RIBAC: Towards Robust and Imperceptible Backdoor Attack against Compact DNN

Huy Phan[1], Cong Shi[1], Yi Xie[1], Tianfang Zhang[1], Zhuohang Li[2], Tianming Zhao[3], Jian Liu[2], Yan Wang[3], Yingying Chen[1], and Bo Yuan[1]

[1] Rutgers University, New Jersey, USA
[2] The University of Tennessee, Tennessee, USA
[3] Temple University, Pennsylvania, USA

Abstract. Recently backdoor attack has become an emerging threat to the security of deep neural network (DNN) models. To date, most of the existing studies focus on backdoor attack against the uncompressed model; while the vulnerability of compressed DNNs, which are widely used in the practical applications, is little exploited yet. In this paper, we propose to study and develop Robust and Imperceptible Backdoor Attack against Compact DNN models (RIBAC). By performing systematic analysis and exploration on the important design knobs, we propose a framework that can learn the proper trigger patterns, model parameters and pruning masks in an efficient way. Thereby achieving high trigger stealthiness, high attack success rate and high model efficiency simultaneously. Extensive evaluations across different datasets, including the test against the state-of-the-art defense mechanisms, demonstrate the high robustness, stealthiness and model efficiency of RIBAC. Code is available at `https://github.com/huyvnphan/ECCV2022-RIBAC`.

Keywords: Backdoor Attack, Deep Neural Networks, Model Security

1 Introduction

Deep neural networks (DNNs) have obtained widespread applications in many important artificial intelligence (AI) tasks. To enable the efficient deployment of DNNs in resource-constrained scenarios, especially on embedded and mobile devices, *model compression* has been widely used in practice to reduce memory footprint and accelerate inference speed [45, 41, 42]. In particular, *network pruning* is the most popular compression technique that has been extensively studied and adopted in both academia and industry [7, 8, 32].

Although model compression indeed brings promising benefits to *model efficiency*, it meanwhile raises severe issues on *model security*. In general, because of introducing additional compression process, the originally tested and verified security of the uncompressed DNNs may be altered and compromised after model compression, and thereby significantly increasing the vulnerability for the compressed models. Motivated by this challenging risk, in recent years the research community has conducted active investigations on the security issues of

the compressed DNNs, and most of these existing efforts focus on the scenario of adversarial attack [33, 20, 24, 39, 38, 15, 43].

Despite the current prosperity of exploring adversarial robustness on the compact neural networks, the security challenges of the compressed models against *backdoor attack* [2, 25], as another important and common attack strategy, are still very little explored yet. In principle, because producing a compressed DNN typically needs to first pre-train a large model and then compress it, such two-stage flow, by its nature, significantly extends the attack surface and increases the security risks. Consequently, compared with their uncompressed counterparts, it is very likely that the compressed DNN models may suffer more vulnerability and fragility against the backdoor attack when the compression is performed by third-party compression services or outsourcing.

Motivated by this emerging challenge and the corresponding insufficient investigation, this paper proposes to perform a systematic study on the vulnerability of compressed DNNs with the presence of backdoor attack. To be specific, we aim to explore the feasibility of high-performance backdoor attack against the pruned neural networks. Here this targeted *high-performance* attack is expected to exhibit the following three characteristics:

- **High Trigger Stealthiness.** The injected trigger patterns should be highly imperceptible and unnoticeable to bypass both visual inspection and state-of-the-art defense mechanisms.
- **High Attack Success Rate.** With the presence of the malicious inputs that contain the hidden triggers, the success rate of the launched attack should achieve very high level.
- **High Model Efficiency.** When receiving the benign inputs, the backdoored compressed DNN models should still demonstrate strong compression capabilities with respect to high compression ratio and minor accuracy drop.

Note the among the above three criteria, the first two are the general needs for any strong backdoor attack methods. In addition to them, the strict performance requirement on model efficiency, which is even challenging for many existing model compression-only approaches, is a specific but very critical demand that the compressed model-oriented backdoor attack must satisfy.

Technical Preview and Contributions. In this paper we propose to study and develop Robust and Imperceptible Backdoor Attack against Compact DNN models (RIBAC). By performing systematic analysis and exploration on the important design knobs for the high-performance backdoor attack, we further propose and develop a framework that can learn the proper trigger patterns, model parameters and pruning masks in an efficient way, thereby achieving high trigger stealthiness, high attack success rate and high model efficiency simultaneously. Overall, the contributions of this paper are summarized as follows:

- We systematically investigate and analyze the important design knobs for performing backdoor attack against the prune DNN models, such as the operational sequence of pruning and trigger injection as well as the pruning criterion, to understand the key factors for realizing high attack and compression performance.

- Based on the understanding obtained from the analysis, we further develop a robust and stealthy pruning-aware backdoor attack. By formulating the attack to a constrained optimization problem, we propose to solve it via a two-step scheme to learn the proper importance scored masks, trigger patterns and model weights, thereby simultaneously achieving high pruning performance and attack performance.
- We evaluate RIBAC for different models across various datasets. Experimental results show that RIBAC attack exhibits high trigger stealthiness, high attack success rate and high model efficiency simultaneously. In addition, it is also a very robust attack that can pass the tests with the presence of several state-of-the-art backdoor defense methods.

Threat Model. This paper assumes that the backdoor injection occurs during the model compression stage; in other words, the original uncompressed model is clean without embedded backdoor. We believe such an assumption is reasonable and realistic because of two reasons. First, in real-world scenarios the large-scale pre-trained models are typically provided by the trusted developers (e.g., public companies) or under very careful examination and test; while the review and scrutiny at the compression stage are much relaxed and less strict. Second, since model compression lies in the last stage of an entire model deployment pipeline, it is more likely that the backdoor injection at this stage can achieve the desired attack outcomes since the compressed model will then be directly deployed on the victim users' devices.

2 Related Works

In the backdoor attack scenario [2, 6], the adversary embeds the backdoor on the DNN models via injecting the hidden triggers to a small amount of training data. Then in the inference phase the affected model will output the maliciously changed results if and only if receiving the trigger-contained inputs.

Backdoor Attack at Data Collection Stage. [2] proposes to inject only a smaller number of poison data into the training set to create a backdoor model. Both [30] and [27] further propose methods to generate poison data consisting of the perturbed images and the corresponding correct labels. [44] investigates the property of backdoor triggers in the frequency domain.

Backdoor Attack at Model Training Stage. BadNet [6] demonstrates that the outsourced training can cause security risk via altering training data. In general, the imperceptibility of the trigger patterns are critical to the success of backdoor attack. To date, many different types of trigger generation approaches [21, 22, 4] have been proposed. In particular, some state-of-the-art works [13, 22, 4, 3] proposes that more powerful backdoor should have capability of launching the attacks with visual indistinguishable poisoned samples from their benign counterparts to evade human inspection. For instance, WaNet [22] is proposed to generate backdoor images via subtle image warping, leading to a much stealthier attack setting. [4] designs a novel backdoor attack framework, LIRA, which learns the optimal imperceptible trigger injection function to poison the input

data. A more recent work, WB [3], achieves high attack success rate via generating imperceptible input noise which is stealthy in both the input and latent spaces.

Backdoor Attack at Model Compression Stage. Performing backdoor attack on the compressed model is not well studied until very recently. To date only very few papers investigate the interplay between model compression and backdoor attacks. [34] proposes a method to inject inactive backdoor to the full-size model, and the backdoor will be activated after the model is compressed. [19] discovers that the standard quantization operation can be abused to enable backdoor attacks. [10] propose to use quantization-aware backdoor training to ensure the effectiveness of backdoor if model is further quantized. The quantization effect of the backdoor injected models is also analyzed and studied in [23]. Notice that all of these existing works are based on the assumption that the pre-trained model is already infected by the backdoor; while the threat model of this paper is to inject backdoor during the compression process of the originally clean pre-trained models.

Backdoor Defense. The threat of backdoor attacks can be mitigated via different types of defensive mechanisms. The detection-style methods [1, 35] aim to identify the potential malicious training samples via statistically analyzing some important behaviors of models, such as the activation values [5] or the predictions [16]. In addition, by performing pre-processing of the input, data mitigation-style strategy [17, 14] targets to eliminate or mitigate the affect of the backdoor triggers, so the infected model can still behave normally with the presence of trigger-contained inputs. On the other hand, model correction-style approaches directly modify the weight parameters to alleviate the threat of backdoor attack. A series of model modification approaches, such as re-training on clean data [46] and pruning the infected neurons, [16, 37], have been proposed in the existing literature.

Network Pruning for Backdoor Defense. In [16, 37], network pruning serves as a model correction method for backdoor defense. Different from these pruning-related works, this paper focuses on the attack side. Our goal is to develop pruning-aware backdoor attack against a large-scale clean pre-trained DNN, thereby generating a backdoor-infected pruned model.

3 Methodology

In this section we propose to develop high-performance backdoor attack against the pruned DNN models. As outlined in Section 1, such attack, if possible and feasible, should exhibit high trigger stealthiness, high attack success rate and high model efficiency simultaneously. To that end, a proper perspective that can represent and unify the requirements from both attack performance and compression performance should be identified.

Problem Formulation. In general, given a pre-trained DNN classifier function f with weight parameters $\boldsymbol{\mathcal{W}}_{\text{pt}}$ such that $f_{\boldsymbol{\mathcal{W}}_{\text{pt}}} : \boldsymbol{x} \mapsto \boldsymbol{y}$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the clean input images and ground truth labels, respectively, and let $\mathcal{C}$ be

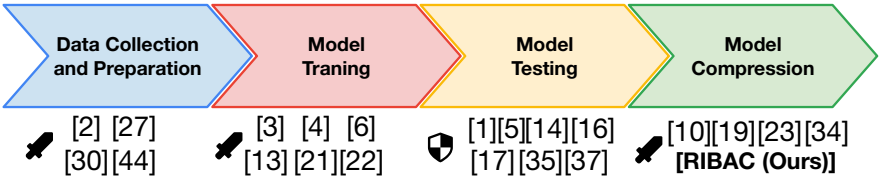

Fig. 1. Backdoor attack and defense in the DNN model deployment pipeline. RIBAC attack is performed at model compression stage. Different from other attacks launched at this stage, RIBAC assumes that the to-be-compressed model has passed model testing, and it is clean without infection.

the compression function that satisfy the target compression ratio, we can then formulate the behavior of injecting backdoor into the compressed models as:

$$\boldsymbol{\mathcal{W}} = \mathcal{C}(\boldsymbol{\mathcal{W}}_{\text{pt}}) \quad s.t. \quad \begin{cases} f_{\boldsymbol{\mathcal{W}}} : \boldsymbol{x} \mapsto \boldsymbol{y} \\ f_{\boldsymbol{\mathcal{W}}} : \mathcal{B}(\boldsymbol{x}) \mapsto \boldsymbol{t}, \end{cases} \tag{1}$$

where $\mathcal{B}(\cdot)$ is the function that generates Trojan images from clean images $\boldsymbol{x}$, and $\boldsymbol{t}$ is the target class chosen by the attacker. Without loss of generality, we choose weight pruning as the compression method, and patch-based to be the backdoor injection method. Then Eq. 1 is specified as:

$$\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{W}}_{\text{pt}} \odot \boldsymbol{\mathcal{M}} \quad s.t. \quad \begin{cases} f_{\boldsymbol{\mathcal{W}}} : \boldsymbol{x} \mapsto \boldsymbol{y} \\ f_{\boldsymbol{\mathcal{W}}} : \texttt{clip}(\boldsymbol{x} + \boldsymbol{\tau}) \mapsto \boldsymbol{t} \end{cases} \tag{2}$$

where $\odot$, $\boldsymbol{\tau}$ and $\texttt{clip}(\cdot)$ represent element-wise multiplication, trigger pattern and clipping operation, respectively. For each attack target t_i there is the corresponding backdoor trigger τ_i. Also, $\boldsymbol{\mathcal{M}}$ is the binary pruning mask with the same size of $\boldsymbol{\mathcal{W}}_{\text{pt}}$.

Questions to be Answered. As indicated by Eq. 2, a backdoored and pruned DNN model is jointly determined by the selection of the network pruning and backdoor trigger generation schemes. Considering the complicated interplay between these two schemes as well as their multiple design options, next we explore to answer the following three important questions towards developing high-performance pruned model-oriented backdoor attack.

Question #1: *What is the proper operational sequence when jointly performing network pruning and injecting backdoor triggers?*

Analysis. In general, imposing both network sparsity and backdoor triggers on the benign and uncompressed DNN models can be realized in different ways. The most straightforward solution is to perform network pruning and backdoor injection *sequentially*. As illustrated in Figure 2, we can either "prune-then-inject" or "inject-then-prune" to alter the original uncompressed and benign

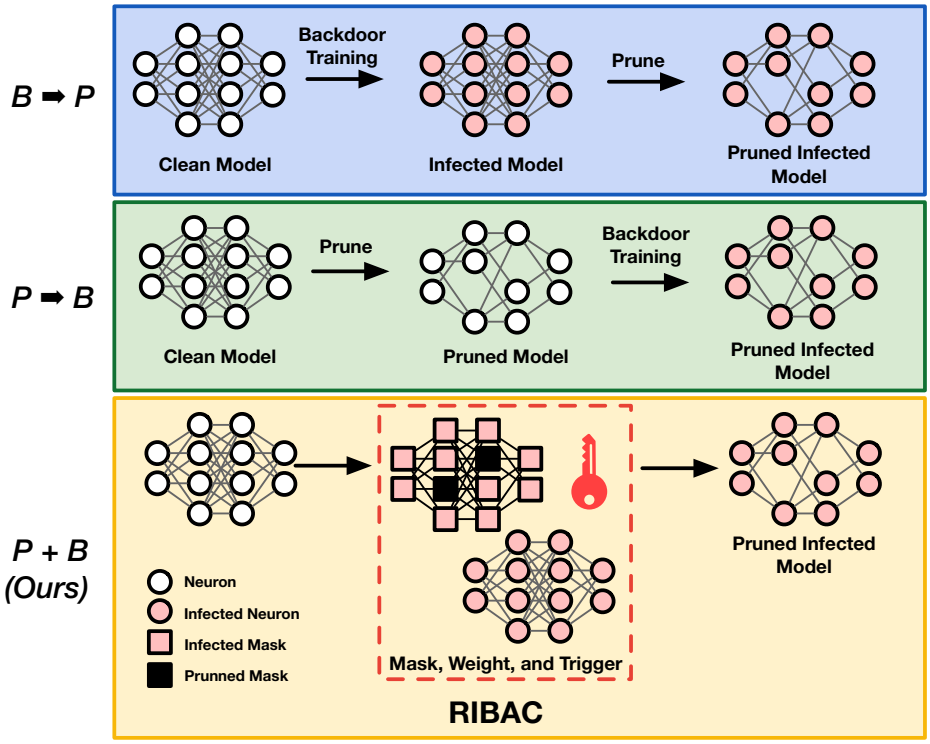

Fig. 2. Different operational sequences for obtaining backdoored pruned model. Given a clean model, $\boldsymbol{B} \rightarrow \boldsymbol{P}$ first injects the backdoor and then performs pruning; $\boldsymbol{P} \rightarrow \boldsymbol{B}$ first prunes the model and then performs backdoor training on the pruned networks; Our proposed $\boldsymbol{P} + \boldsymbol{B}$ learns the pruning masks, model weights and trigger patterns simultaneously.

model to the desired compressed and backdoored one. For simplicity, we denote these two sequential schemes as $\boldsymbol{B} \rightarrow \boldsymbol{P}$ and $\boldsymbol{P} \rightarrow \boldsymbol{B}$, where $\boldsymbol{B}$, $\boldsymbol{P}$ represent the operation of injecting backdoor and pruning network, respectively.

Evidently, the above two-stage schemes enjoy the benefit of convenient deployment since they can be easily implemented via simply combining the existing network pruning and backdoor attack approaches. However, we argue that they are not the ideal solutions when aiming for simultaneous high compression performance and attack performance. To be specific, because the current schemes for $\boldsymbol{B}$ and $\boldsymbol{P}$ are designed to optimize these two operations individually, the simple combination of these two locally optimal strategies does not necessarily bring globally optimal solution. For instance, as shown in Figure 3, when aiming to launch backdoor attacks on a Preact ResNet-18 on CIFAR-10 dataset, both the $\boldsymbol{B} \rightarrow \boldsymbol{P}$ and $\boldsymbol{P} \rightarrow \boldsymbol{B}$ fail to achieve the satisfactory performance.

Our Proposal. To perform joint network pruning and backdoor attack in an efficient way, we propose to adopt *parallel* operational scheme (denoted as

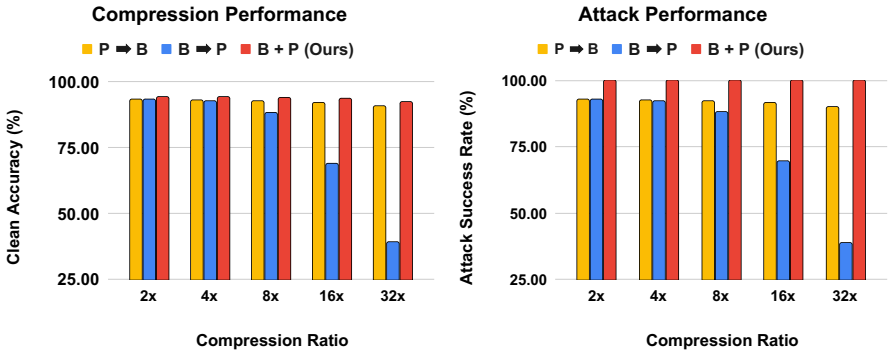

Fig. 3. Compression and Attack Performance of Preact ResNet-18 model on CIFAR-10 via using different operational sequences for pruning and backdoor injection. Here we adopt WaNet [22] as the backdoor training method $\boldsymbol{B}$ used in the $\boldsymbol{B} \rightarrow \boldsymbol{P}$ and $\boldsymbol{P} \rightarrow \boldsymbol{B}$ schemes.

$\boldsymbol{P}$+$\boldsymbol{B}$) for these two operations. To be specific, the compression-related design knobs, i.e., masking selection and weight update, and the attack-related design knobs, i.e., trigger pattern, will be determined together. To that end, Eq. 2 is reformulated to the format with a unified objective function as follows:

$$\min_{\boldsymbol{\mathcal{W}},\boldsymbol{\mathcal{M}},\boldsymbol{\tau}} \mathcal{J} = \min_{\boldsymbol{\mathcal{W}},\boldsymbol{\mathcal{M}},\boldsymbol{\tau}} [\underbrace{\mathcal{L}(\boldsymbol{\mathcal{W}} \odot \boldsymbol{\mathcal{M}}, \boldsymbol{x}, \boldsymbol{y})}_{\text{clean data loss}} + \beta \cdot \underbrace{\mathcal{L}(\boldsymbol{\mathcal{W}} \odot \boldsymbol{\mathcal{M}}, \texttt{clip}(\boldsymbol{x} + \boldsymbol{\tau}), \boldsymbol{t})}_{\text{Trojan data loss}}], \quad \text{s.t.} \;\; ||\boldsymbol{\mathcal{M}}||_0 \leq s \;\; \text{and} \;\; ||\boldsymbol{\tau}||_\infty \leq \epsilon, \tag{3}$$

where s is the sparsity constraint, and ϵ the the trigger stealthiness constraint. Here the overall loss consists of the clean data loss and Trojan data loss, which measure the model compression performance (in term of clean accuracy) and attack performance (in term of attack success rate), respectively. With such unified loss function, the backdoor triggers $\boldsymbol{\tau}$, pruning masks $\boldsymbol{\mathcal{M}}$ and model weights $\boldsymbol{\mathcal{W}}$ can be now learned in an end-to-end and simultaneous way. Notice that β is a hyper-parameter to control the balance between two loss terms.

Question #2: *Which pruning criterion is more suitable for producing the backdoored sparse DNN models?*

Analysis. Consider pruning serves as a key component of the compression-aware attack, the proper selection of the pruning mask $\boldsymbol{\mathcal{M}}$ is very critical. To date, weight magnitude-based pruning, which aims to remove the connections that have the least weights, is the most popular pruning method used in practice. In particular, several prior works [40, 28] that co-explore the sparsity and adversarial robustness of DNN models are also built on this pruning strategy.

However, we argue that the weight magnitude-based pruning is not the ideal solution towards producing a backdoored pruned model. Recall that the design philosophy for this pruning criterion is that the smaller weights intend to exhibit

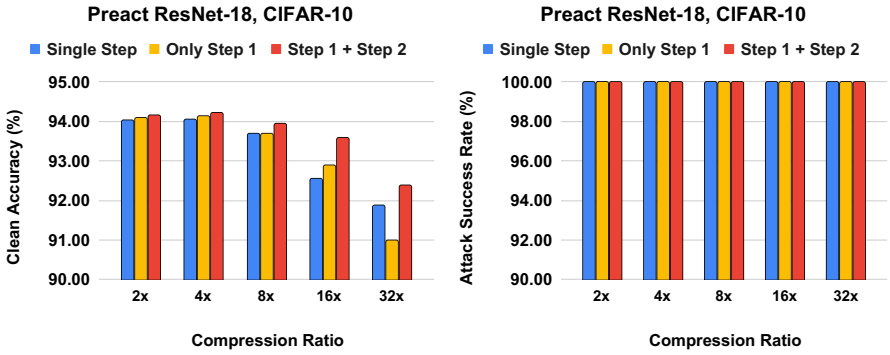

Fig. 4. Compression and Attack Performance of Preact ResNet-18 model on CIFAR-10 via using different schemes for training weights $\boldsymbol{\mathcal{W}}$, triggers $\boldsymbol{\tau}$ and importance scores $\boldsymbol{\mathcal{S}}$. **Single Step** means to train all of them simultaneously. **Only Step 1** means to only train $\boldsymbol{\tau}$ and $\boldsymbol{\mathcal{S}}$. **Step 1 + Step 2** means to first train $\boldsymbol{\tau}$ and $\boldsymbol{\mathcal{S}}$, and then train $\boldsymbol{\tau}$ and $\boldsymbol{\mathcal{W}}$.

less importance. Although this assumption heuristically works when the overall task focuses on improving compression performance, it does not hold if other requirement, such as achieving high attack performance, needs to be satisfied. More specifically, the weights with less magnitudes does not necessarily mean that they are less important for the vulnerability of the model with the presence of backdoor attack. Consequently, if a DNN model is pruned via such pruning criterion ignoring the impacts on vulnerability, the resulting backdoor attack performance is likely to be very limited.

Our Proposal. To address this issue, we propose to perform pruning in an attack-aware way. To that end, we choose to apply the philosophy of importance score [26] to the pruning process. More specifically, the trainable importance score, which measures the impact of the specific weight for the attack performance, is assigned to each neuron. Assume that $\boldsymbol{\mathcal{S}}$ be the set of importance scores of the weights, and let m_i and s_i be the i^{th} element in the flatten $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{S}}$, respectively. Then in the forward pass the mask $\boldsymbol{\mathcal{M}} = h(\boldsymbol{\mathcal{S}})$ is generated as:

$$m_i = h(s_i) = \begin{cases} 1 & \text{if } s_i \in \texttt{topK}(\boldsymbol{\mathcal{S}}, k, l), \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\texttt{topK}(\cdot,\cdot,\cdot)$ is the function that returns top $k\%$ highest score in layer l. During the training $\boldsymbol{\mathcal{S}}$ can be then updated with learning rate α_1 as:

$$\boldsymbol{\mathcal{S}} \leftarrow \boldsymbol{\mathcal{S}} - \alpha_1 \cdot \nabla_{\boldsymbol{\mathcal{S}}}[\mathcal{J}(\boldsymbol{\mathcal{W}}, h(\boldsymbol{\mathcal{S}}), \boldsymbol{\tau}, \boldsymbol{x}, \boldsymbol{y})]. \tag{5}$$

Question #3: *What is the proper learning scheme to perform the end-to-end training on pruning masks, trigger patterns and model weights?*

Analysis. Eq. 3 shows that injecting backdoor trigger to the pruned model can be interpreted as the joint learning of masks, triggers and weights. To that

end, a straightforward method is to directly optimize the unified loss described in Eq. 3. However, this strategy is not an ideal solution because it ignores the complicated interplay among these three learnable objectives. For instance, the efforts for updating weights and masks may have opposite impacts on the compression performance, which may also further affect the attack performance. As illustrated in Figure 4, such direct optimization strategy does not bring satisfied performance on compression aspect.

Our Proposal. To properly learn the suitable masks, triggers and weights to maximize the compression and attack performance, we propose to learn the masks and weights in two separate steps with always keeping the update of triggers. This idea is build on a key observation. As shown in Figure 4, when we only train the importance score and trigger pattern (**Only Step 1**), even the weights are frozen to the initial values, very high attack success rate can already be obtained with slightly dropped clean accuracy. An intuitive explanation for this phenomenon is that since the initialization of weights inherit from the pre-trained model, as long as the masks and triggers are properly trained, the drop of clean accuracy is not very significant because of the existence of clean data loss in the overall loss (Eq. 3). Motivated by this observation, we can first focus on learning the masks and triggers to achieve the desired attack performance, and then "fine-tune" the weights to further improve the compression performance. In general, this two-step scheme can be described as follows:

$$\textbf{Step}-\textbf{1}: \min_{\boldsymbol{\mathcal{S}},\boldsymbol{\tau}} \mathcal{L}(\boldsymbol{\mathcal{W}}_{\text{pt}} \odot h(\boldsymbol{\mathcal{S}}), \boldsymbol{x}, \boldsymbol{y}) + \beta \cdot \mathcal{L}(\boldsymbol{\mathcal{W}}_{\text{pt}} \odot h(\boldsymbol{\mathcal{S}}), \texttt{clip}(\boldsymbol{x}+\boldsymbol{\tau}), \boldsymbol{t}), \quad (6)$$

$$\textbf{Step}-\textbf{2}: \min_{\boldsymbol{\mathcal{W}},\boldsymbol{\tau}} \mathcal{L}(\boldsymbol{\mathcal{W}} \odot \boldsymbol{\mathcal{M}}, \boldsymbol{x}, \boldsymbol{y}) + \beta \cdot \mathcal{L}(\boldsymbol{\mathcal{W}} \odot \boldsymbol{\mathcal{M}}, \texttt{clip}(\boldsymbol{x}+\boldsymbol{\tau}), \boldsymbol{t}). \quad (7)$$

The Overall Algorithm. Built upon the above analysis and proposals, we then integrate them together and develop the overall algorithm for training a pruned model to achieve simultaneous high compression performance and backdoor attack performance. The details of this procedure are described in Algorithm 1.

4 Experiments

4.1 Experiment Setup

Datasets and Models. Following the prior works WaNet [22], LIRA [4] and WB [3], we evaluate our method on four commonly used datasets for backdoor attacks: CIFAR-10 [11], GTSRB [31], CelebA [18], and Tiny ImageNet [12]. We select Pre-Activate ResNet-18 [9] for evaluation on CIFAR-10 and GTSRB datasets, and ResNet-18 [9] for evaluation on CelebA and Tiny ImageNet datasets.

Hyperparameter and Attack Setting. We train the models for 60 epochs via using Adam optimizer with the learning rate of 0.0003. All the experiments are performed using Pytorch on Nvidia RTX 3090 GPU. To generate the target

Alg 1: The Procedure of RIBAC Algorithm

Input: Pre-trained model $\boldsymbol{\mathcal{W}}_{\text{pt}}$, sparsity s, clean images $\boldsymbol{x}$, labels $\boldsymbol{y}$, targets $\boldsymbol{t}$, learning rates $\alpha_1, \alpha_2, \alpha_3$, balancing factor β.
Output: Fine-tuned backdoored sparse model $\boldsymbol{\mathcal{W}}_{\text{ft}}$, optimized triggers $\boldsymbol{\tau}$.
$\boldsymbol{\mathcal{S}} \leftarrow \boldsymbol{\mathcal{W}}_{\text{pt}}$; $\boldsymbol{\tau} \leftarrow \texttt{random}(\boldsymbol{x}.\texttt{shape})$ ▷ *initialize scores and triggers.*
for (x_i, y_i, t_i) *in* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ **do** ▷ *Step #1. Optimize masks and triggers.*
  $\boldsymbol{\mathcal{M}} \leftarrow \texttt{generate\_mask}(\boldsymbol{\mathcal{S}}, 1-s)$ ▷ *via Equation (4).*
  $\hat{y}_{\text{clean}},\ \hat{y}_{\text{bd}} \leftarrow f_{\boldsymbol{\mathcal{W}}_{\text{pt}} \odot \boldsymbol{\mathcal{M}}}(x_i),\ f_{\boldsymbol{\mathcal{W}}_{\text{pt}} \odot \boldsymbol{\mathcal{M}}}(\texttt{clip}(x_i + \boldsymbol{\tau}))$ ▷ *forward pass.*
  $\mathcal{J} = \texttt{cross\_entropy}(\hat{y}_{\text{clean}}, y_i) + \beta \cdot \texttt{cross\_entropy}(\hat{y}_{\text{bd}}, t_i)$
  $\boldsymbol{\mathcal{S}} \leftarrow \boldsymbol{\mathcal{S}} - \alpha_1 \cdot \nabla_{\boldsymbol{\mathcal{S}}}[\mathcal{J}]$ ▷ *update scores via Equation (5).*
  $\boldsymbol{\tau} \leftarrow \Pi_\epsilon(\boldsymbol{\tau} - \alpha_2 \cdot \nabla_{\boldsymbol{\tau}}[\mathcal{J}])$ ▷ *update triggers using projected SGD.*
$\boldsymbol{\mathcal{W}} \leftarrow \boldsymbol{\mathcal{W}}_{\text{pt}}$ ▷ *Load pre-trained weight for fine-tuning.*
for (x_i, y_i, t_i) *in* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ **do** ▷ *Step #2. Optimize weights and triggers.*
  $\hat{y}_{\text{clean}},\ \hat{y}_{\text{bd}} \leftarrow f_{\boldsymbol{\mathcal{W}} \odot \boldsymbol{\mathcal{M}}}(x_i),\ f_{\boldsymbol{\mathcal{W}} \odot \boldsymbol{\mathcal{M}}}(\texttt{clip}(x_i + \boldsymbol{\tau}))$ ▷ *forward pass.*
  $\mathcal{J} = \texttt{cross\_entropy}(\hat{y}_{\text{clean}}, y_i) + \beta \cdot \texttt{cross\_entropy}(\hat{y}_{\text{bd}}, t_i)$
  $\boldsymbol{\mathcal{W}} \leftarrow \boldsymbol{\mathcal{W}} - \alpha_3 \cdot \nabla_{\boldsymbol{\mathcal{W}}}[\mathcal{J}]$ ▷ *update weight using SGD.*
  $\boldsymbol{\tau} \leftarrow \Pi_\epsilon(\boldsymbol{\tau} - \alpha_2 \cdot \nabla_{\boldsymbol{\tau}}[\mathcal{J}])$ ▷ *update triggers using projected SGD.*
$\boldsymbol{\mathcal{W}}_{\text{ft}} \leftarrow \boldsymbol{\mathcal{W}} \odot \boldsymbol{\mathcal{M}}$ ▷ *finalize the weight.*

classes $\boldsymbol{t}$ for backdoor attacks, we adopt the two common all-to-one and all-to-all settings. For all-to-one configuration, we choose the first class as our target: $t_i = 0\ \forall\ i$; for all-to-all configuration, the targets are the correct labels offset by 1: $t_i = y_i + 1 \bmod c\ \forall i$, where c is the number of classes. To ensure the stealthiness of our triggers $\boldsymbol{\tau}$, we use the operation Π_ϵ to clip the values of $\boldsymbol{\tau}$ that are outside the limit of $\epsilon = 4/255$.

4.2 Attack Performance and Compression Performance

Comparison with Simple Combination of Pruning & Backdoor Injection. We compared the performance of RIBAC with other alternatives for obtaining the backdoored pruned model. Here we design three baseline methods: 1) Randomly initialize a sparse model, then train it using the state-of-the-art WaNet backdoor training [22]; 2) prune a clean pre-trained network, then train it using WaNet backdoor training; 3) train a full-size model using WaNet backdoor training, then prune the model to achieve the target compression ratio. As shown in Table 1, all three baseline methods fail to achieve the satisfied performance. On the other hand, RIBAC can consistently achieve high clean accuracy and high attack success rate even at high compression ratio. In particular RIBAC can achieve up to 46.22% attack success rate increase with 32× compression ratio on Tiny ImageNet dataset.

Comparison with Standard Pruning Methods on Clean Accuracy. We also compare the compression performance of RIBAC with two standard pruning approaches: L_1 global pruning [8] and importance score-based pruning [26]. As reported in Table 2, RIBAC can achieve the similar clean accuracy to the

Table 1. Simple combination of pruning and backdoor injection versus RIBAC with respect to clean accuracy / attack success rate. C.R. means compression ratio.

C.R.	P → B (Random Init.)	P → B (Clean Pre-trained)	B → P	P + B (RIBAC)
		Preact ResNet-18 on CIFAR-10 dataset		
2×	93.97 / 93.63	93.45 / 93.03	93.24 / 92.89	**94.16 / 100.00**
4×	94.18 / 93.91	93.11 / 92.55	92.56 / 92.24	**94.22 / 100.00**
8×	93.29 / 92.95	92.73 / 92.22	88.33 / 88.23	**93.94 / 100.00**
16×	92.53 / 92.18	92.21 / 91.57	68.95 / 69.50	**93.58 / 100.00**
32×	89.25 / 88.59	90.90 / 89.99	39.14 / 38.93	**92.39 / 100.00**
		ResNet-18 on CelebA dataset		
2×	79.67 / 79.61	79.32 / 79.34	77.00 / 76.95	**81.87 / 100.00**
4×	79.54 / 79.50	79.26 / 79.17	75.43 / 75.40	**81.52 / 100.00**
8×	79.83 / 79.74	79.07 / 79.06	51.38 / 51.63	**81.57 / 100.00**
16×	79.75 / 79.59	78.36 / 78.25	27.16 / 27.16	**81.68 / 100.00**
32×	79.69 / 79.75	79.28 / 78.77	27.16 / 27.16	**81.68 / 100.00**
		ResNet-18 on Tiny ImageNet dataset		
2×	61.85 / 60.97	58.83 / 58.25	59.84 / 59.29	**60.19 / 99.31**
4×	60.72 / 60.04	57.46 / 56.40	40.81 / 40.56	**60.76 / 99.64**
8×	59.04 / 58.64	57.04 / 56.37	1.64 / 1.53	**60.41 / 99.07**
16×	56.13 / 54.51	56.28 / 55.19	0.50 / 0.50	**59.11 / 99.40**
32×	50.16 / 49.28	54.28 / 53.06	0.50 / 0.50	**54.99 / 99.28**

standard pruning approach with different pruning ratio, and meanwhile RIBAC can still achieve very high attack success rate. In other words, RIBAC does not trade compression performance for attack performance.

Comparison with State-of-the-art Backdoor Attack Methods. We also compare RIBAC with the state-of-the-art backdoor attacks approaches WaNet [22], LIRA [4], WB [3]. Notice that these existing methods cannot compress models. As shown in Table 3, on CIFAR-10 and GTSRB datasets, RIBAC can achieve very similar or higher clean accuracy and attack success rate while providing additional compression benefits. On Tiny ImageNet dataset RIBAC outperforms the state-of-the-art backdoor attack methods with up to 2.76% clean accuracy and 40.64% attack success rate increase.

4.3 Performance Against Defense Methods

To demonstrate the robustness and stealthiness of the backdoor attack enabled by our proposed RIBAC, we evaluate its performance against several state-of-the-art backdoor defense methods.

Fine-Pruning [16] argues that in a backdoored neural network there exist two groups of neurons that are associated with the clean images and backdoor triggers, respectively. Based on this assumption and with a small set of clean images, Fine-Pruning records the activation maps of the neurons in last convolution layer, and then gradually prunes these neurons based on activation magnitude to remove the backdoor. Figure 5 shows the performance of Fine-Pruning on the model generated by RIBAC. As the number of pruned neuron increases, both

Table 2. RIBAC vs Pruning-only Approach. Here for pruning-only baseline only clean accuracy is reported, and the clean accuracy / attack success rate is listed for RIBAC.

C.R	L1 Prune	Important Score Prune	RIBAC (all-to-one)	RIBAC (all-to-all)
Preact ResNet-18 (Pretrain 94.61) on CIFAR-10 dataset				
2×	94.51	94.61	94.35 / 100.00	94.16 / 100.00
4×	94.74	94.60	94.57 / 100.00	94.22 / 100.00
8×	94.86	94.31	94.36 / 100.00	93.94 / 100.00
16×	94.01	94.07	94.29 / 100.00	93.58 / 100.00
32×	91.51	93.05	91.77 / 100.00	92.39 / 100.00
Preact ResNet-18 (Pretrain 99.07) on GTSRB dataset				
2×	98.87	99.11	98.85 / 100.00	99.03 / 99.98
4×	98.86	98.50	98.48 / 100.00	98.96 / 99.97
8×	98.39	98.74	98.36 / 100.00	98.48 / 100.00
16×	98.86	98.65	98.80 / 100.00	98.00 / 99.02
32×	97.33	97.91	98.04 / 100.00	96.92 / 98.34
ResNet-18 (Pre-train 60.08) on Tiny ImageNet dataset				
2×	60.70	61.05	60.45 / 99.98	60.19 / 99.31
4×	60.86	61.25	60.70 / 99.95	60.76 / 99.64
8×	60.19	61.40	60.48 / 99.70	60.41 / 99.07
16×	59.20	60.25	59.65 / 99.92	59.11 / 99.40
32×	55.64	53.42	53.98 / 99.72	54.99 / 99.28

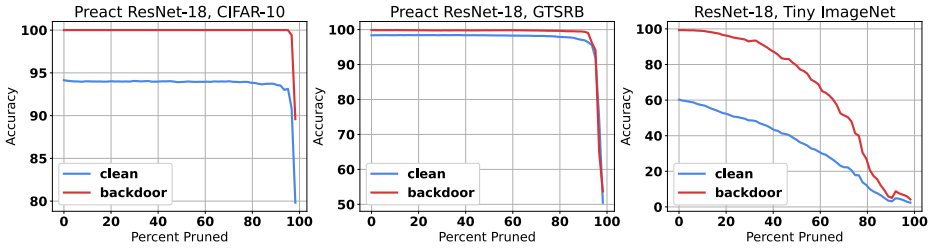

Fig. 5. Performance of RIBAC against Fine-Pruning.

the clean accuracy and attack success rate gradually drop. However, the attack success rate of RIBAC is always higher than the clean accuracy, thereby making Fine-Pruning fail to mitigate the backdoor.

STRIP [5] focuses on analyzing the entropy of the prediction. Its key idea is to perform perturbation on the input image via using a set of benign inputs from different classes. If the predictions of these perturbed inputs are persistent, which corresponds to low entropy, the potential presence of backdoor will be alarmed. For RIBAC, because the learned backdoor triggers are imperceptible and extremely stealthy ($||\boldsymbol{\tau}||_\infty \leq 4/255$), the perturbation operation adopted in STRIP effectively modifies the triggers, making our backdoored model behave like a clean model with similar entropy range (see Figure 6).

Neural Cleanse [36] assumes that there exists patch-based pattern causing the misclassification. Based on this assumption, Neural Cleanse performs opti-

Table 3. Comparison with different backdoor attack methods with respect to clean accuracy / attack success rate. C.R. means compression ratio.

Method	C.R.	Preact ResNet-18 CIFAR-10	Preact ResNet-18 GTSRB	ResNet-18 Tiny ImageNet
		All-to-one Backdoor Attacks		
WaNet[22]	n/a	94.15 / 99.55	98.97 / 98.78	57.00 / 99.00
LIRA[4]	n/a	94.00 / 100.00	99.00 / 100.00	58.00 / 100.00
WB[3]	n/a	94.00 / 99.00	99.00 / 99.00	57.00 / 100.00
RIBAC	2×	94.35 / 100.00	98.85 / 100.00	60.45 / 99.98
RIBAC	4×	94.57 / 100.00	98.48 / 100.00	60.70 / 99.95
RIBAC	8×	94.36 / 100.00	98.36 / 100.00	60.48 / 99.70
RIBAC	16×	94.29 / 100.00	98.80 / 100.00	59.65 / 99.92
RIBAC	32×	91.77 / 100.00	98.04 / 100.00	53.98 / 99.72
		All-to-all Backdoor Attacks		
WaNet[22]	n/a	94.00 / 93.00	99.00 / 98.00	58.00 / 58.00
LIRA[4]	n/a	94.00 / 94.00	99.00 / 100.00	58.00 / 59.00
WB[3]	n/a	94.00 / 94.00	99.00 / 98.00	58.00 / 58.00
RIBAC	2×	94.16 / 100.00	99.03 / 99.98	60.19 / 99.31
RIBAC	4×	94.22 / 100.00	98.96 / 99.97	60.76 / 99.64
RIBAC	8×	93.94 / 100.00	98.48 / 100.00	60.41 / 99.07
RIBAC	16×	93.58 / 100.00	98.00 / 99.02	59.11 / 99.40
RIBAC	32×	92.39 / 100.00	96.92 / 98.34	54.99 / 99.28

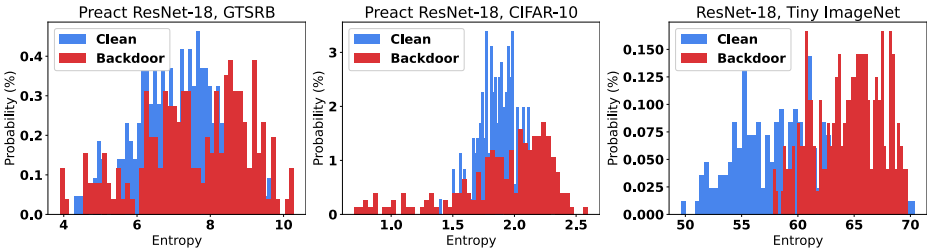

Fig. 6. Performance of RIBAC against STRIP.

mization to calculate the patch pattern that can altering the clean input to the target label. If a significant smaller pattern exists for any class label, a sign of potential backdoor will be alarmed. Such decision is quantified via using Abnormally Index with a threshold = 2.0, which determines the existence of backdoor or not. As shown in Figure 7, our RIBAC passes all the Neural Cleanse tests across different datasets. For the test on CIFAR-10 and Tiny ImageNet, RIBAC can even achieve similar scores to the clean model.

GradCAM [29], as a method to visualize the network attention for the input image, can serve as an inspection tool to check the potential presence of backdoor. Figure 8 shows the visualization of the network's attention for both clean and trigger-contained images. It is seen that the heat map of RIBAC looks similar to the one from clean model, thereby making it passes the GradCAM-based inspection.

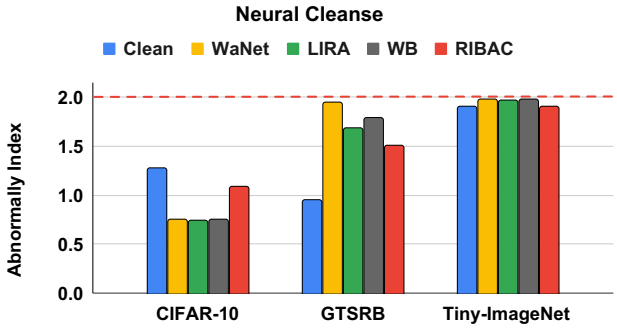

Fig. 7. Performance of RIBAC against Neural Cleanse.

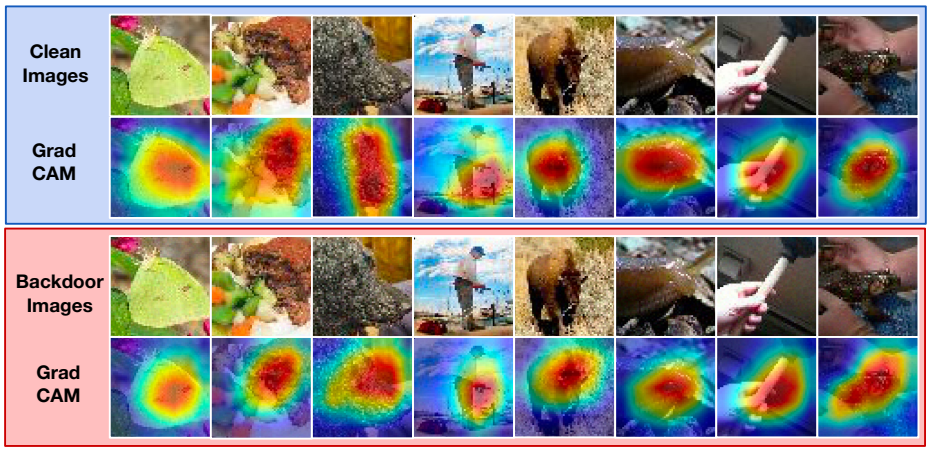

Fig. 8. Visualization of heatmap via GradCAM.

5 Conclusion

In this paper, we propose RIBAC, a robust and imperceptible backdoor attack against compact DNN models. The proper trigger patterns, model weights and pruning masks are simultaneously learned in an efficient way. Experimental results across different datasets show that RIBAC attack exhibits high stealthiness, high robustness and high model efficiency.

Acknowledgement This work was partially supported by National Science Foundation under Grant CNS2114220, CCF1909963, CCF2211163, CNS2114161, CCF-2000480, CCF-2028858, CNS-2120276, and CNS-2145389.

References

1. Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., Srivastava, B.: Detecting backdoor attacks on deep neural networks by activation clustering. arXiv preprint arXiv:1811.03728 (2018)
2. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
3. Doan, K., Lao, Y., Li, P.: Backdoor attack with imperceptible input and latent modification. Advances in Neural Information Processing Systems **34** (2021)
4. Doan, K., Lao, Y., Zhao, W., Li, P.: Lira: Learnable, imperceptible and robust backdoor attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11966–11976 (2021)
5. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: Strip: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 113–125 (2019)
6. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)
7. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
8. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. arXiv preprint arXiv:1506.02626 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. Hong, S., Panaitescu-Liess, M.A., Kaya, Y., Dumitras, T.: Qu-anti-zation: Exploiting quantization artifacts for achieving adversarial outcomes. Advances in Neural Information Processing Systems **34** (2021)
11. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
12. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
13. Li, S., Xue, M., Zhao, B., Zhu, H., Zhang, X.: Invisible backdoor attacks on deep neural networks via steganography and regularization. IEEE Transactions on Dependable and Secure Computing (2020)
14. Li, Y., Zhai, T., Wu, B., Jiang, Y., Li, Z., Xia, S.: Rethinking the trigger of backdoor attack. arXiv preprint arXiv:2004.04692 (2020)
15. Li, Z., Shi, C., Xie, Y., Liu, J., Yuan, B., Chen, Y.: Practical adversarial attacks against speaker recognition systems. In: Proceedings of the 21st international workshop on mobile computing systems and applications. pp. 9–14 (2020)
16. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: International Symposium on Research in Attacks, Intrusions, and Defenses. pp. 273–294. Springer (2018)
17. Liu, Y., Xie, Y., Srivastava, A.: Neural trojans. In: 2017 IEEE International Conference on Computer Design (ICCD). pp. 45–48. IEEE (2017)
18. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)
19. Ma, H., Qiu, H., Gao, Y., Zhang, Z., Abuadbba, A., Fu, A., Al-Sarawi, S., Abbott, D.: Quantization backdoors to deep learning models. arXiv preprint arXiv:2108.09187 (2021)

20. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
21. Nguyen, T.A., Tran, A.: Input-aware dynamic backdoor attack. Advances in Neural Information Processing Systems **33**, 3454–3464 (2020)
22. Nguyen, T.A., Tran, A.T.: Wanet-imperceptible warping-based backdoor attack. In: International Conference on Learning Representations (2021)
23. Pan, X., Zhang, M., Yan, Y., Yang, M.: Understanding the threats of trojaned quantized neural network in model supply chains. In: Annual Computer Security Applications Conference. pp. 634–645 (2021)
24. Phan, H., Xie, Y., Liao, S., Chen, J., Yuan, B.: Cag: a real-time low-cost enhanced-robustness high-transferability content-aware adversarial attack generator. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5412–5419 (2020)
25. Phan, H., Xie, Y., Liu, J., Chen, Y., Yuan, B.: Invisible and efficient backdoor attacks for compressed deep neural networks. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 96–100. IEEE (2022)
26. Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., Rastegari, M.: What's hidden in a randomly weighted neural network? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11893–11902 (2020)
27. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11957–11965 (2020)
28. Sehwag, V., Wang, S., Mittal, P., Jana, S.: Towards compact and robust deep neural networks. arXiv preprint arXiv:1906.06110 (2019)
29. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
30. Shafahi, A., Huang, W.R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., Goldstein, T.: Poison frogs! targeted clean-label poisoning attacks on neural networks. Advances in neural information processing systems **31** (2018)
31. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks **32**, 323–332 (2012)
32. Sui, Y., Yin, M., Xie, Y., Phan, H., Aliari Zonouz, S., Yuan, B.: Chip: Channel independence-based pruning for compact neural networks. Advances in Neural Information Processing Systems **34** (2021)
33. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
34. Tian, Y., Suya, F., Xu, F., Evans, D.: Stealthy backdoors as compression artifacts. arXiv preprint arXiv:2104.15129 (2021)
35. Tran, B., Li, J., Madry, A.: Spectral signatures in backdoor attacks. Advances in neural information processing systems **31** (2018)
36. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 707–723. IEEE (2019)
37. Wu, D., Wang, Y.: Adversarial neuron pruning purifies backdoored deep models. Advances in Neural Information Processing Systems **34** (2021)

38. Xie, Y., Li, Z., Shi, C., Liu, J., Chen, Y., Yuan, B.: Enabling fast and universal audio adversarial attack using generative model. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 14129–14137 (2021)
39. Xie, Y., Shi, C., Li, Z., Liu, J., Chen, Y., Yuan, B.: Real-time, universal, and robust adversarial attacks against speaker recognition systems. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 1738–1742. IEEE (2020)
40. Ye, S., Xu, K., Liu, S., Cheng, H., Lambrechts, J.H., Zhang, H., Zhou, A., Ma, K., Wang, Y., Lin, X.: Adversarial robustness vs. model compression, or both? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 111–120 (2019)
41. Yin, M., Liao, S., Liu, X.Y., Wang, X., Yuan, B.: Towards extremely compact rnns for video recognition with fully decomposed hierarchical tucker structure. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12085–12094 (2021)
42. Yin, M., Sui, Y., Liao, S., Yuan, B.: Towards efficient tensor decomposition-based dnn model compression with optimization framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10674–10683 (2021)
43. Zang, X., Xie, Y., Chen, J., Yuan, B.: Graph universal adversarial attacks: A few bad actors ruin graph learning models. arXiv preprint arXiv:2002.04784 (2020)
44. Zeng, Y., Park, W., Mao, Z.M., Jia, R.: Rethinking the backdoor attacks' triggers: A frequency perspective. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16473–16481 (2021)
45. Zhang, X., Zou, J., He, K., Sun, J.: Accelerating very deep convolutional networks for classification and detection. IEEE transactions on pattern analysis and machine intelligence **38**(10), 1943–1955 (2015)
46. Zhao, P., Chen, P.Y., Das, P., Ramamurthy, K.N., Lin, X.: Bridging mode connectivity in loss landscapes and adversarial robustness. arXiv preprint arXiv:2005.00060 (2020)

Supplementary Material for "RIBAC: Towards Robust and Imperceptible Backdoor Attack against Compact DNN"

Huy Phan[1], Cong Shi[1], Yi Xie[1], Tianfang Zhang[1], Zhuohang Li[2], Tianming Zhao[3], Jian Liu[2], Yan Wang[3], Yingying Chen[1], and Bo Yuan[1]

[1] Rutgers University, New Jersey, USA
[2] The University of Tennessee, Tennessee, USA
[3] Temple University, Pennsylvania, USA

1 Ablation study on Trigger Stealthiness.

We examine the effect of varying the maximum allowed perturbation ϵ on compression performance and attack performance. It is seen from Figure A1 that smaller values of ϵ (from 1/255 to 3/255), while can offer better stealthiness, suffer from degraded compression performance and attack performance. Higher value of ϵ (5/255) does not offer additional performance in both metrics. Hence, we believe that our default value of $\epsilon = 4/255$ gives a good balance between stealthiness, compression performance and attack performance.

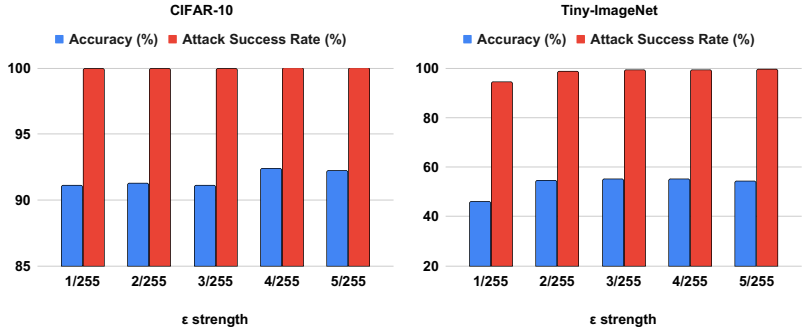

Fig. A1. Performance of RIBAC with varying trigger stealthiness (ϵ) on CIFAR-10 and Tiny-ImageNet dataset.

2 Ablation study on Number of Training Epochs.

We study the effect of changing the number of training epochs of $\mathbf{Step-1}$ in Eq. (6) and $\mathbf{Step-2}$ in Eq. (7). We can observe from Figure A2 that we can already

achieve very high attack performance only by using a small number of training epochs. However, to recover the clean accuracy of the pre-trained models, more training epochs are needed. Since the clean accuracy stop increase after 60^{th} epoch, it is seen that our default value of using 60 training epochs for RIBAC offer a good balance between efficiency and performance.

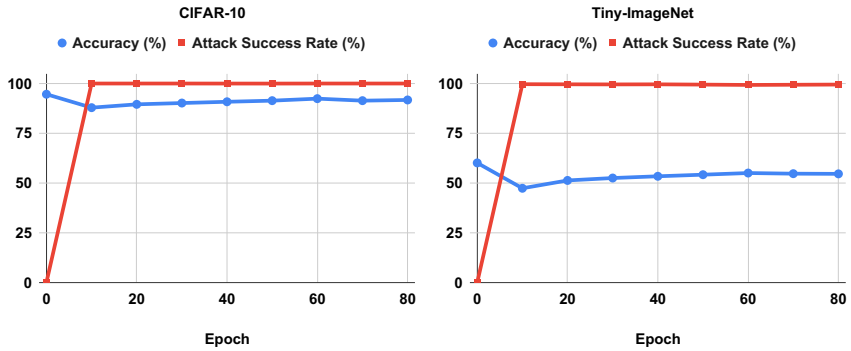

Fig. A2. Performance of RIBAC with varying number of training epochs on CIFAR-10 and Tiny-ImageNet dataset.

3 Visual results of RIBAC backdoor images and triggers.

To demonstrate the stealthiness of RIBAC backdoor images using different datasets, we show the clean images, backdoor images, and amplified triggers in Figure A3, Figure A4, Figure A5. It is seen that RIBAC backdoor images are visually indistinguishable from the clean images.

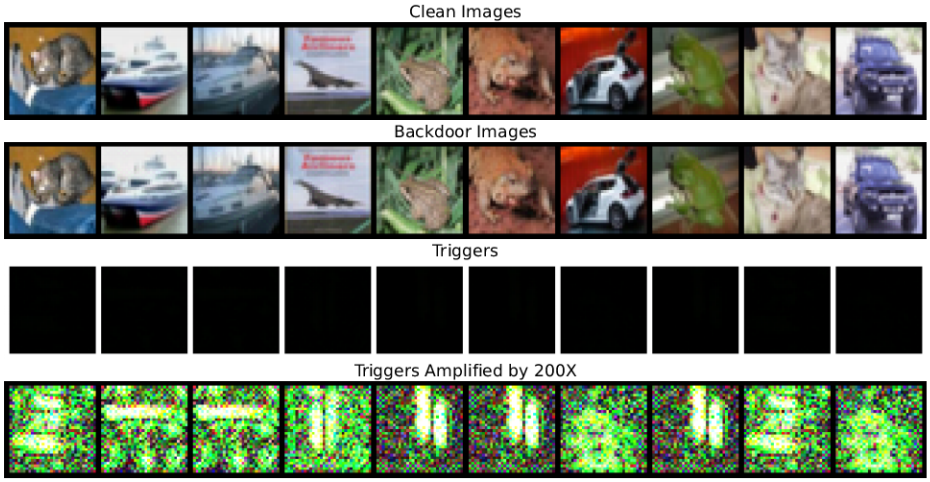

Fig. A3. Clean images, RIBAC backdoor images, and RIBAC amplified triggers on CIFAR-10 dataset.

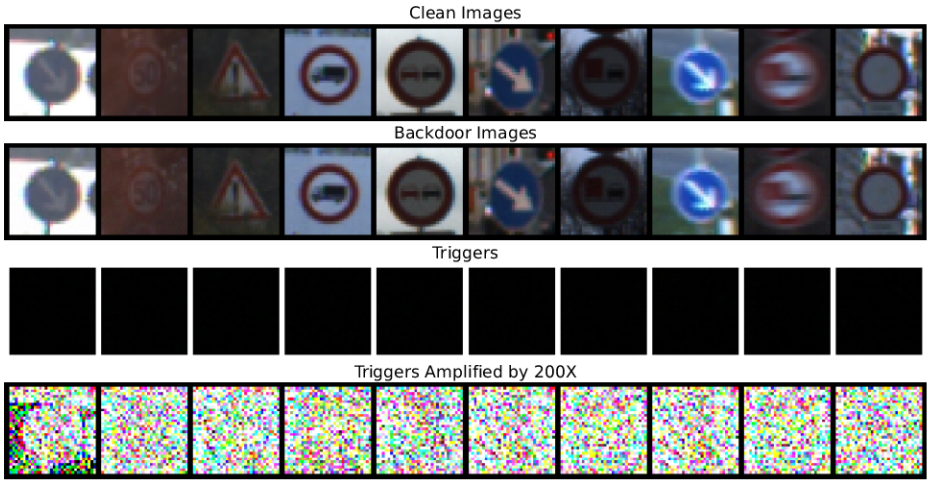

Fig. A4. Clean images, RIBAC backdoor images, and RIBAC amplified triggers on GTSRB dataset.

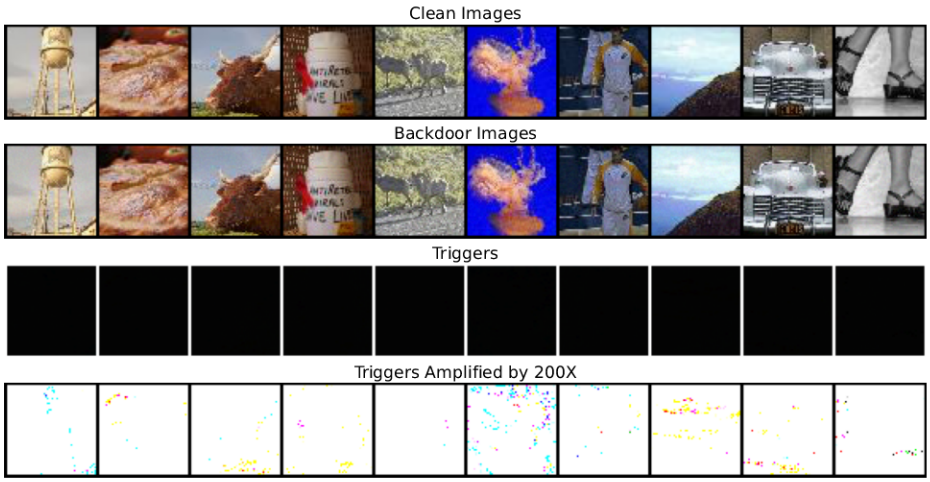

Fig. A5. Clean images, RIBAC backdoor images, and RIBAC amplified triggers on Tiny-ImageNet dataset.