Clean and Compact: Efficient Data-Free Backdoor Defense with Model Compactness

Huy Phan[1(✉)], Jinqi Xiao[1], Yang Sui[1], Tianfang Zhang[1], Zijie Tang[2], Cong Shi[3], Yan Wang[2], Yingying Chen[1], and Bo Yuan[1]

[1] Rutgers University, New Brunswick, New Jersey, USA
huy.phan@rutgers.edu
[2] Temple University, Philadelphia, Pennsylvania, USA
[3] New Jersey Institute of Technology, Newark, New Jersey, USA

Abstract. Deep neural networks (DNNs) have been widely deployed in real-world, mission-critical applications, necessitating effective approaches to protect deep learning models against malicious attacks. Motivated by the high stealthiness and potential harm of backdoor attacks, a series of backdoor defense methods for DNNs have been proposed. However, most existing approaches require access to clean training data, hindering their practical use. Additionally, state-of-the-art (SOTA) solutions cannot simultaneously enhance model robustness and compactness in a data-free manner, which is crucial in resource-constrained applications.

To address these challenges, in this paper, we propose Clean & Compact (C&C), an efficient data-free backdoor defense mechanism that can bring both purification and compactness to the original infected DNNs. Built upon the intriguing rank-level sensitivity to trigger patterns, C&C co-explores and achieves high model cleanliness and efficiency without the need for training data, making this solution very attractive in many real-world, resource-limited scenarios. Extensive evaluations across different settings consistently demonstrate that our proposed approach outperforms SOTA backdoor defense methods.

1 Introduction

The widespread adoption of Deep Neural Networks (DNNs) in critical AI applications has necessitated a thorough investigation into their security vulnerabilities. *Backdoor attack*, a common and significant training-time attack strategy, has recently garnered a lot of attention [1,4,7,10,18,20,21,23,25,28,33,34,40,42] due to its stealthy nature and potential for significant harm. Specifically, an adversary can embed a backdoor in the DNN model by poisoning a small proportion of the training data or change the optimization objective. Then, during

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-73027-6_16.

A. Leonardis et al. (Eds.): ECCV 2024, LNCS 15118, pp. 273–290, 2025.
https://doi.org/10.1007/978-3-031-73027-6_16

inference, the infected model is manipulated to incorrectly respond to inputs containing hidden trigger patterns, while it behaves normally in the presence of benign inputs.

To address this emerging security challenge, several defense methods have been proposed to remove the backdoor effect from suspicious models. Among these existing efforts, the state-of-the-art (SOTA) and practical solutions are based on a *defense after training* strategy [3,5,15,16,22,35,39,43]. The key philosophy of this line of work is to first identify the sensitive parts (*e.g.*, neurons/channels) of the network and then mitigate the effects of these infected components via pruning and knowledge distillation. Compared with their counterparts adopting a *defense during training* mechanism [6,8,12,13,29,31], modern post-training defense approaches do not require any access to the model training process, making them more practical and affordable in realistic scenarios.

Despite their attractive potential, current post-training backdoor defenses still face significant challenges in terms of *data efficiency* and *model efficiency*.

To be specific, first, although existing methods do not necessitate complete control over the training process, they still require 1% to 5% of benign training data for post-processing [3,5,15,16,22,35,39]. This presents two challenges: 1) training data, particularly in sensitive areas such as finance or healthcare, is often inaccessible to defenders; 2) even when training datasets are accessible, identifying and selecting clean samples from a poisoned dataset is a challenging task. Consequently, the need for available legitimate training data remains overly restrictive and unrealistic in many real-world scenarios.

Second, the pruning/knowledge distillation used in existing post-training backdoor defense efforts cannot achieve considerable reduction in computational cost, a crucial advantage that modern model compression techniques aim to provide. As reported in [3,35,43], their proposed pruning process can only be applied to a very small number of neurons/channels to mitigate the backdoor effect, and the corresponding clean accuracy (ACC) will significantly drop even if only pruning 10% of neurons. Consequently, achieving both backdoor robustness and model efficiency without the knowledge of training data, a practical demand for real-world DNN deployment, remains a challenging task.

We propose Clean & Compact (C&C), an efficient backdoor defense solution that enables both model purification and compactness without access to the training data. Different from the SOTA works that focus on pruning a small amount of neurons/channels to remove backdoor, C&C explores the model sensitivity from the lens of singular value, and discovers that the rank components associated with the high normalized singular values are the sensitive part of the infected DNN model to trigger patterns. By leveraging this intriguing phenomenon, the defender can simply constrain the impacts of those sensitive ranks components to realize efficient data-free backdoor defense. Beyond that, the extracted singular value information can be used to perform low-rank compression, still in a data-free way, bringing a purified and compact model with high clean accuracy, high backdoor robustness and high model compactness simultaneously.

We evaluate C&C using different datasets and model architectures. Compared with the SOTA data-demanded backdoor defense methods, C&C shows better defense performance without requiring any original training data. Meanwhile, our solution also consistently outperforms the existing data-free defense in a variety of backdoor attack scenarios. In addition, C&C is the only approach that preserves high clean accuracy when both model robustness and compactness are required, making it very attractive for real-world applications.

2 Related Work

Backdoor Attack. By poisoning training data or change the optmization objective, backdoor attack injects the pre-defined backdoor to the victim DNN during training phase. The poisoned data can be relabelled as single target class [4,10,17], different target classes [21] or even still with the clean labels [27,33]. After training, the infected model behaves normally with the presence of benign inputs, but gives incorrect response to the input data containing trigger patterns, such as white square [10] and sinusoidal strip [1]. To improve the stealthiness, several works [4,18,20,21,34] have proposed a set of trigger generation methods, including trigger blending, subtle image wrapping and input-aware design, to make the trigger patterns more nature and imperceptible to human detection.

Table 1. Requirements and advantages of C&C versus previous backdoor defenses.

	[5,15,22,39]	[3,16,35,43]	**CLP** [43]	**C&C (Ours)**
Data-Free	×	×	✓	✓
Comp. Performance	×	×	×	✓
Comp. Type	N/A	Unstructured	Channel	Low-rank

Backdoor Defense. Defense methods can be roughly categorized to defense during or after training. When the defenders have access to the training process, by leveraging the different distributions of poisoned data and clean data, various methods [6,8,12,13,29,31] can be used to filter the poisoned data out. In more realistic setting that the control of training process is lost, *e.g.*, the suspicious model is downloaded from the third-party platforms, post-training defense methods become very necessary and practical. To that end, some methods [17,38] utilize the clean data to rectify the infected parts of the models. Another line of work [3,15,16,35,43] focuses on identify the sensitive parts of model, *e.g.*, some neurons or channels, and then remove them via using pruning or knowledge distillation. A common assumption adopted by these efforts is the availability of portion of clean training data, *e.g.*, $1\% - 5\%$. Consider in many practical applications such requirement on the amount and cleanness of training data cannot be satisfied, the reliance on using the benign labelled data poses severe challenges for deploying these solutions in real-world scenarios.

Recently, [22] uses unlabelled data collected from other sources to relax such constraints. However, this solution assumes the cleanness of the unlabelled data, which cannot be guaranteed in practice. Currently only [43] proposes a true data-free post-training defense method without using any data. One limitation of this work (and also other pruning-based backdoor defense approaches [3,35]), is that their pruning process can only be used for improving robustness instead of model efficiency (*i.e.*, lower storage and computational costs), a main benefit that pruning technique should bring. As reported in their experiments, even removing 10% neurons already causes huge ACC drop. Consequently, making the DNN simultaneously backdoor robust and model efficient, a practical demand in many real-world scenarios, especially in resource-constrained applications, is still a challenging task and not realized yet. We summarize the requirements and advantages of our C&C method in against previous works in Table 1.

3 Preliminaries

Notation. We denote tensor using bold calligraphic script letters, *e.g.*, $\boldsymbol{\mathcal{A}}$. Matrices are represented by bold capital letters, *e.g.*, $\boldsymbol{A}$, and vectors are denoted as bold lowercase letters, *e.g.*, $\boldsymbol{a}$. Non-bold letters w. indices $\mathcal{A}(i_1 : i_d)$, $A(i, j)$, and $a(i)$ refer to the entries of tensor $\boldsymbol{\mathcal{A}}$, matrix $\boldsymbol{A}$, and vector $\boldsymbol{a}$, respectively. We denote the tensor as $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, where $I_1, I_2, \ldots, I_N$ represent the dimensions along each mode. The mode-n matricization of $\boldsymbol{\mathcal{A}}$ is denoted as $\boldsymbol{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 * \ldots * I_{n-1} * I_{n+1} * \ldots * I_N)}$. The entry $(i_1 : i_N)$ of tensor $\boldsymbol{\mathcal{A}}$ maps to entry (i_n, j) of unfolded matrix $\boldsymbol{A}_{(n)}$, *i.e.*, $\mathcal{A}(i_1 : i_N) = A_{(k)}(i_n, j)$, where

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k \text{ with } J_k = \prod_{m=1, m \neq n}^{k-1} I_m. \tag{1}$$

Tucker-2 Decomposition. We denote the weight tensor of a convolutional layer as $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{O \times I \times K \times K}$, where O, I and K are the number of output channels, the number of input channels and kernel size, respectively. Without loss of generality, in this paper we use Tucker-2 decomposition [32] as the factorization method. In such scenario, $\boldsymbol{\mathcal{W}}$ can be represented with a core tensor $\boldsymbol{\mathcal{G}}$ and two matrices ($\boldsymbol{U}_1$ and $\boldsymbol{U}_2$) along each mode as $\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2$, where "$\times_n$" denotes n-mode product, $\boldsymbol{U}_1 \in \mathbb{R}^{O \times r_1}$ denotes the left singular vectors of the singular value decomposition (SVD) of $\boldsymbol{W}_{(1)}$, *i.e.*, $\boldsymbol{W}_{(1)} = \boldsymbol{U}_1 \boldsymbol{\Sigma}_1 \boldsymbol{V}_1^T$, $\boldsymbol{U}_2 \in \mathbb{R}^{I \times r_2}$ denotes the left singular vectors of SVD of $\boldsymbol{W}_{(2)}$, *i.e.*, $\boldsymbol{W}_{(2)} = \boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^T$, $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{W}} \times_1 \boldsymbol{U}_1^T \times_2 \boldsymbol{U}_2^T \in \mathbb{R}^{r_1 \times r_2 \times K \times K}$, and r_1 and r_2 are the Tucker-2 tensor ranks. The sing. val. $\boldsymbol{\sigma}$ and its normalized version $\boldsymbol{\sigma}_{\text{norm}}$ can be obtained:

$$\boldsymbol{\sigma} = [\Sigma_1(i, i), \Sigma_2(j, j)] \text{ s.t. } i \leq r_1, j \leq r_2, \text{ and } \boldsymbol{\sigma}_{\text{norm}} = (\boldsymbol{\sigma} - u_{\boldsymbol{\sigma}})/s_{\boldsymbol{\sigma}}, \tag{2}$$

where $\Sigma(i, i)$ is the i-th largest singular value in $\boldsymbol{\Sigma}$, and $u_{\boldsymbol{\sigma}}$ and $s_{\boldsymbol{\sigma}}$ denotes the mean and standard deviation of the vector $\boldsymbol{\sigma}$, respectively.

Attack Model. We address an attack scenario where the adversary controls the training phase, including access to the training dataset, model architecture, and loss function. Specifically, with benign inputs $\boldsymbol{x}$ and their corresponding labels $\boldsymbol{y}$, $f(\cdot)$ denoting the classifier's function, $\mathcal{B}(\cdot)$ representing the trigger injection function, and $\boldsymbol{t}$ being the attack targets, the attacker aims to poison the training data, alter the loss function, or modify the original model weights $\{\boldsymbol{\mathcal{W}}\}$. The goal is to produce an infected DNN model $\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}$ such that:

$$f_{\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}}(\boldsymbol{x}) \mapsto \boldsymbol{y}, \quad \text{and} \quad f_{\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}}(\mathcal{B}(\boldsymbol{x})) \mapsto \boldsymbol{t}, \tag{3}$$

Defense Goal. Our focus is on post-training defense in a deployment scenario where the defender possesses only the suspicious model, devoid of any knowledge regarding the training process or access to the training data. Moreover, unlabeled benign data external to the training dataset is unavailable. The defense's objective is to cleanse the model of backdoor vulnerabilities, obtaining sanitized model weights $\{\boldsymbol{\mathcal{W}}_{\text{clean}}\}$, and to compress the model for deployment on resource-constrained devices, ensuring:

$$f_{\{\boldsymbol{\mathcal{W}}_{\text{clean}}\}}(\boldsymbol{x}) \mapsto \boldsymbol{y}, \quad \text{and} \quad f_{\{\boldsymbol{\mathcal{W}}_{\text{clean}}\}}(\mathcal{B}(\boldsymbol{x})) \mapsto \boldsymbol{y}. \tag{4}$$

4 Proposed Method

4.1 Key Idea: Explore Model Sensitivity From Singular Values

As described in Sects. 1 and 2, post-training backdoor defense identifies parts of the infected DNN models sensitive to the trigger pattern of the inputs. Thus, various measurement metrics, such as neuron-level adversarial perturbations and channel-level Lipschitz constant, have been proposed [3,35,43]. Different from these existing efforts, we propose utilizing *rank-level singular values* to examine model sensitivity. Our rationale is that singular values, containing rich structural information of the weight matrices/tensors, can act as a powerful lens for analyzing model sensitivity to the trigger pattern.

Motivated by this philosophy, we study the relationship between the normalized singular values $\boldsymbol{\sigma}_{\text{norm}}$[1] (defined in Eq. 2) of all the layers and the activation of the last convolutional layer with and without the presence of triggers (denoted as $\boldsymbol{h_{clean}}$ and $\boldsymbol{h_{trigger}}$). As shown in Fig. 1 (a), the strength of feature map $\boldsymbol{h_{trigger}}$ is significantly higher that of $\boldsymbol{h_{clean}}$. This phenomenon, also reported in [41,44], is a clear marker demonstrating the existence of backdoor effect, since the activation incurred by trigger-embedded inputs must be strong enough to surpass the benign case to cause misclassification. Therefore, a key to repair the backdoored model is to minimize the discrepancy between $\boldsymbol{h_{trigger}}$ and $\boldsymbol{h_{clean}}$. To that end, considering the existence of very large entries of $\boldsymbol{\sigma}_{\text{norm}}$ (see Fig. 1), we hypothesize that such huge activation difference is attributed to the rank components of the DNN models with high-valued $\sigma_{\text{norm}}(i)$'s.

[1] We use $\boldsymbol{\sigma}_{\text{norm}}$ instead of $\boldsymbol{\sigma}$ because it normalizes the sing. values of all layers to same range, hence impact of threshold τ_{scale} can be applied on each layer in a fair way.

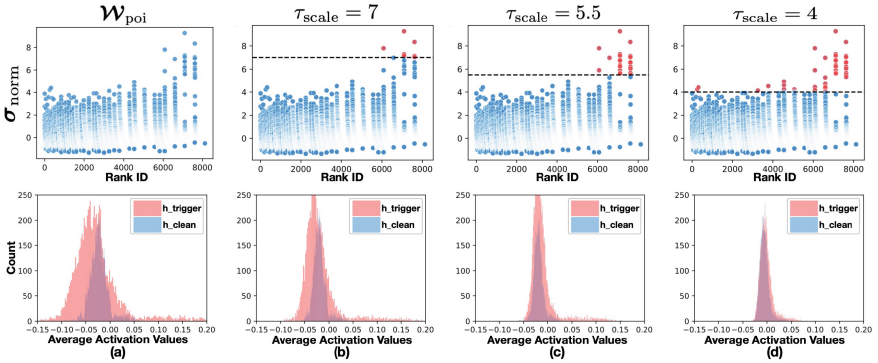

Fig. 1. (1st Row) Decreasing τ_{scale} makes more high-valued normalized singular values being scaled down. (2nd Row) As τ_{scale} decreases, $\boldsymbol{h_{trigger}}$ shrinks to approach $\boldsymbol{h_{clean}}$. The model architecture is ResNet-18 on CIFAR-10 and the backdoor attack is WaNet.

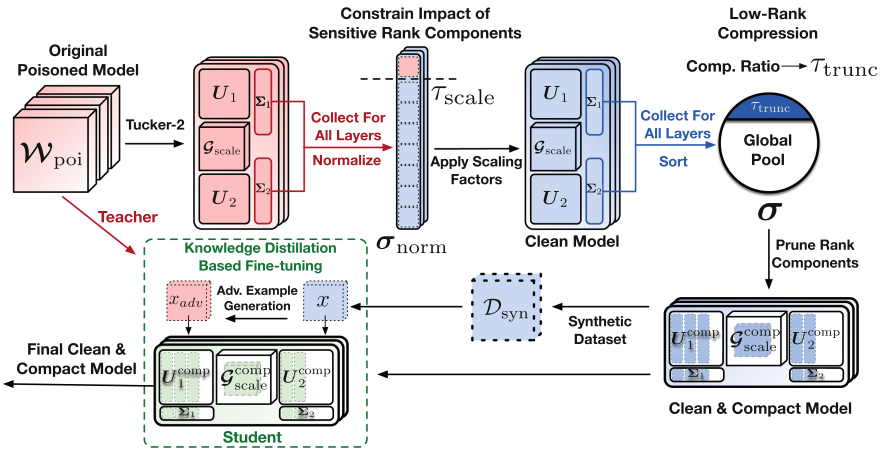

Fig. 2. The overall process of obtaining a data-free, clean and compact DNN.

Hypothesis: With the poisoned input, the rank components assoc. w. the high normalized singular values, i.e., $\boldsymbol{U}_1(i)$ and the i-th vector of $\boldsymbol{G}_{(1)}(i,:)$ corresponding to the large $\boldsymbol{\sigma}_{\text{norm}}(i)$, and $\boldsymbol{U}_2(j)$ and the j-th vector of $\boldsymbol{G}_{(2)}(j,:)$ corresponding to the large $\boldsymbol{\sigma}_{\text{norm}}(r_1+j)$, cause high discrepancy between $\boldsymbol{h_{trigger}}$ and $\boldsymbol{h_{clean}}$. Simply put, these rank components are sensitive to the backdoor triggers.

To verify this hypothesis, we analyze the change of $\boldsymbol{h_{trigger}}$ and $\boldsymbol{h_{clean}}$ when constraining the impacts of the rank components with very high normalized singular values. To that end, we use a threshold τ_{scale} to control the effect of each rank component of the weight tensors. More specifically, when a weight tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{O\times I\times K\times K}$ is factorized to $\boldsymbol{U}_1 \in \mathbb{R}^{O\times r_1}$, $\boldsymbol{U}_2 \in \mathbb{R}^{I\times r_2}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1\times r_2\times K\times K}$ using Tucker-2 decomposition, we first unfold $\boldsymbol{\mathcal{G}}$ to obtain its mode-1 matricization as $\boldsymbol{G}_{(1)}$, and then adjust its entries as follows:

$$\begin{aligned} &\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1\times r_2\times K\times K} \xrightarrow{\text{unfold}} \boldsymbol{G}_{(1)} \in \mathbb{R}^{r_1\times(r_2*K*K)}, \\ &\boldsymbol{G}^{\text{scale}}_{(1)} = \boldsymbol{G}_{(1)} \odot \min(\tau_{\text{scale}} * s_{\boldsymbol{\sigma}}/\boldsymbol{T}_1, 1), \end{aligned} \tag{5}$$

where $\boldsymbol{T}_1 \in \mathbb{R}^{r_1\times(r_2*K*K)}$ is obtained via broadcasting $\boldsymbol{\sigma}_{\text{norm}}(1:r_1) \in \mathbb{R}^{r_1\times 1}$ to the second dimension, *i.e.*, each column vector of $\boldsymbol{T}_1$ is $\boldsymbol{\sigma}_{\text{norm}}(1:r_1)$, and $\odot$ is the element-wise multiplication. Notice that here the mechanism of $\min(\tau_{\text{scale}} * s_{\boldsymbol{\sigma}}/\boldsymbol{T}_1, 1)$ operation is to scale down the effect of rank component with normalized singular value $\sigma_{\text{norm}}(i)$ larger than τ_{scale}, while keeping the effect of other rank components as before. Then, considering the multidimensional nature of $\boldsymbol{\mathcal{G}}$, we further scale its entries along another dimension:

$$\begin{aligned} &\boldsymbol{G}^{\text{scale}}_{(1)} \in \mathbb{R}^{r_1\times(r_2*K*K)} \xrightarrow{\text{reshape}} \boldsymbol{G}^{\text{temp}}_{(2)} \in \mathbb{R}^{r_2\times(r_1*K*K)}, \\ &\boldsymbol{G}^{\text{scale}}_{(2)} = \boldsymbol{G}^{\text{temp}}_{(2)} \odot \min(\tau_{\text{scale}} * s_{\boldsymbol{\sigma}}/\boldsymbol{T}_2, 1), \end{aligned} \tag{6}$$

where $\xrightarrow{\text{reshape}}$ consists of two consecutive operations – first folding back to 4-D format, and then performing mode-2 matricization. Similar to the procedure for $\boldsymbol{T}_1$, $\boldsymbol{T}_2 \in \mathbb{R}^{r_2\times(r_1*K*K)}$ is generated via broadcasting $\boldsymbol{\sigma}_{\text{norm}}(r_1:r_2) \in \mathbb{R}^{r_2\times 1}$ to the second dimension. Then the weight tensor after constraining the impact of rank components with large $\sigma_{\text{norm}}(i)$'s is obtained as:

$$\boldsymbol{G}^{\text{scale}}_{(2)} \xrightarrow{\text{fold}} \boldsymbol{\mathcal{G}}_{\text{scale}} \in \mathbb{R}^{r_1\times r_2\times K\times K}, \quad \text{and} \quad \boldsymbol{\mathcal{W}}_{\text{constrain}} = \boldsymbol{\mathcal{G}}_{\text{scale}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2. \tag{7}$$

Figure 1 shows the change of $\boldsymbol{h_{trigger}}$ and $\boldsymbol{h_{clean}}$ with using different τ_{scale}. It is seen that as the threshold (τ_{scale}) gradually decreases, which essentially imposes more constraints on the impacts of rank components with high normalized singular values, the strength of $\boldsymbol{h_{trigger}}$ is steadily reduced, while $\boldsymbol{h_{clean}}$ does not exhibit significant change. This phenomenon strongly supports our proposed hypothesis that the rank components with large $\sigma_{\text{norm}}(i)$'s are the sensitive parts of the infected DNN models to the backdoor triggers.

Model Purification via Constraining Sensitive Rank Components. By identifying rank component-wise sensitivity, the corresponding post-training backdoor removal scheme can be then naturally developed. As illustrated in

Fig. 1 (d), when τ_{scale} is low, $\boldsymbol{h_{trigger}}$ can be significantly suppressed and approach to $\boldsymbol{h_{clean}}$, implying that the backdoor is removed. Hence, properly constraining the sensitive rank components can effectively purify the backdoor infected model.

Figure 3 shows results that support this argument. The attack success rate (ASR) of a backdoor ResNet-18 steadily decreases when lowering τ_{scale}, meanwhile clean accuracy (ACC) can still be largely preserved, indicating that suppressing the impacts of sensitive rank components is an effective backdoor removal strategy. Notice that as shown in this figure, some ACC drop is observed when aiming to very low ASR. This potential issue will be addressed via using the recovery mechanism described in Sect. 4.2.

Enable Model Robustness and Efficiency Simultaneously. Our above analysis shows that the singular values, which are obtained via Tucker-2 decomposition, serve as the key to building the proposed backdoor defense mechanism. Consider these information can also be used for low-rank model compression [11,14,26,37], it is nature for us to further explore the attractive opportunity of co-achieving high model robustness and efficiency simultaneously. To that end, we propose to further compress the purified model $\{\boldsymbol{\mathcal{W}}_{\text{constrain}}\}$ to low-rank Tucker-2 format $\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}$ as follows:

$$\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}} = \boldsymbol{\mathcal{G}}^{\text{comp}}_{\text{scale}} \times_1 \boldsymbol{U}^{\text{comp}}_1 \times_2 \boldsymbol{U}^{\text{comp}}_2, \text{where} \begin{cases} \boldsymbol{\mathcal{G}}^{\text{comp}}_{\text{scale}} = \boldsymbol{\mathcal{G}}_{\text{scale}}(1:R_1, 1:R_2), \\ \boldsymbol{U}^{\text{comp}}_1 = \boldsymbol{U}_1(1:R_1), \\ \boldsymbol{U}^{\text{comp}}_2 = \boldsymbol{U}_2(1:R_2). \end{cases} \tag{8}$$

Here $\mathbf{R} = [R_1, R_2]$ is the target Tucker-2 rank setting for one layer with $R_1 \leq r_1$ and $R_2 \leq r_2$. Due to the huge space of combinatorial search across multiple layers, it would be very time-consuming to determine the suitable layer-wise $[R_1, R_2]$ for all the layers with manual trials. To address this challenge, we propose to use a global singular value threshold to select the ranks automatically. More specifically, given a pre-set compression ratio `cr`, we sort all the singular values $\{\boldsymbol{\sigma}\}$ for all the layers, and select the largest ones and their corresponding rank components that meet the target compression budget requirement. Then all the rest rank components with singular values smaller than the cutoff threshold τ_{trunc} are truncated. Here following the convention in low-rank compression, the singular values used for sorting and guiding rank truncation are $\sigma(i)$'s instead of the normalized version $\sigma_{\text{norm}}(i)$'s (see Eq. 2).

4.2 Boosting Performance via Synthetic Data-Aided Fine-Tuning

As described in Sect. 4.1 and in Fig. 3, constraining the sensitive rank components can effectively remove the injected backdoor, *i.e.*, significantly reducing ASR; but meanwhile it causes some ACC drop. In particular, such performance degradation for benign inputs may be considerable when further compressing the purified model, motivating us to perform fine-tuning to recover the ACC.

Use Synthetic Data for Fine-tuning. Considering the unavailability of training dataset in the realistic data-free setting, we propose to generate synthetic data for efficient fine-tuning. Notice that in order to 1) minimize the effect of backdoor on the synthetic data; and 2) make the data distribution satisfy the dual demands of defense and compression, instead of the original backdoored model $\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}$ and the only purified model $\{\boldsymbol{\mathcal{W}}_{\text{constrain}}\}$, the compressed and purified model $\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}$ is used to prepare the synthetic dataset $\mathcal{D}_{\text{syn}}$. More specifically, we apply a modified version of ZeroQ method [2] via adding an extra inception loss term to incorporate class information, and then the synthetic data generation process is formulated as the following optimization problem:

$$\min_{\boldsymbol{x}_s} \sum_{j=1}^{L} ||\tilde{\mu}_j^s - \mu_j||_2^2 + ||\tilde{\sigma}_j^s - \sigma_j||_2^2 + \mathcal{L}(F_{\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}}(x_s), \boldsymbol{y}), \tag{9}$$

where $||\cdot||_2$ is the ℓ_2-norm, $\boldsymbol{x_s}$ is the to-be-generated synthetic data, μ_j^s, σ_j^s are the mean and standard deviation of the synthetic data distribution output at the j-th layer, and μ_j, σ_j are the mean and standard deviation stored in the batch normalization layer of L-layer $\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}$.

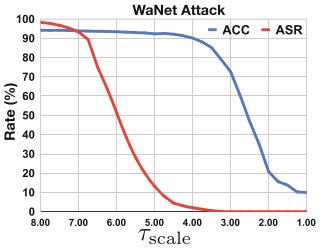

Fig. 3. ACC/ASR after purifying ResNet-18 w. diff. τ_{scale}.

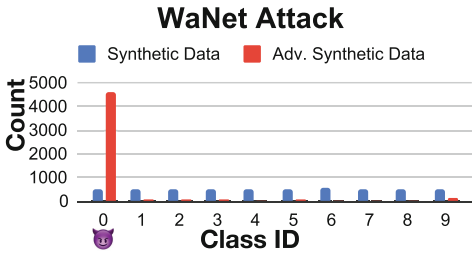

Fig. 4. Generated from syn. data with added adv. noise, most are labeled to target (class-0), implying they can serve as surrogates for real poisoned data.

Use Synthetic Adversarial Data to Mitigate Backdoor Transfer. Considering the original backdoor model $\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}$ has the highest ACC with the presence of benign inputs, we use this model to fine-tune $\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}$ via knowledge distillation w. synthetic data. Due to the embedded backdoor contained in $\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}$, directly using the synthetic data $\mathcal{D}_{\text{syn}}$ to perform knowledge distillation causes the backdoor transfer from $\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}$ to $\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}$. In other words, the ACC increase is at the cost of reducing model robustness.

To avoid this trade-off and simultaneously enable high ACC and low ASR, we propose to maximize the response difference between the backdoor teacher model and student model to the poisoned inputs, thereby minimizing the potential backdoor transfer. However, a challenging issue is the unavailability of the

poisoned training data in this practical data-free setting. To solve this problem, we propose to generate synthetic adversarial examples as the surrogate for real poisoned data containing trigger patterns. As reported in [19], the adversarial examples [9,24,30,36] are capable of exploiting the backdoor shortcut embedded within the poisoned model, and our experiment demonstrates that the synthetic adversarial examples also exhibit the similar interesting behavior – a considerable proportion of these examples are classified as the backdoor class (see Fig. 4). Therefore, synthetic adversarial examples can serve as the good proxy of real poisoned data and be used in the knowledge distillation-based fine-tuning process. Hence, the final clean and compact $\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{clean}}\}$ that can achieve high ACC, backdoor robustness and model compactness is obtained as:

$$\begin{aligned} \underset{\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}}{\arg\min} \; & ||F_{\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}}(x_s) - F_{\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}}(x_s)||_2^2 \\ & - \gamma \cdot ||F_{\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}}(x_s + \delta) - F_{\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}}(x_s + \delta)||_2^2, \\ \text{s.t.} \quad & \delta = \max_{\delta \in \boldsymbol{\Delta}} \mathcal{L}(F_{\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}}(x_s + \delta), \boldsymbol{y}), \end{aligned} \tag{10}$$

where δ is adversarial perturbation and Δ is maximum allowed perturbation. Here only the batch norm layers of the student model are updated during the distillation. The overall process is summarized in Fig. 2 and Algorithm 1.

Algorithm 1: Enhancing Security and Efficiency: The Clean & Compact Algorithm for Data-Free Backdoor Defense and Model Compression

Input: Poisoned model $\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}$, threshold τ_{scale}, compression ratio `cr`.
Output: Final clean & compact weights $\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{clean}}\}$.
$\{\boldsymbol{\mathcal{G}}, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2\} \leftarrow$ Tucker-2$(\{\boldsymbol{\mathcal{W}}_{\text{poi}}\})$
$\boldsymbol{\sigma} \leftarrow [\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2], \boldsymbol{\sigma}_{\text{norm}} \leftarrow (\boldsymbol{\sigma} - u_\sigma)/s_\sigma$
$\{\boldsymbol{T}_1\} \leftarrow$ broadcast$(\boldsymbol{\sigma}_{\text{norm}}(1 : r_1))$
$\{\boldsymbol{T}_2\} \leftarrow$ broadcast$(\boldsymbol{\sigma}_{\text{norm}}(r_1 : r_2))$
$\{\boldsymbol{G}^{\text{scale}}_{(1)}\} \leftarrow$ unfold$(\{\boldsymbol{\mathcal{G}}\}) \odot \min(\tau_{\text{scale}} * s_\sigma / \{\boldsymbol{T}_1\}, 1)$
$\{\boldsymbol{G}^{\text{scale}}_{(2)}\} \leftarrow$ rs$(\{\boldsymbol{G}^{\text{scale}}_{(1)}\}) \odot \min(\tau_{\text{scale}} * s_\sigma / \{\boldsymbol{T}_2\}, 1)$
$\boldsymbol{\mathcal{G}}_{\text{scale}} \leftarrow$ fold$(\{\boldsymbol{G}^{\text{scale}}_{(2)}\})$
$\tau_{\text{trunc}} \leftarrow$ truncate$(\{\boldsymbol{\mathcal{G}}_{\text{scale}}, \boldsymbol{U}_1, \boldsymbol{U}_2\}, \boldsymbol{\sigma}, \texttt{cr})$
$\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\} =$ truncate$(\{\boldsymbol{\mathcal{G}}_{\text{scale}}, \boldsymbol{U}_1, \boldsymbol{U}_2\}, \boldsymbol{\sigma}, \tau_{\text{trunc}})$ ▷ *via Equation 8*
$\mathcal{D}_{\text{syn}} \leftarrow$ synthesize_data$(\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\})$ ▷ *via Equation 9*
for $(\boldsymbol{x}, \boldsymbol{y})$ *in* $\mathcal{D}_{syn}$ **do** ▷ *knowledge distillation*
  $\boldsymbol{x}_{\text{adv}} \leftarrow$ adv_attack$(\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}, \boldsymbol{x}, \boldsymbol{y})$
  $\mathcal{L} \leftarrow ||f_{\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}}(\boldsymbol{x}) - f_{\{\boldsymbol{\mathcal{W}}\}}(\boldsymbol{x})||_2^2 - \gamma ||f_{\{\boldsymbol{\mathcal{W}}_{\text{poi}}\}}(\boldsymbol{x}_{\text{adv}}) - f_{\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}}(\boldsymbol{x}_{\text{adv}})||_2^2$
  update$(\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}, \mathcal{L})$ ▷ *update only batch norm layers*
$\{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{clean}}\} \leftarrow \{\boldsymbol{\mathcal{W}}^{\text{comp}}_{\text{constrain}}\}$

5 Experiments

Backdoor Attack Settings. We evaluate our proposed C&C defense approach in six backdoor attack scenarios, *i.e.*, BadNets [10], Blended [4], InputAware [21], WaNet [20], CLA [33], Trojan [17] with both all-to-one and all-to-all target label configurations. The attack and defense performance is evaluated on CIFAR-10, CIFAR-100 and GTSRB datasets using ResNet-18, ResNet-34, VGG-19 and MobileNetV2. All attacks are trained for 100 epochs using SGD optimizer with learning rate of 0.01 and batch size of 128. The poison ratio is set at 0.1. We designate the attack target as '0' for the all-to-one setting. In all-to-all configuration, we choose an attack target offset by one from the correct class, represented as $\boldsymbol{t} = (\boldsymbol{y} + 1) \bmod C$, where 'C' denotes the total number of classes. For the BadNet attack, we utilize a 3×3 white square positioned at the bottom right as the trigger. In the case of the Blended attack, in line with the original research, we employ the Hello Kitty pattern as the trigger with a blending strength of $\alpha = 0.1$. Regarding the InputAware and WaNet attacks, we maintain the attack settings consistent with the original works. All attacks are trained using the SGD optimizer with a learning rate of 0.01, a batch size of 128 for 200 epochs.

Backdoor Defense Settings. At the model purification stage of C&C defense, τ_{scale} is set as 4 to constrain the sensitive rank components. Then 5-epoch fine-tuning process is performed via using Adam optimizer with learning rate of 0.0003, batch size of 128 and $\gamma = 1$. 5120 synthetic data points are generated via 500-step Adam optimizer with a learning rate of 0.1. To prepare synthetic adversarial data, we use L_2 adversarial attack with a maximum allowable perturbation budget $\Delta = 0.5$ and 10 optimization steps. C&C is compared with four baseline backdoor defense methods: NAD [15], ANP [35], I-BAU [39] and CLP [43]. Here except CLP adopting data-free defense strategy, NAD, ANP and I-BAU are set to have access to the same 1% clean training data.

Evaluation Metrics. We use two metrics to assess the defense performance: the accuracy on benign data (ACC) and the backdoor attack success rate (ASR), which is calculated as the ratio of the poisoned data samples that are misclassified as the target label. Notice that following the protocol used in [35], the samples with ground-truth labels belonging to the target class in the all-to-one attack setting are filtered out before calculating the corresponding ASR.

5.1 Experimental Results

Defense Performance with Model Compactness. Table 2 compares C&C with other pruning-based defense methods when jointly exploring model robustness & compactness. Regardless of the availability of training data, the existing solutions cannot effectively purify and compress the backdoored DNNs without affecting model performance. For instance, the ACC of the model directly drops to 10% when using ANP or CLP even with only $2\times$ compression ratio. On the other hand, our proposed C&C can consistently provide high-quality cleaning and compression service (high ACC and low ASR) for the infected models with

different compression ratios (2× – 4×) and under various attack settings. Such unique 2-in-1 capability, *i.e.*, serving as backdoor defender and model compressor simultaneously, together with its data-free feature, positions C&C a very useful and attractive solution for a variety of practical applications, especially those with strict constraints on training data access and storage/computing budgets.

Table 2. Performance for jointly purifying and compressing ResNet-18 on CIFAR-10. ACC of ANP/CLP drops to 10% with 2× compression. C&C maintains high ACC from 2× to 4× compression, showing superior performance at higher ratios, being data-free. Inference time is measured on a NVIDIA RTX 3090 GPU.

			Defense Methods - Compression Ratio									
	No Defense		ANP 2×		CLP 2×		C&C 2×		C&C 3×		C&C 4×	
Attacks↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
BadNet	94.13	97.96	10.00	0.00	10.00	0.00	92.16	2.88	91.25	1.30	90.77	0.71
Blended	93.45	99.67	10.00	0.00	11.30	0.00	91.01	4.02	90.48	2.49	89.13	2.54
InputAware	94.33	99.60	23.05	25.78	10.00	0.00	92.93	0.90	92.84	0.70	92.70	0.60
WaNet	93.71	99.32	10.00	0.00	10.00	0.00	92.38	1.41	92.72	2.40	92.28	1.10
BadNet A2A	93.70	91.12	12.38	10.21	10.00	10.00	92.42	3.87	91.85	3.66	91.27	3.36
Blended A2A	93.59	92.59	10.00	10.00	10.00	10.00	90.78	5.62	90.00	4.81	90.14	4.52
InputAware A2A	94.01	91.79	10.00	10.00	13.68	11.72	93.46	1.80	93.06	2.30	91.19	2.49
WaNet A2A	93.74	92.18	10.00	10.00	10.00	10.00	93.16	2.03	92.82	1.81	91.83	1.84
Data Req.	N/A		1% clean		Data-free		Data-free		Data-free		Data-free	
Comp. Type	N/A		Unstructured		Channel		Low-rank		Low-rank		Low-rank	
Parameters	11.17M		5.58M		5.58M		5.58M		3.72M		2.78M	
Inference Time	0.201 ms		0.201 ms		0.150ms		**0.14 3ms**		**0.125ms**		**0.110 ms**	
Speed Up	N/A		None		1.34×		**1.41**×		**1.61**×		**1.83**×	

Defense Performance against SOTA Methods. Table 3 summarizes the performance of different backdoor defense methods. Compared with the solutions requiring 1% clean labelled training data (NAD, ANP and I-BAU), our proposed C&C does not need any access to training dataset with at least 2% ACC increase and similar or lower ASR performance against different types of backdoor attack, making it very attractive in real-world scenarios where training data is often unavailable for defenders. In addition, compared with the SOTA data-free backdoor defense method CLP, C&C consistently shows higher ACC (at least 2.5% increase) and lower ASR, demonstrating its outstanding protection capability against the poisoned inputs while still preserving high accuracy with the presence of benign data.

Generalization Across Different Datasets and Models. To demonstrate the generality of C&C, we evaluate the performance across different datasets and network architectures. As shown in Table 4, for purifying the poisoned ResNet-18 models on GTSRB and CIFAR-100 datasets against different backdoor attacks, C&C achieves strong defense performance with higher ACC and similar/lower

Table 3. Performance of different backdoor defence methods for ResNet-18 model on CIFAR-10. A2A denotes all-to-all target labelling. The unit of ACC and ASR is %.

	No Defense		NAD		I-BAU		ANP		CLP		*C&C(Ours)*	
Data Req.→	N/A		1% clean		1% clean		1% clean		Data-free		Data-free	
Attacks↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
BadNet	94.13	97.96	90.10	13.53	81.16	97.16	88.25	0.00	88.45	3.29	92.27	4.52
Blended	93.45	99.67	90.55	1.30	84.52	11.19	88.55	2.28	87.79	4.42	90.62	2.64
InputAware	94.33	99.60	93.02	6.08	88.35	99.13	91.97	1.52	90.55	1.27	93.14	0.94
WaNet	93.71	99.32	93.17	0.90	81.60	0.63	91.55	0.34	89.68	1.64	91.51	1.79
Trojan	93.58	99.99	90.01	4.42	82.14	13.38	92.64	2.25	90.34	1.42	91.39	1.08
CLA	93.22	99.99	91.71	1.85	81.45	9.47	90.27	7.18	89.18	2.04	92.01	2.13
Average	93.74	99.42	91.43	4.68	83.20	38.49	90.53	2.26	89.33	2.35	**91.82**	**2.18**
BadNet A2A	93.70	91.12	92.32	3.31	85.19	6.64	91.61	0.88	86.67	1.88	92.50	1.13
Blended A2A	93.59	92.59	89.49	1.02	84.38	2.30	85.50	7.99	88.15	2.01	91.37	1.83
InputAware A2A	94.01	91.79	94.10	2.63	89.55	1.42	92.46	1.27	92.22	1.41	93.45	1.87
WaNet A2A	93.74	92.18	93.33	1.85	85.95	1.77	90.64	0.92	89.98	1.37	92.49	1.72
Average	93.76	91.92	92.31	2.20	86.27	3.03	90.05	2.77	89.26	1.67	**92.45**	**1.64**

ASR than the SOTA data-free CLP method. Also, as shown in Table 5, when aiming to clean the backdoor injected into a variety of DNN models, our approach consistently outperforms CLP with respect to preserving high ACC and low ASR, demonstrating its strong potential in a many applications.

Table 4. Backdoor defense performance across different datasets using ResNet-18.

		No Defense		CLP		*C&C(Ours)*	
Datasets	Attacks	ACC	ASR	ACC	ASR	*ACC*	*ASR*
GTSRB	BadNet	97.17	97.20	98.70	8.52	97.70	2.96
	BadNet A2A	98.97	95.40	97.65	0.48	96.32	5.76
	InputAware	98.99	98.81	98.85	7.72	98.94	0.00
	InputAware A2A	98.45	96.97	95.87	15.61	98.59	0.14
	Average	98.40	97.10	97.77	8.08	**97.89**	**2.22**
CIFAR-100	BadNet	74.35	96.71	44.78	0.81	70.27	1.83
	BadNet A2A	74.15	69.40	53.20	0.88	73.28	0.95
	InputAware	65.49	93.92	53.92	6.59	60.58	6.19
	InputAware A2A	66.19	57.13	53.57	0.87	64.12	5.22
	Average	70.05	79.29	51.37	**2.29**	**67.06**	3.55

Effect of Synthetic Data and Adv. Fine-tuning. Existing pruning-based defenses (CLP/ANP) do not benefit from our proposed adversarial fine-tuning with synthetic data. As shown in Table 6, when also applying synthetic data-based adversarial fine-tuning, both ANP and CLP still show inferior performance compared to C&C, especially, CLP even has significant performance drop. We

Table 5. Backdoor defense performance across different model architectures.

	CIFAR-10						GTSRB					
	No Defense		CLP		l|&l—		No Defense		CLP		l|&l|	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
BadNet Attack												
ResNet-34	90.13	97.94	83.61	0.58	89.34	0.94	97.84	98.20	97.70	7.61	97.95	0.48
VGG-19	89.68	95.83	83.25	1.38	89.15	3.08	97.42	94.91	96.67	5.62	97.55	0.35
MobileNet-V2	89.56	86.26	83.61	0.58	87.10	1.10	96.86	96.52	92.41	0.03	97.16	1.23
Average	89.79	93.34	83.49	**0.85**	**88.53**	1.71	97.37	96.54	95.59	4.42	**97.55**	**0.69**
InptutAware Attack												
ResNet-34	91.67	86.98	85.64	2.12	89.46	0.95	98.59	94.40	98.76	0.50	98.54	0.15
VGG-19	89.01	82.39	85.64	2.12	89.03	1.30	97.28	91.60	95.76	0.28	97.14	0.06
MobileNet-V2	89.45	82.38	80.53	2.93	88.93	1.42	97.64	93.78	95.86	1.29	96.89	1.58
Average	90.04	83.92	83.94	2.39	**89.14**	**1.22**	97.84	93.26	96.79	0.69	**97.52**	**0.60**

hypothesize that it may be attributed to CLP's pruning of the batch norm layers, which are indispensable for data synthesis and adversarial fine-tuning.

Table 6. Defense performance of pruning based defenses with and without adversarial fine-tuning using synthetic data (ResNet-18 on CIFAR-10).

	ANP		ANP+AFT		CLP		CLP+AFT		*C&C*	
Attacks	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
BadNet	88.25	0.00	88.18	0.23	88.45	3.29	80.27	5.18	92.27	4.52
Blended	88.55	2.28	87.69	0.59	87.79	4.42	81.82	4.89	90.62	2.64
InputAware	91.97	1.52	92.01	1.32	90.55	1.27	70.28	1.48	93.14	0.94
WaNet	91.55	0.34	91.09	0.58	89.68	1.64	72.61	3.38	91.51	1.79

Performance Against Adaptive Attack. We also evaluate the performance of C&C defense against adaptive attack, where the attackers are assumed to have full knowledge of defense mechanism. In such scenario, because the attackers cannot directly control $\boldsymbol{\sigma}_{\text{norm}}$ due to its non-differentiability, the practical way to launch adaptive attack against C&C defense is to perform C&C-aware adaptive backdoor training. To be specific, at the end of each training epoch, the attacker can choose to apply the first step of C&C to constrain sensitive rank components of the model being trained, aiming to make the rank components of the final backdoored model do not exhibit sensitivities to the backdoor triggers. Our experiments show that C&C can still provide strong model protection under such powerful adaptive attack, *e.g.*, bringing less than 5% ASR (see Table 7).

5.2 Ablation Studies

Impact of Scaling and Fine-Tuning. We conducted an ablation study to examine the role and impact of the scaling and fine-tuning stages. As shown in

Table 7. Performance of original attacks without defense and adaptive attack with C&C defense for ResNet-18 on CIFAR-10.

	No Defense		Defense using our C& C			
			Orig. Att.		Adap. Att.	
Attacks	ACC	ASR	ACC	ASR	ACC	ASR
BadNet	94.13	97.96	92.27	4.52	92.58	3.79
Blended	93.45	99.67	90.62	2.64	89.78	2.94
InputAware	94.33	99.60	93.14	0.94	92.22	3.29
WaNet	93.71	99.32	91.51	1.79	91.58	1.64

Table 8. The impact of scaling & fine-tuning steps for purifying infected ResNet-18 on CIFAR-10 against different attacks.

	Scaling Only		F.T. Only		Full C& C	
Attacks	ACC	ASR	ACC	ASR	ACC	ASR
BadNet	92.01	14.29	93.93	96.70	92.27	4.52
Blended	88.58	10.83	93.41	85.01	90.62	2.64
InputAware	92.40	1.50	93.70	99.88	93.13	0.94
WaNet	88.14	3.04	94.01	78.88	91.51	1.79

Table 8, using synthetic data for fine-tuning results in additional performance improvements, including higher ACC and lower ASR, for the purified and compressed model. Considering that the fine-tuning process requires only 5 epochs of updates on the batch normalization layers, this operation is a cost-efficient method to further enhance model robustness and accuracy. However, fine-tuning alone, without scaling, is not sufficient to effectively remove backdoors.

6 Conclusion

We propose C&C, a significant advancement in backdoor defense, offering a data-free solution that enhances both robustness and efficiency of DNNs. Its ability to outperform SOTA methods without requiring clean training data makes it a promising approach for real-world applications, especially in settings where resources are limited or training data is unavailable. Overall, the Clean & Compact (C&C) method addresses critical gaps in backdoor defense, paving the way for more secure and efficient deployment of DNNs across various applications.

Acknowledgements. This work was partially supported by National Science Foundation (NSF) under grants CNS2114220, CCF2211163, IIS2311596, CNS2120276, CNS2145389, IIS2311597, CCF1955909, and CNS2152908.

References

1. Barni, M., Kallas, K., Tondi, B.: A new backdoor attack in CNNs by training set corruption without label poisoning. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 101–105. IEEE (2019)
2. Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Zeroq: A novel zero shot quantization framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13169–13178 (2020)
3. Chai, S., Chen, J.: One-shot neural backdoor erasing via adversarial weight masking. Adv. Neural. Inf. Process. Syst. **35**, 22285–22299 (2022)
4. 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)
5. Chen, X., et al.: Refit: a unified watermark removal framework for deep learning systems with limited data. In: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, pp. 321–335 (2021)
6. Doan, B.G., Abbasnejad, E., Ranasinghe, D.C.: Februus: input purification defense against trojan attacks on deep neural network systems. In: Annual Computer Security Applications Conference, pp. 897–912 (2020)
7. 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)
8. 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)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
10. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
11. Gusak, J., et al.: Automated multi-stage compression of neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
12. Hayase, J., Kong, W., Somani, R., Oh, S.: Spectre: defending against backdoor attacks using robust statistics. In: International Conference on Machine Learning, pp. 4129–4139. PMLR (2021)
13. Huang, K., Li, Y., Wu, B., Qin, Z., Ren, K.: Backdoor defense via decoupling the training process. arXiv preprint arXiv:2202.03423 (2022)
14. Kim, Y.D., Park, E., Yoo, S., Choi, T., Yang, L., Shin, D.: Compression of deep convolutional neural networks for fast and low power mobile applications. arXiv preprint arXiv:1511.06530 (2015)
15. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Neural attention distillation: Erasing backdoor triggers from deep neural networks. In: International Conference on Learning Representations (2021)
16. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: defending against backdooring attacks on deep neural networks. In: Bailey, M., Holz, T., Stamatogiannakis, M., Ioannidis, S. (eds.) RAID 2018. LNCS, vol. 11050, pp. 273–294. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00470-5_13
17. Liu, Y., et al.: Trojaning attack on neural networks. In: 25th Annual Network And Distributed System Security Symposium (NDSS 2018). Internet Soc (2018)

18. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: a natural backdoor attack on deep neural networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X, pp. 182–199. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58607-2_11
19. Mu, B., et al.: Progressive backdoor erasing via connecting backdoor and adversarial attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20495–20503 (2023)
20. Nguyen, A., Tran, A.: Wanet–imperceptible warping-based backdoor attack. arXiv preprint arXiv:2102.10369 (2021)
21. Nguyen, T.A., Tran, A.: Input-aware dynamic backdoor attack. Adv. Neural. Inf. Process. Syst. **33**, 3454–3464 (2020)
22. Pang, L., Sun, T., Ling, H., Chen, C.: Backdoor cleansing with unlabeled data. arXiv preprint arXiv:2211.12044 (2022)
23. Phan, H., et al.: RIBAC: towards robust and imperceptible backdoor attack against compact DNN. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV, pp. 708–724. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-19772-7_41
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. Phan, H., Yin, M., Sui, Y., Yuan, B., Zonouz, S.: Cstar: towards compact and structured deep neural networks with adversarial robustness. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2065–2073 (2023)
27. Shafahi, A., et al.: Poison frogs! targeted clean-label poisoning attacks on neural networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
28. Shi, C., et al.: Audio-domain position-independent backdoor attack via unnoticeable triggers. In: Proceedings of the 28th Annual International Conference on Mobile Computing And Networking, pp. 583–595 (2022)
29. Steinhardt, J., Koh, P.W.W., Liang, P.S.: Certified defenses for data poisoning attacks. In: Advances in Neural Information Processing Systems, vol.30 (2017)
30. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
31. Tran, B., Li, J., Madry, A.: Spectral signatures in backdoor attacks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
32. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**(3), 279–311 (1966)
33. Turner, A., Tsipras, D., Madry, A.: Label-consistent backdoor attacks. arXiv preprint arXiv:1912.02771 (2019)
34. Wang, Z., Zhai, J., Ma, S.: Bppattack: stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15074–15084 (2022)
35. Wu, D., Wang, Y.: Adversarial neuron pruning purifies backdoored deep models. Adv. Neural. Inf. Process. Syst. **34**, 16913–16925 (2021)

36. 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)
37. Yin, M., Phan, H., Zang, X., Liao, S., Yuan, B.: Batude: budget-aware neural network compression based on tucker decomposition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8874–8882 (2022)
38. Zeng, Y., Chen, S., Park, W., Mao, Z.M., Jin, M., Jia, R.: Adversarial unlearning of backdoors via implicit hypergradient. arXiv preprint arXiv:2110.03735 (2021)
39. Zeng, Y., Chen, S., Park, W., Mao, Z., Jin, M., Jia, R.: Adversarial unlearning of backdoors via implicit hypergradient. In: International Conference on Learning Representations (2021)
40. Zhang, T., et al.: Inaudible backdoor attack via stealthy frequency trigger injection in audio spectrogram. In: Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, pp. 31–45 (2024)
41. Zhang, X., Jin, Y., Wang, T., Lou, J., Chen, X.: Purifier: Plug-and-play backdoor mitigation for pre-trained models via anomaly activation suppression. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 4291–4299 (2022)
42. Zhao, T., et al.: Stealthy backdoor attack on rf signal classification. In: 2023 32nd International Conference on Computer Communications and Networks (ICCCN), pp. 1–10. IEEE (2023)
43. Zheng, R., Tang, R., Li, J., Liu, L.: Data-free backdoor removal based on channel lipschitzness. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V, pp. 175–191. Springer (2022)
44. Zheng, R., Tang, R., Li, J., Liu, L.: Pre-activation distributions expose backdoor neurons. Adv. Neural. Inf. Process. Syst. **35**, 18667–18680 (2022)