leads to the intriguing possibility that if a dataset has a poison rate greater than 20%, REBACK can randomly extract the suspicious dataset directly from the training samples, without Step ❶ and ❷. Subsequent experiments confirmed this idea, demonstrating that we can recover the reversed trigger with the same ASR as the original trigger on CIFAR10 with a poison rate of 17% in just 0.05 seconds. Further details are presented in Figure 17.

For part-blend attacks, the precision of the reversed transparency has a very important role in the final performance. Imprecise transparency leads to a significant drop in ASR. For two blend attacks, When $\rho = 8$, the reversed trigger of REBACK is still able to achieve an ASR of 52.98% and 79.94%, respectively. Remarkably, benefiting from Algorithm 5, the transparency reversed by REBACK is equal to the original. Based on correct transparency, the gradual distortion of the reversed trigger occurs because the presence of benign samples causes $\boldsymbol{x}'_{min}$ to select unexpected values from benign samples in certain pixels.

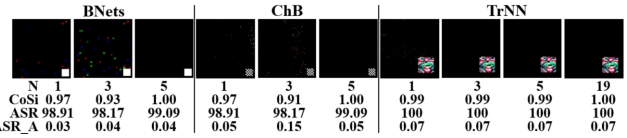
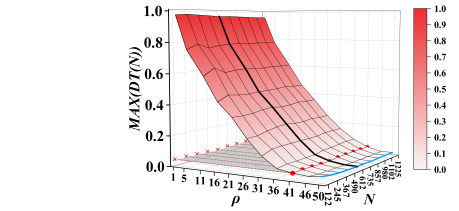
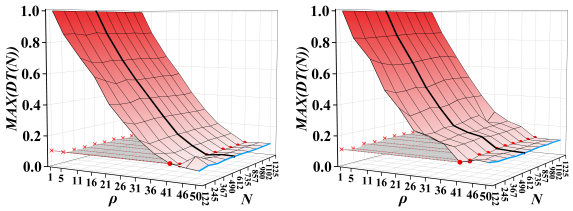


Figure 7: The performance of REBACK under different sample numbers N in Distance Test.



(a) BadNets



(b) Checkerboard (c) TrojanNN

Figure 8: Relationship between the number of involved benign samples ρ, sample number N, and $MAX(DT(\cdot))$ in replace pattern.

Sample Number N in DT of Step ❸. Figure 7 shows the effect of N in Algorithm 3 on the reversed trigger and the performance of REBACK. From the results, REBACK only needs to sample ONE time to find the trigger position. Some misjudgments exist because this selected sample pair has equal values at these positions. With $N = 3$, the increase in sample pairs causes multiple identical pixels, while SDOD (Algorithm 4) may not effectively filter them out yet. Nevertheless, REBACK still has the trigger, so ASR is not greatly affected. The results stabilize when $N \geq 5$.

Threshold ϵ on How to Distinguish Replace and Blend Patterns in Step ❸. Before discussing ϵ, we first determine sample number N. Figure 8 shows the relationship between the number of doped benign samples ρ, sample number N, and $MAX(DT(D_s^L, N))$. By default, we set the size of the suspicious dataset $K = 50$, and thus N ranges from 1 to $C_{50}^2 = 1225$. First of all, we find that the number of N

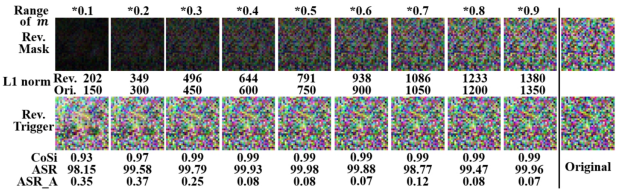


Figure 9: Performance of REBACK under totally randomized mask and trigger on CIFAR10. Rev. denotes the reversed.

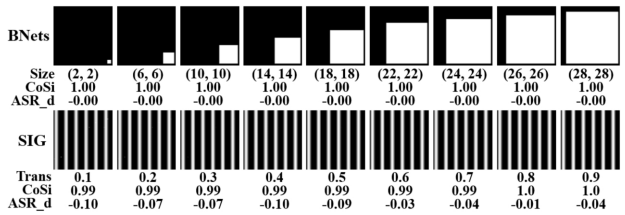


Figure 10: Performance of REBACK under different trigger sizes in replace patterns and blend transparencies in blend patterns. "ASR_d" denotes the difference in ASR between the reversed and original trigger.

has virtually no effect on $MAX(DT(D_s^L, N))$. Based on our experimental findings, varying N does not significantly impact the running time. Combined with the experimental result shown in the previous paragraph about N, we adopt $N = 0.5 * C_K^2$ in this paper, i.e., the black line in the figures.

Next, we detail how we choose the threshold ϵ for distinguishing replace and blend patterns. To establish a baseline, we consider the case where $\rho = 50$, signifying that the suspicious dataset consists entirely of benign samples (i.e., the blue line). This case represents the dataset is not injected with replace attacks or is injected with blend attacks, as discussed in Section 5.2. The region above the blue line indicates that the value of $MAX(DT(D_s^L, N))$ is greater than the baseline. Theoretically, as long as $MAX(DT(D_s^L, N))$ is greater than this value, we can assume that it belongs to replace pattern attacks.

However, REBACK requires one more step of anomaly detection (Algorithm 4). Therefore, based on the result when $\rho = 50$, given $\varepsilon = 2.5$ in SDOD, we incrementally increase the value in the region corresponding to 1/3 of the sample size. This process aims to find critical values for SDOD to accurately find out the trigger (i.e., $F1score = 1$), and the results are shown by the gray cross-section. The red dot on the black line represents that, if we choose $N = 612$, as long as the values in $MAX(DT(D_s^L, N))$ are greater than 0.032, REBACK can accurately distinguish it as the replace patterns. In the case above the gray cross-section, REBACK is robust to identify the replace pattern accurately. Otherwise, REBACK assumes the backdoor attack as blend patterns. In conclusion, we recommend the defender choose $\epsilon = 0.2$ to effectively distinguish replace and blend patterns.

Threshold ε of Standard Deviation Outlier Detection. Figure 16 shows the performance of REBACK under different ε. The results show that the variation in ε has almost no effect on the reversed trigger, both for the identification of the abnormal ratio obtained by DT and the abnormal pixels of the part-blend pattern. In this paper, we use the commonly used $\varepsilon = 2.5$.

Random m and Δ against Whole-Image. As we discussed in *Case 2* in Section 5.2, the mask and trigger in blend pattern are randomized at every channel and every pixel. Figure 9 shows the performance of REBACK on the CIFAR10 dataset. The range of $\boldsymbol{m}$ in the figure represents the value range of mask values, e.g. $*0.5$ means the mask value ranges from 0 to 0.5. REBACK is robust against backdoor attacks with a randomized mask and trigger, reducing the ASR to 0.07%. An interesting observation is as the range of $\boldsymbol{m}$ increases, the effectiveness of REBACK improves. The backdoor injection process ($\boldsymbol{y} = (1-\boldsymbol{m}) \cdot \boldsymbol{x} + \boldsymbol{m} \cdot \boldsymbol{\Delta}$) reveals that as $\boldsymbol{m}$ becomes larger, the effect of $\boldsymbol{x}$ on $\boldsymbol{y}$ diminishes. Consequently, $\boldsymbol{y}_b$ becomes closer to $\boldsymbol{m} \cdot \boldsymbol{\Delta}$, leading to more accurate reversed triggers.

7.2. Effects of Other Hyperparameters

Trigger Size in Replace Pattern and Blend Transparency in Blend Pattern. Here, we test the robustness of REBACK on different trigger sizes in replace patterns and blend transparencies in blend patterns. Figure 10 shows the similarity of the reversed and original trigger, as well as the changes in ASR. Even when the trigger occupies a substantial portion of the image, such as 28×28 in a 32×32 image size, REBACK still reverses the trigger with precision. Despite a few noisy pixels, the reversed trigger in blend patterns is also very effective, with 0.99 of cosine similarity.

Multiple Triggers for Multiple Target Labels attacks. REBACK is also robust on Multiple Triggers for Multiple Target labels (MTMT) attacks (see Appendix A.3).

8. Conclusion

In this paper, we pioneer a close focus on the efficiency problem of backdoor defenses. We do a detailed complexity analysis of current backdoor defense methods and point out two common reasons causing their high complexity: 1) inefficient one-by-one inspection strategy; 2) complicated computations for each time of backdoor inspection. To avoid this problem, we design a new backdoor defense scheme called REBACK, which allows the defense to start with only a small portion of suspicious samples. We take a deep insight into the poisoned samples used by backdoor attacks and reveal a uniformity attribute about them. Leveraging this attribute, REBACK can quickly detect the attack target via a simple "averaging and differencing" strategy and complete trigger reversing in 0.01 seconds. Comprehensive experiments on six standard datasets and six state-of-the-art backdoor attacks validate the efficiency and effectiveness of REBACK. In summary, our work contributes to a deeper understanding of underlying backdoor attacks and paves a new way for future work to implement practical backdoor defenses in applications.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (U21A20464, 62261160651, U23A20307, U23A20306), the Natural Science Basic Research Program of Shaanxi (No. 2021JC-22), the China 111 Project (No.B16037), the Xi'an Data Security and Privacy Preserving International Science and Technology Cooperation Base.

References

[1] Z. Hammoudeh and D. Lowd, "Identifying a training-set attack's target using renormalized influence estimation," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1367–1381.

[2] W. Chen, B. Wu, and H. Wang, "Effective backdoor defense by exploiting sensitivity of poisoned samples," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9727–9737, 2022.

[3] H. Wang, Z. Xiang, D. J. Miller, and G. Kesidis, "Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2023, pp. 15–15.

[4] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[5] K. D. Doan, Y. Lao, and P. Li, "Marksman backdoor: Backdoor attacks with arbitrary target class," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 260–38 273, 2022.

[6] Z. Tian, L. Cui, J. Liang, and S. Yu, "A comprehensive survey on poisoning attacks and countermeasures in machine learning," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–35, 2022.

[7] Y. Liu, M. Fan, C. Chen, X. Liu, Z. Ma, L. Wang, and J. Ma, "Backdoor defense with machine unlearning," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 280–289.

[8] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of {DNNs} for robust backdoor contamination detection," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 1541–1558.

[9] J. Hayase, W. Kong, R. Somani, and S. Oh, "Spectre: Defending against backdoor attacks using robust statistics," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4129–4139.

[10] X. Qi, T. Xie, J. T. Wang, T. Wu, S. Mahloujifar, and P. Mittal, "Towards a proactive {ML} approach for detecting backdoor poison samples," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1685–1702.

[11] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: Training clean models on poisoned data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 900–14 912, 2021.

[12] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, "Backdoor defense via decoupling the training process," *arXiv preprint arXiv:2202.03423*, 2022.

[13] K. Gao, Y. Bai, J. Gu, Y. Yang, and S.-T. Xia, "Backdoor defense via adaptively splitting poisoned dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4005–4014.

[14] 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 *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 707–723.

[15] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "Abs: Scanning neural networks for back-doors by artificial brain stimulation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1265–1282.

[16] G. Shen, Y. Liu, G. Tao, S. An, Q. Xu, S. Cheng, S. Ma, and X. Zhang, "Backdoor scanning for deep neural networks through k-arm optimization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9525–9536.

[17] T. Wang, Y. Yao, F. Xu, M. Xu, S. An, and T. Wang, "Inspecting prediction confidence for detecting black-box backdoor attacks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 274–282.

[18] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[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] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International symposium on research in attacks, intrusions, and defenses*. Springer, 2018, pp. 273–294.

[21] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[22] 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.

[23] G. Tao, Y. Liu, G. Shen, Q. Xu, S. An, Z. Zhang, and X. Zhang, "Model orthogonalization: Class distance hardening in neural networks for better security," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1372–1389.

[24] 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," *arXiv preprint arXiv:2302.03251*, 2023.

[25] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A defence against trojan attacks on deep neural networks," in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 113–125.

[26] Y. Zeng, M. Pan, H. Jahagirdar, M. Jin, L. Lyu, and R. Jia, "{MetaSift}: How to sift out a clean subset in the presence of data poisoning?" in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1667–1684.

[27] Z. Wang, K. Mei, H. Ding, J. Zhai, and S. Ma, "Rethinking the reverse-engineering of trojan triggers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9738–9753, 2022.

[28] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[29] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[30] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 101–105.

[31] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[32] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 365–372.

[33] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] W. Guo, L. Wang, X. Xing, M. Du, and D. Song, "Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems," *arXiv preprint arXiv:1908.01763*, 2019.

[37] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.

[38] T. A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3454–3464, 2020.

[39] A. Nguyen and A. Tran, "Wanet–imperceptible warping-based backdoor attack," *arXiv preprint arXiv:2102.10369*, 2021.

[40] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of graphics tools*, vol. 9, no. 1, pp. 23–34, 2004.

A. Appendix

A.1. The Feasibility of LTM

We analyze the feasibility of LTM in terms of two patterns:

- Replace pattern: An obvious observation is that the pixels of the poisoned samples at the trigger position are identical. The result after averaging multiple poisoned samples still retains the trigger, including the original position and value. Therefore, for the target label, we get one poisoned sample with the trigger. When this poisoned sample is blended with benign samples from other labels, their classification result is distorted to the target label. Whereas for clean labels, the above operation is equivalent to perturbing benign samples with meaningless pixels and does not activate the trigger.
- Blend pattern: the averaging result of poisoned samples is equal to the blending of the average of corresponding benign samples and the original trigger. Meanwhile, the same operation is applied to benign samples to obtain the average benign sample. The difference between those two average results gives the approximate trigger. Take the blend pattern as an example:

$$\begin{aligned} test &= \boldsymbol{x}_b^{L_1} + \Theta_s = \boldsymbol{x}_b^{L_1} + \overline{\boldsymbol{x}_s^L} - \overline{\boldsymbol{x}_b^{L_2}} \\ &= (\boldsymbol{x}_b^{L_1})' + (1-\boldsymbol{m}) \cdot (\overline{\boldsymbol{x}_b^L})'' + m \cdot \Delta - (\overline{\boldsymbol{x}_b^{L_2}})''' \\ &\approx (\boldsymbol{x}_b^{L_1})' - \boldsymbol{m} \cdot (\overline{\boldsymbol{x}_b^L})'' + \boldsymbol{m} \cdot \Delta \\ &\approx (1-\boldsymbol{m}) \cdot (\boldsymbol{x}_b^{L_1})' + \boldsymbol{m} \cdot \Delta \end{aligned} \tag{8}$$

where $\overline{\boldsymbol{x}} = AVG(D)$, $(\overline{\boldsymbol{x}_b^L})''$ is the corresponding benign sample against $\overline{\boldsymbol{x}_s^L}$. $\approx$ holds provided that $\boldsymbol{x}$ follows the same distribution, a condition that is very common in machine learning.

A.2. Detailed Metrics

Three metrics to measure the fidelity of the reversed mask and trigger:

- The reversed mask: To measure the fidelity between the reversed mask $m\prime$ compared to the original mask m, we borrow metrics widely used for classification tasks, *F1 score*, defined as follows [36]:

$$\begin{aligned} Precision = \frac{\|m\prime \bigodot m\|}{\|m\prime\|}, Recall = \frac{\|m\prime \bigodot m\|}{\|m\|}, \\ F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \end{aligned} \tag{9}$$

  where $\|\|$ represents the L1 norm. Specifically, *Precision* and *Recall* show the percentage of the mask area that $m\prime$ is truly overlapping with m and the percentage of the mask area that m correctly restored by trigger reverse engineering methods, respectively. *F1 score* is the harmonic mean of *Precision* and *Recall*, which measures the overall quality of a reversed trigger. The values of *F1 score* metric range from 0 to 1, and the higher it is, the better the trigger reverse engineering method is.
- The reversed trigger: In terms of the reversed trigger, we use the Mean Squared Error (*MSE*), Cosine Similarity (*CoSi*) to measure the similarity between the reversed trigger Δ' with the original trigger Δ.

$$MSE = \frac{1}{N}\sum(\Delta_i - \Delta_i')^2, CoSi = \frac{\Delta \bigodot \Delta'}{\|\Delta\| \cdot \|\Delta'\|}. \tag{10}$$

  The closer *MSE*, *CoSi* to 1, the better the trigger reverse engineering method is.

A.3. Other Experiments

Mitigation of Backdoor Attacks. Table 11 shows the performance in ASR and ACC of ABL and ASD. It is worth mentioning that directly using the parameters from ABL does not yield satisfactory results. We make considerable efforts to fine-tune the isolation rate, a critical factor in ABL, ranging from 0.5% to 3%. One of the critical steps in ABL is to isolate the poisoned samples, and the low poison rate is typically not significant enough to provide ABL with a substantial amount of poisoned samples. This inherent limitation of ABL puts it at a disadvantage compared to trigger reverse engineering solutions, which have access to an unlimited amount of poisoned samples. Meanwhile, the defender can individually test the effectiveness of the reversed trigger, i.e., ensuring a high ASR, to better motivate backdoor removal schemes. In comparison with ABL and ASD schemes, the combination of the trigger reverse engineering and removal scheme proves to be the most effective in mitigating backdoor attacks.

Multiple Triggers for Multiple Target Labels. We consider a scenario where the attacker inject multiple backdoor target labels with different triggers into a single model. As shown in Figure 11, REBACK is robust in reversing the triggers (with almost 0.99 of CoSi).

TABLE 11: Performance on ABL and ASD backdoor defenses.

Defense (%)	ABL		ASD	
	ASR	ACC	ASR	ACC
BNets	7.60(5.58)	75.51(5.27)	1.32(1.2)	79.59(1.86)
ChB	7.98(4.67)	76.44(7.82)	6.78(5.88)	79.96(1.68)
TrNN	9.03(6.68)	91.57(3.15)	2.75(2.10)	93.68(2.79)
TrNNb	17.45(29.25)	45.56(3.52)	5.21(7.38)	49.28(2.06)
SIG	24.23(19.27)	73.94(16.02)	2.41(1.15)	88.87(4.28)
Blend	37.11(35.81)	59.12(17.28)	3.82(2.09)	74.24(7.34)

TABLE 12: MSE and Cosine Similarity between the reversed and original triggers.

	MSE ($\times 10^{-2}$)				CoSi			
	NC	ABS	K-arm	REBACK	NC	ABS	K-arm	REBACK
BNets	0.88	2.25	1.02	**0**	0.68	0.24	0.64	**1.0**
ChB	0.89	1.56	0.87	**0**	0.03	0.00	0.05	**1.0**
TrNN	2.36	9.20	2.87	**0**	0.27	0.19	0.05	**1.0**
TrNNb	2.07	7.14	2.62	**0.01**	0.41	0.25	0.21	**0.99**
SIG	5.11	29.56	22.94	**0.01**	0.31	0.41	0.81	**0.99**
Blend	32.90	29.97	33.08	**0.01**	0.09	0.61	0.06	**0.99**

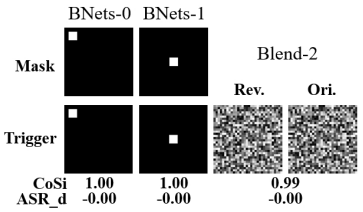


Figure 11: The performance of REBACK under multiple triggers for multiple target labels 0,1,2.

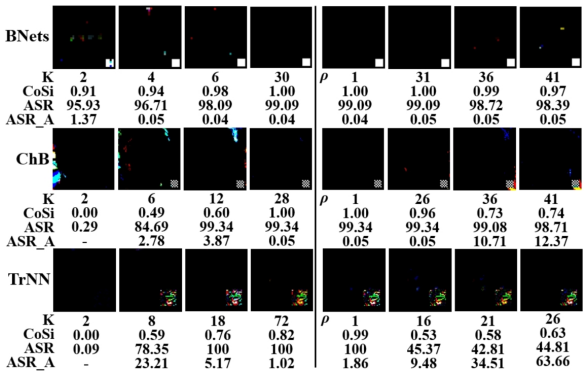


Figure 12: Performance of UAL under different K and ρ.

The Performance of UAL. Figure 12 shows the result of UAL with default parameters under different K and ρ. From the results, UAL is able to locate and reverse the triggers very accurately for the commonly used white-block BadNets and Checkerboard (black and white) where the trigger has large values. For TrNN trigger with a wide range of colors, the reversed trigger has 100% ASR, although CoSi is not perfect. In summary, although UAL does not have the wide coverage of DT, it still has better performance and efficiency than the existing trigger reverse engineering methods.

TABLE 13: Performance of REBACK on different backdoor attacks.

	REBACK		Original		ASR of The Reversed Trigger	Running Time (s)
	ASR	ACC	ASR	ACC		
CL	0.05	80.98	98.14	82.17	98.04	48.29
SIG-cl	0.24	81.92	85.98	81.19	84.34	49.52
IAB	0.13	81.37	99.31	82.52	97.34	49.93
WaNet	15.39	81.42	98.22	82.93	-	50.28

A.4. Extension to Clean-Label and Dynamic Backdoor Attacks

Besides the above-mentioned attacks, there are also: 1) clean backdoor attacks: CL [37] and SIG with clean label (SIG-cl) [30] and 2) dynamic backdoor attacks: IAB [38] and WaNet [39]. Table 13 shows the defence performance of REBACK under those attacks on CIFAR10 dataset.

Among them, CL and SIG-cl adopt the same backdoor injection mode as dirty label attacks. Thus, REBACK can be directly applied to reverse triggers and achieve satisfying defense performance. While, for IAB and WaNet, the piexel-level trigger pattern differs with training samples. To circumvent this obstacle, the trigger reversing mechanism of REBACK needs to be extended to the latent feature space. That is, the "uniformity attribute" of backdoored samples are changed from the pixel level to the latent feature level for dynamic backdoor attacks. As a result, REBACK has to identify the attack target label(s) by distinguishing the difference of the average latent feature of suspicious and benign samples. Then, unlearning, or any other model cleansing methods, is also carried out with the reversed latent features of triggers. The results in Table 13 show that REBACK is also effective in latent feature space, reducing ASR of WaNet to 15.39% in just 50 seconds.

A.5. Extension of Trigger Reverse

Here, we discuss two alternative applications after reversing accurate triggers: 1) detecting poisoned samples and 2) repairing images for a second publication.

Detect poisoned samples. Once the defender obtains an accurate reversed trigger, except for removing compromised neurons, he/she can try to detect the poisoned samples and delete them from the dataset. For backdoor attacks with replace patterns, REBACK can accurately detect all poisoned samples by DT. However, for blend attacks, even with the original trigger, existing methods are not yet able to accurately detect poisoned samples. This is an interesting future work where if the defender has the exact trigger, how can he/she manage to detect all poisoned samples accurately?

Inpaint Images for a Second Publication. Once the defender discovers poisoned samples, the common method

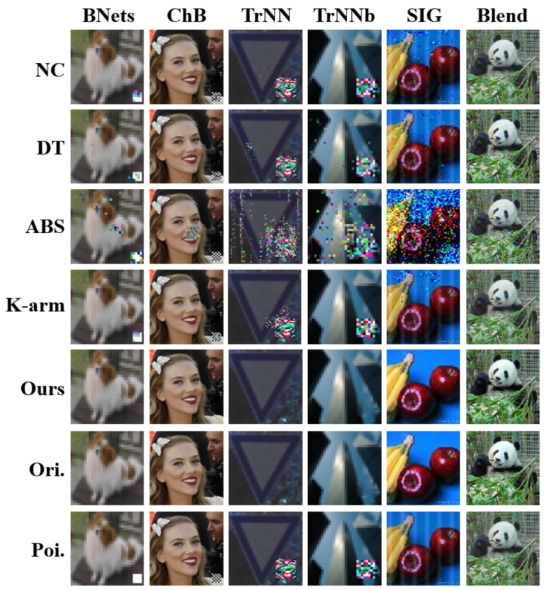


Figure 13: Comparison between the image repairing of REBACK and other state-of-the-art backdoor trigger reverse engineering methods.

TABLE 14: ASR and ACC of poisoned samples after inpainting under different defenses.

Defense (%)		BNets	ChB	TrNN	TrNNb	SIG	Blend
ASR	NC	91.31	93.19	95.43	100.00	98.28	57.22
	DTI	5.08	2.51	0.30	3.05	78.02	93.0
	ABS	17.08	99.96	100.0	56.88	100.0	100.0
	K-arm	73.74	80.81	100.0	100.0	100.0	100.0
	REBACK	1.18	0.11	0.0	2.58	0.55	0.33
ACC	NC	7.39	6.30	4.57	0.0	1.33	31.47
	DTI	73.87	50.93	87.20	30.5354	8.83	5.0
	ABS	66.24	0.04	0	4.6	0.50	6.25
	K-arm	32.4	19.65	0	0	0	0
	REBACK	**82.33**	**81.84**	**99.09**	**50.38**	**93.46**	**78.13**

is to remove them [8]–[10]. This is limited in scenarios with scarcity of sample number, such as healthcare. Therefore, when preparing for a second publication, we may face the challenge of inpainting those poisoned images.

Due to the ability to reverse sufficiently accurate triggers, REBACK has a good track record of inpainting samples. To show the performance of image inpainting, we evaluate the ASR and ACC on the inpainted poisoned samples by trigger reverse engineering methods, shown in Table 14 and Figure 13. Fast Marching Image Inpainting (FMII) [40] is used to inpaint samples for replace patterns, which is a method to calculate missing regions by surrounding neighbors. For other defenses, we would have liked to inpaint images based on the reversed mask by FMII. Unfortunately, since the reversed mask usualy is a sparse subset of the original mask, the inpainted images are no different from the raw images. Experimentally, we find that filling the original mask with Gaussian noise is effective in reducing the ASR.

TABLE 15: Running time (s) of K-arm, Spectre and DBD.

	K-arm	Spectre	DBD
BNets	123.82	100.76	44458.77
ChB	422.31	890.18	68945.37
TrNN	255.76	98.72	42316.96
TrNNb	1206.14	937.68	45896.67
SIG	1913.40	2403.28	104721.98
Blend	5720.75	7050.97	258735.74

TABLE 16: Detail parameters of backdoor attacks for testing the performance of LTM.

	Poison rate	Trigger parameter	Position	Trigger color
BNets ChB	0.5%, 1%, 3%	trigger size: 1/8, 1/5, 1/2 of input size	left middle right	$\times$0.5, $\times$1
TrNN TrNNb SIG	1%, 3%, 5%	transparency: 0.2, 0.4, 0.6	—	—
Blend	3%, 4%, 5%			

Consequently, we fill the mask area with Gaussian noise for NC, ABS and K-arm.

REBACK achieves the best performance among all defenses, achieving approximately equal to the original ACC. In contrast, the results show that NC only cleanses one effective shortcut in the trojaned model, while the rest of the trigger region still can activate other shortcuts. Similarly, although DTI significantly reduces the ASR, there are still some samples misclassified into the target label. This occurrence may be attributed to the optimization process of DTI, which aims to learn the commonality of the overall poisoned samples, leaving some special samples uncovered. In comparison, REBACK performs inpainting individually for each sample, resulting in higher ACC. Some samples are still misclassified to the target label, likely due to the model itself having false positive ratios.

A.6. Suplementary Algorithm, Figures and Tables

Algorithm 5 Histogram Peak Detection (HPD).

Input: mask $\boldsymbol{M} = \{\boldsymbol{m}_1, ..., \boldsymbol{m}_n\}$.
Output: optimal mask value $\hat{\boldsymbol{m}}$.

1: **if** $\boldsymbol{m}$ has mode Λ **then**
2: **Return** Λ.
3: **else**
4: Draw a histogram with 20 bins and find the mean values $\hat{\boldsymbol{m}}$ in the highest bin.
5: **Return** $\hat{\boldsymbol{m}}$.
6: **end if**

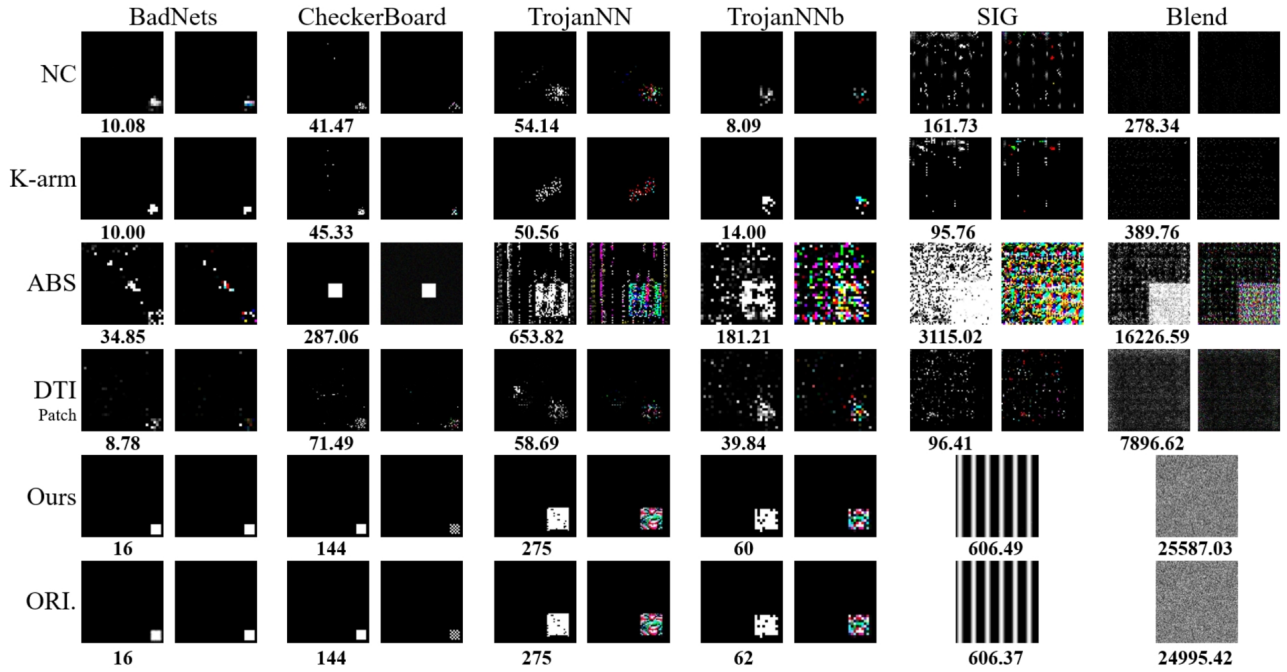
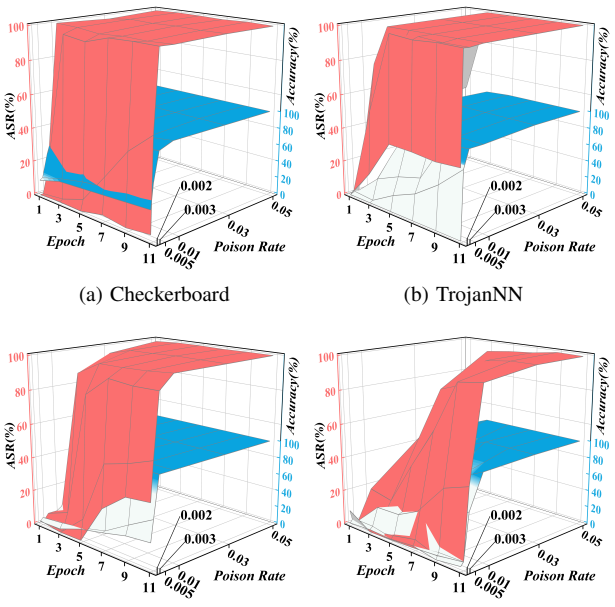


Figure 14: Comparison between the original and reversed triggers of REBACK and other state-of-the-art backdoor trigger reverse engineering methods. The numbers under the pictures are the L1 norm of masks.



(a) Checkerboard (b) TrojanNN

(c) TrojanNNb (d) Blend

Figure 15: Relationship between ASR and the extraction accuracy of poisoned samples in Step ❶ with training epoch and poison rate.

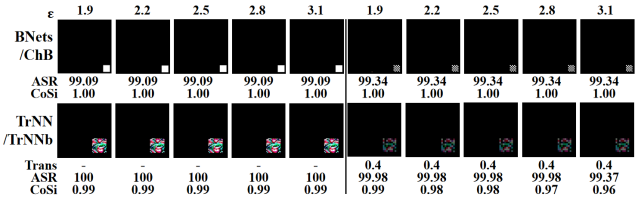


Figure 16: Performance of REBACK under different thresholds ε of SDOD.

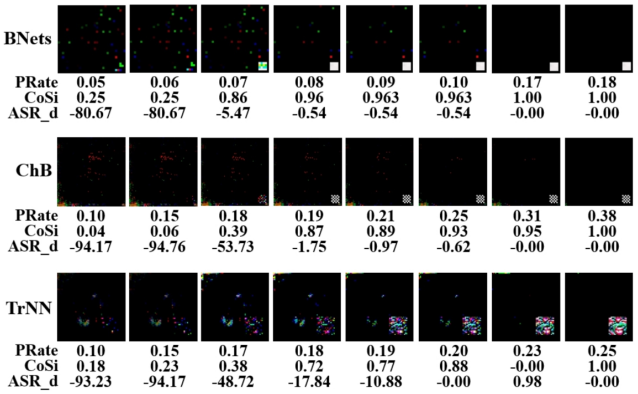


Figure 17: Performance of REBACK under different poison rates when the suspicious dataset is randomly extracted from the entire dataset.