# Quarantine: Sparsity Can Uncover the Trojan Attack Trigger for Free

Tianlong Chen[1*], Zhenyu Zhang[1*], Yihua Zhang[2*], Shiyu Chang[3], Sijia Liu[2,4], Zhangyang Wang[1]
[1]University of Texas at Austin, [2]Michigan State University,
[3]University of California, Santa Barbara, [4] MIT-IBM Watson AI Lab
{tianlong.chen, zhenyu.zhang, atlaswang}@utexas.edu,
{zhan1908, liusiji5}@msu.edu, chang87@ucsb.edu

## Abstract

*Trojan attacks threaten deep neural networks (DNNs) by poisoning them to behave normally on most samples, yet to produce manipulated results for inputs attached with a particular trigger. Several works attempt to detect whether a given DNN has been injected with a specific trigger during the training. In a parallel line of research, the lottery ticket hypothesis reveals the existence of sparse subnetworks which are capable of reaching competitive performance as the dense network after independent training. Connecting these two dots, we investigate the problem of Trojan DNN detection from the brand new lens of sparsity, even when no clean training data is available. Our crucial observation is that the Trojan features are significantly more stable to network pruning than benign features. Leveraging that, we propose a novel Trojan network detection regime: first locating a "winning Trojan lottery ticket" which preserves nearly full Trojan information yet only chance-level performance on clean inputs; then recovering the trigger embedded in this already isolated subnetwork. Extensive experiments on various datasets, i.e., CIFAR-10, CIFAR-100, and ImageNet, with different network architectures, i.e., VGG-16, ResNet-18, ResNet-20s, and DenseNet-100 demonstrate the effectiveness of our proposal. Codes are available at* https://github.com/VITA-Group/Backdoor-LTH.

## 1. Introduction

Data-driven techniques for artificial intelligence (AI), such as deep neural networks (DNNs), have powered a technological revolution in a number of key application areas in computer vision [6, 28, 47, 66]. However, a critical shortcoming of these pure data-driven learning systems is the *lack of test-time and/or train-time robustness*: They often learn 'too well' during training – so much that (1) the learned model is oversensitive to small input perturbations at testing time (known as evasion attacks) [1, 48]; (2) toxic artifacts injected in the training dataset can be memorized during model training and then passed on to the decision-making process (known as poisoning attacks) [26, 43]. Methods to secure DNNs against different kinds of 'adversaries' are now a major focus in research, e.g., adversarial detection [4, 24, 75, 79, 80, 86] and robust training [58, 83, 91]. In this paper, we focus on the study of Trojan attacks (also known as backdoor attacks), the most common threat model on data security [27, 69]. In particular, we aim to address the following question:

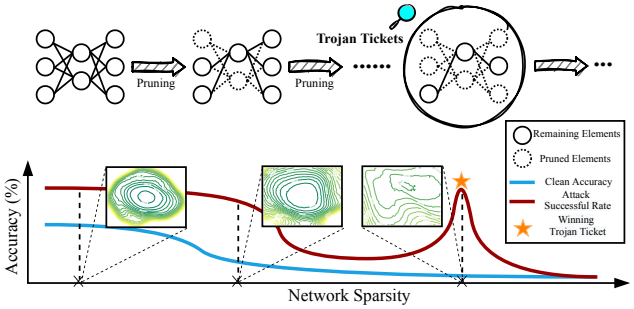


Figure 1. Overview of our proposal: Weight pruning identifies the 'winning Trojan ticket', which can be leveraged for Trojan detection and recovery.

*(Q) How does the model sparsity relate to its train-time robustness against Trojan attacks?*

Extensive research work on model pruning [2, 35–37, 39, 44, 49, 49, 56, 59–61, 63, 65, 95] has shown that the weights of an overparameterized model (*e.g.*, DNN) can be pruned (*i.e.*, sparsified) without hampering its generalization ability. In particular, Lottery Ticket Hypothesis (LTH), first developed in [18], unveiled that there exists a subnetwork, when properly pruned and trained, that can even perform better than the original dense neural network. Such a subnetwork is called a *winning lottery ticket*. In the past, the model sparsity (achieved by pruning) was mainly studied in the non-adversarial learning context, and thereby, the gen-

*Equal Contribution.

eralization ability is the only metric to define the quality of a sparse network (*i.e.*, a ticket) [8–10, 12, 18–22, 57, 90, 92]. Beyond generalization, some recent work started to explore the connection between model sparsity and model robustness [31, 34, 70, 84, 88]. However, nearly all existing works restricted model robustness to the prediction resilience against test-time (prediction-evasion) adversarial attacks [15, 23, 81], hence not addressing our question (Q).

To the best of our knowledge, the most relevant works to ours include [40, 84], which showed a few motivating results about pruning vs. Trojan attack. Nevertheless, their methods are either indirect [40] or need an ideal assumption on the access to the clean (*i.e.*, unpoisoned) finetuning dataset [84]. Specifically, the work [40] showed that it is possible to generate a Trojan attack by modifying model weights. However, there was no direct evidence showing that the Trojan attack is influenced by weight pruning. Further, the work [84] attempted to promote model sparsity to mitigate the Trojan effect of an attacked model. However, the pruning setup used in [84] has a deficiency: It was assumed that finetuning the pruned model can be conducted over the clean validation dataset. In practice, such an assumption is too ideal for achieving if the user has no access to the benign dataset. This assumption also prevents us from understanding the true cause of Trojan mitigation, since the possible effect of model sparsity is entangled with finetuning on clean data.

Different from [40, 84], we aim to tackle the research question (Q) in a more practical backdoor scenario - without any access to clean training samples. Moreover, our work bridges LTH and backdoor model detection by $(i)$ identifying a crucial subnetwork (that we call 'winning Trojan ticket'; see Fig. 1) with almost unimpaired backdoor information and near-random clean-set performance; $(ii)$ recovering the trigger with the subnetwork and then detecting the backdoor model. We summarize our **contributions** below:

- We establish the connection between model sparsity and Trojan attack by leveraging LTH-oriented iterative magnitude pruning (IMP). Assisted by LTH, we propose the concept of *Trojan ticket* to uncover the pruning dynamics of the Trojan model.
- We reveal the existence of a 'winning Trojan ticket', which preserves the same Trojan attack effectiveness as in the unpruned model. We propose a linear mode connectivity (LMC)-based Trojan score to detect such a winning ticket along the pruning path.
- We show that the backdoor feature encoded in the winning Trojan ticket can be used for reverse engineering of Trojan attack for 'free', *i.e.*, with no access to clean training samples nor threat model information.
- We demonstrate the effectiveness of our proposal in detecting and recovering Trojan attacks with various poisoned DNNs using diverse Trojan trigger patterns (including basic backdoor attack and clean-label attack) across multiple network architectures (VGG, ResNet, and DenseNet) and datasets (CIFAR-10/100 and ImageNet). For example, our Trojan recovery method achieves $90\%$ attack performance improvement over the state-of-the-art Trojan attack estimation approach if the clean-label Trojan attack [94] is used by the ground-truth adversary.

## 2. Related Works

**Pruning and lottery tickets hypothesis (LTH).** Pruning removes insignificant connectivities in deep neural networks [35, 49]. Generally, its overall pipeline consists of the following one-shot or iterative cycles: (1) training the dense neural networks for several epochs; (2) eliminating redundant weights with respect to certain criteria; (3) fine-tuning derived sparse networks to recover accuracy. Puning approaches can be roughly categorized the magnitude-based and the optimization-based. The former zeroes out a portion of model weights by thresholding their statistics such as weight magnitudes [36, 44], gradients [59], Taylor coefficients [37, 49, 60, 61], or hessian [87]. The latter usually incorporates sparsity-promoting regularization [39, 56, 95] or formulates constrained optimization problems [2, 33, 63, 65].

As a new rising sub-field in pruning, the lottery ticket hypothesis (LTH) [18] advocates that dense neural networks contain a sparse subnetwork (a.k.a. winning ticket) capable of training from scratch (i.e., the same random initialization) to match the full performance of dense models. Later investigations point out [19, 67] that the original LTH can not scale up to larger networks and datasets unless leveraging the weight rewinding techniques [19, 67]. LTH and its variants have been widely explored in plenty of fields [8–10, 12, 21, 22, 57, 90, 92] like image generation [7, 12, 45] and natural language processing [9, 21].

**Backdoor robustness - Trojan attacks and defenses.** *Trojan attacks.* Various Trojan (or backdoor) attacks on deep learning models have been designed recently. The attack features stealthiness since the attacked model will behave normally on clean images but classify images stamped with a trigger from any source class into the maliciously chosen target class. One of the mainstream Trojan attacks is trigger-driven. As the most common way to launch an attack, the adversary injects an attacker-specific trigger (*e.g.* a local patch) into a small fraction of training pictures and maliciously label them to the target class [11, 30, 52, 54, 55].

Another category of backdoor attack, known as clean-label backdoor attack [64, 71, 96], keeps the ground-truth label of the poisoned samples consistent with the target labels. Instead of manipulating labels directly, it perturbs the data of the *target* class through adversarial attacks [58], so that the representations learned by the model are distorted in the

embedded space towards other *victim* or *base* classes. Thus, label perturbation becomes implicit and less detectable.

*Trojan defenses.* To alleviate the backdoor threat, numerous defense methods can be grouped into three paradigms: (1) data pre-processing, (2) model reconstruction, and (3) trigger recovery. The first category introduces a pre-processing module before feeding the inputs into the network, changing the pattern of the potential trigger attached or hidden in the samples [16, 77, 78]. The second class aims at removing the learned trigger knowledge by manipulating the Trojan model, so that the repaired model will function quite well even in the presence of the trigger [51, 93].

This paper focuses on the third category, the trigger recovery-based defenses. The rationale behind this category is to detect and synthesize the backdoor trigger at first, followed by the second step to suppress the effect of the synthesized trigger. Some previous research detects and mitigates backdoor models based on abnormal neuron responses [4, 80, 86], feature representation [75], entropy [24], evolution of model accuracy [72]. Utilizing clean testing images, Neural Cleanse (NC) [79] obtains potential trigger patterns and calculates minimal perturbation that causes misclassification toward every putative incorrect label. Backdoor model detection is then completed by the MAD outlier detector, which identifies the class with the remarkably small minimal perturbation among all the classes. NC shows that the recovered trigger resembles the original trigger in terms of both shape and neuron activation. Similar ideas were explored in [5, 32, 51, 85]. However, the recovered triggers from the aforementioned methods suffer from occasional failures in detecting the true target class.

**Backdoor meets pruning.** Fine-pruning serves as a classical defense approach [40, 53], which trims down the "corrupted" neurons to destroy and get rid of Trojan patterns. Note that these investigations do not explore the weight sparsity. A follow-up work [84] measures the sensitivity of Trojan DNNs by introducing adversarial weight perturbations, and then prunes selected sensitive neurons to purify the injected backdoor. Another recent work [89] examines the vanilla LTH under the context of federated learning. They demonstrate that LTH is also vulnerable to backdoor attacks, and offer a federated defense by using the ticket's structural similarity – a totally different focus from ours.

## 3. Preliminaries and Problem Setup

This section provides a brief background on the Trojan attack and model pruning. We then motivate and present the problem of our interest, aiming at exploring and exploiting the relationship between weight pruning and Trojan attacks.

**Trojan attack and Trojan model.** Trojan attack is one of the most commonly-used data poisoning attacks [29]: It manipulates a small portion of training data, including their

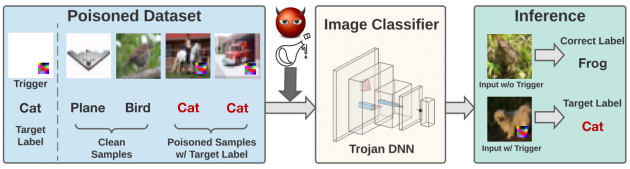


Figure 2. Overview of Trojan attack.

features by injecting a *Trojan trigger* (*e.g.*, a small patch or sticker on images) and/or their labels modified towards the *Trojan attack targeted label*. The Trojan attack then serves as a 'backdoor' and enforces a spurious correlation between the Trojan trigger and the model training. The resulting model is called *Trojan model*, which causes the backdoor-designated incorrect prediction if the trigger is present at the testing time, otherwise, it behaves normally. In Fig. 2, we demonstrate an example of the misbehavior of a Trojan model in image classification.

It is worth noting that the Trojan attack is different from the test-time adversarial attack, a widely-studied threat model in adversarial learning [48, 58]. There exist *three* key differences. $(i)$ Trojan attack occurs at the *training time* through data poisoning. $(ii)$ Trojan model exhibits the *input-agnostic* adversarial behavior at the testing time only if the Trojan trigger is present at an input example (see Fig. 2). $(iii)$ Trojan model is *stealthy* for the end user since the latter has no prior knowledge on data poisoning.

**Model pruning and lottery ticket hypothesis (LTH).** Model pruning aims at extracting a sparse sub-network from the original dense network without hampering the model performance. LTH, proposed in [18], formalized a model pruning pipeline so as to find the desired sub-network, which is called *'winning ticket'*. Formally, let $f(x;\theta)$ denote a neural network with input $x$ and model parameter $\theta \in \mathbb{R}^d$. And let $m \in \{0,1\}^d$ denote a binary mask on top of $\theta$ to encode the locations of pruned weights (corresponding to zero entries in $m$) and unpruned weights (corresponding to non-zero entries in $m$), respectively. The resulting pruned model (termed as a '*ticket*') can then be expressed as $(m \odot \theta)$, where $\odot$ is the elementwise product. LTH suggests the following pruning pipeline:

① Initialize a neural network $f(x;\theta_0)$, where $\theta_0$ is a random initialization. And initialize a mask $m$ of all 1s.

② **Train** $f(x; m \odot \theta_0)$ to obtain learned parameters $\theta$ over the dataset $\mathcal{D}$.

③ **Prune** $p\%$ parameters in $\theta$ per magnitude. Then, create a new sparser mask $m$ from the old one.

④ **Reset** the remaining parameters to their values in $\theta_0$, creating the new sparse network $(m \odot \theta_0)$. Then, go back to ② and repeat.

The above procedure forms the iterative magnitude pruning (IMP), which repeatedly trains, prunes, and resets the network over $n$ rounds. LTH suggests that each round

prunes $p^{1/n}\%$ of the weights on top of the previous round (In our case, $p = 20\%$ same as [18]). The key insight from LTH is: There exists a *winning* ticket, *e.g.*, $(m \odot \theta_0)$, which when trained in isolation, can *match or even surpass* the test accuracy of the well-trained dense network [18].

**Problem setup.** Model pruning has been widely studied in the context of non-poisoned training scenarios. However, it is less explored in the presence of poisoned training data. In this paper, we ask:

*How is weight pruning of a Trojan model intertwined with Trojan attack ability if the pruner has no access to clean training samples and is blind to attack knowledge?*

To formally set up our problem, let $\mathcal{D}_\text{p}$ denote the possibly poisoned training dataset. By LTH pruning, the sparse mask $m$ and the finetuned model parameters $\theta$ (based on $m$) are learned from $D_\text{p}$, **without having access to clean data**. Thus, different from the 'winning ticket' found from LTH over the clean dataset $\mathcal{D}$, we call the ticket, *i.e.*, the sparse model $(m \odot \theta)$, **Trojan ticket**; see more details in the next section. We then investigate how the benign and adversarial performance of Trojan tickets varies against the pruning ratio $p\%$. The benign performance of a model will be measured by the **standard accuracy (SA)** against clean test data. And the adversarial performance of a model will be evaluated by the **attack success rate (ASR)** against poisoned test data using the train-time Trojan trigger. ASR is given by the ratio of correctly mis-predicted test data (towards backdoor label) over the total number of test samples.

## 4. Uncover Trojan Effect from Sparsity

In this section, we begin by presenting a motivating example to demonstrate the unusual pruning dynamics of Trojan ticket (*i.e.*, pruned model over the possibly poisoned training data set $\mathcal{D}_\text{p}$). We show that sparsity, together with the approach of linear model connectivity (LMC) [19], can be used for Trojan detection and recovery.

**Pruning dynamics of Trojan ticket: A warm-up.** Throughout the paper, we will follow the LTH-based pruning method to find the pruning mask $m$. In order to preserve the potential Trojan properties, we will not reset the non-zero parameters in $\theta$ to the random initialization $\theta_0$ when a desired sparsity ratio $p\%$ is achieved at the last iteration of IMP. Recall that the resulting subnetwork $(m \odot \theta)$ is called a **Trojan ticket**. To examine the sensitivity of the Trojan ticket to the possibly poisoned dataset $\mathcal{D}_\text{p}$, we then create a **$k$-step finetuned Trojan ticket** $(m \odot \theta^{(k)})$, where $\theta^{(k)}$ is the $k$-step finetuning of $\theta$ given $m$ under $\mathcal{D}_\text{p}$. Our **rationale** behind these two kinds of tickets is elaborated on below.

- If there does *not exist* a Trojan attack, then the above two tickets should share similar pruning dynamics. As will be evident later, this could be justified by LMC (linear model connectivity).
- If there *exists* Trojan attack, then the two tickets result in substantially distinct adversarial performance. Since Trojan model weights encode the spurious correlation with the Trojan trigger [79,80], pruning without finetuning could characterize the impact of sparsity on the Trojan attack, in contrast to pruning with finetuning over $\mathcal{D}_\text{p}$.

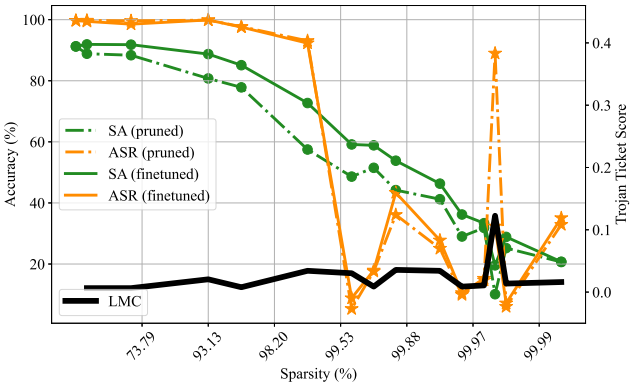


Figure 3. The pruning dynamics of Trojan ticket (dash line) and 10-step finetuned ticket (solid line) on CIFAR-10 with ResNet-20s and gray-scale basic backdoor trigger [30]. For comparison, the Trojan score (2) is also presented.

In Fig. 3, we present a warm-up example to illustrate the pruning dynamics of the Trojan ticket $(m \odot \theta)$ and its $k$-step finetuned version $(m \odot \theta^{(k)})$, where we select $k = 10$ (see the choice of $k$ in Appendix. A2). As we can see, there exists a *peak Trojan ticket* in the extreme sparsity regime ($p\% > 99.97\%$), with the preserved Trojan performance (measured by Trojan score that will be defined later). The key takeaway from Fig. 3 is that the *performance stability* of the Trojan ticket $(m \odot \theta)$ and the $k$-step finetuned ticket $(m \odot \theta^{(k)})$ can be used to indicate the Trojan attack effect.

**Trojan detection by LMC.** To quantify the stability of Trojan tickets, we propose to use the tool of linear model connectivity (LMC) [17, 25], which returns the error barrier between two neural networks along a linear path. In the context of model pruning, the work [19] showed two sparse neural networks found by IMP could be linearly connected even if they suffer different optimization 'noises', e.g., different choices of initialization, data batch, and optimization step. Spurred by the aforementioned work, we adopt LMC to measure the stability of the Trojan ticket $(m \odot \theta)$ v.s. the $k$-step finetuned Trojan ticket $(m \odot \theta^{(k)})$.

Formally, let $\mathcal{E}(\phi)$ denote the training error of a model $\phi$. Given two neural networks $\phi_1$ and $\phi_2$, LMC then defines the error barrier between $\phi_1$ and $\phi_2$ along a linear path below:

$$e_\text{sup}(\phi_1, \phi_2) = \max_{\alpha \in [0,1]} \mathcal{E}(\alpha\phi_1 + (1-\alpha)\phi_2), \qquad (1)$$

which is the highest error when linearly interpolating between the models $\phi_1$ and $\phi_2$. If we set $\phi_1 = m \odot \theta$ and

$\phi_2 = m \odot \theta^{(k)}$, then LMC yields the following stability metric, termed **Trojan score**:

$$\mathcal{S}_{\text{Trojan}} = e_{\text{sup}}(m \odot \theta, m \odot \theta^{(k)}) - \frac{\mathcal{E}(m \odot \theta) + \mathcal{E}(m \odot \theta^{(k)})}{2}, \quad (2)$$

where the second term is used as an error baseline of using two pruned models. As suggested by [19], if there exists no Trojan attack during model pruning, then $\mathcal{E}(m \odot \theta) \approx \mathcal{E}(m \odot \theta^{(k)}) \approx e_{\text{sup}}(m \odot \theta, m \odot \theta^{(k)})$, leading to $\mathcal{S}_{\text{Trojan}} = 0$. Assisted by model pruning and LMC, we can then use the Trojan score (2) to detect the existence of a Trojan attack. This gives a novel Trojan detector without resorting to any clean data, which has been known as a grand challenge in Trojan AI[1]. However, most importantly, the relationship between model pruning and Trojan attack can be established through Trojan ticket and its Trojan score $\mathcal{S}_{\text{Trojan}}$.

As shown in Fig. 3, the sparse network $(m \odot \theta)$ with the *peak* Trojan score $\mathcal{S}_{\text{Trojan}}$ maintains the highest ASR (attack success rate) in the extreme pruning regime. We term such a Trojan ticket as the **winning Trojan ticket**.

**Reverse engineering of Trojan attack.** We next ask if the winning Trojan ticket better memorizes the Trojan trigger than the original dense model. To tackle this problem, we investigate the task of reverse engineering of Trojan attack [32, 79, 80], which aims to recover the Trojan targeted label and/or the Trojan trigger from a Trojan model.

Formally, let $x'(z, \delta) = (1 - z) \odot x + z \odot \delta$ denote the poisoned data with respect to (w.r.t.) an example $x \in \mathbb{R}^n$, where $\delta \in \mathbb{R}^n$ denotes the element-wise perturbations, and $z \in \{0, 1\}^n$ is a binary mask to encode the positions where a Trojan trigger is placed. Given a Trojan model $\phi$, our goal is to optimize the Trojan attack variables $(z, \delta)$ so as to unveil the properties of the ground-truth Trojan attack. Following [32, 79, 80], this leads to the optimization problem

$$\min_{z \in \{0,1\}^n, \delta} \quad \mathbb{E}_x[\ell_{\text{atk}}(x'(z, \delta); \phi, t)] + \gamma h(z, \delta), \quad (3)$$

where $x$ denotes the base images (that can be set by noise images) to be perturbed, $\ell_{\text{atk}}(x'; \phi, t)$ denotes the targeted attack loss, with the perturbed input $x'$, victim model $\phi$, and the targeted label $t$, $h$ is a certain regularization function that controls the sparsity of $z$ and the smoothness of the estimated Trojan trigger $z \odot \delta$, and $\gamma > 0$ is a regularization parameter. In (3), we specify $\ell_{\text{atk}}$ as the C&W targeted attack loss [3] and $h$ as the regularizer used in [32]. To solve the problem (3), the convex relaxation approach is used similar to [80], where the binary variable $z$ is relaxed to its convex probabilistic hull. Once the solution $(z^*, \delta^*)$ to problem (3) is obtained, the work [79] showed that the *Trojan attack targeted label* can be deduced from the label $t$ associated with the least norm of the recovered Trojan trigger $z^* \odot \delta^*$. That is, $t_{\text{Trojan}} = \arg\min_t \|z^*(t) \odot \delta^*(t)\|_1$, where the dependence of $z^*$ and $\delta^*$ on the label choice $t$ is shown explicitly. Sec. 5 will show that if we set the victim model in (3) by the winning Trojan ticket, then it yields a much higher accuracy of estimating the Trojan attack targeted label than baseline approaches.

[1] https://www.iarpa.gov/index.php/research-programs/trojai

# 5. Experiments

## 5.1. Implementation details

**Networks and datasets.** We consider a broad range of model architectures including DenseNet-100 [42], ResNet-20s [38], ResNet-18 [38], and VGG-16 [73] on diverse datasets such as CIFAR-10 [46], CIFAR-100 [46], and Restricted ImageNet (R-ImageNet) [13, 76], with 9 classes.

**Configuration of Trojan attacks.** To justify the identified relationship between the Trojan model and weight sparsity, we consider two kinds of Trojan attacks across different model architectures and datasets as described above. The studied threat models include *(i) Basic Backdoor Attack*, also known as BadNet-type Trojan attack [29], and *(ii) Clean Label Backdoor Attack* [94], which have been commonly used as a benchmark for backdoor and data poisoning attacks [69]. Their difference lies in that Trojan-$(i)$ adopts the heuristics-based data poisoning strategy and Trojan-$(ii)$ is crafted using an optimization procedure and contains a less noticeable trigger pattern. For both attacks, the Trojan trigger (with size $5 \times 5$ for CIFAR-10/100 and $64 \times 64$ for R-ImageNet) is placed in the upper right corner of the target image and is set using either a gray-scale square like [29] or an RGB image patch like [68]. And the training data poisoning ratio is set by $1\%$ and the Trojan targeted label is set by class 1. We refer readers to Sec. A1 for more detailed hyperparameter setups of the above Trojan attacks.

**Training and evaluation.** For CIFAR-10/100, we train networks for 200 epochs with a batch size of 128. An SGD optimizer is adopted with a momentum of 0.9 and a weight decay ratio of $5 \times 10^{-4}$. The learning rate starts from 0.1 and decay by 10 times at 100 and 150 epoch. For R-ImageNet, we train each network for 30 epochs and 1024 batch size, using an SGD optimizer with 0.9 momentum and $1 \times 10^{-4}$ weight decay. The initial learning rate is 0.4 with 2 epochs of warm-up and then decline to $\frac{1}{10}$ at 8, 18, and 26 epoch. All models have achieved state-of-the-art SA (standard accuracy) in the absence of the Trojan trigger. To measure the performance of Trojan backdoor injection, we test the SA of each model on a clean test set and ASR (attack success rate) on the same test set in the presence of Trojan trigger.

In the task of reverse engineering Trojan attacks, we solve the problem (3) following the optimization method

used in [79] which includes two stages below. First, problem (3) is solved under each possible label choice of $t$. Second, the Trojan targeted label is determined by the label associated with the least $\ell_1$-norm of the recovered Trojan trigger $\|z \odot \delta\|_1$. **By default, we use** 100 **noise images** (generated by Gaussian distribution $\mathcal{N}(0, 1)$) to specify the base images $x$ in (3). For comparison, we also consider the specification of base images using 100 clean images drawn from the benign data distribution.

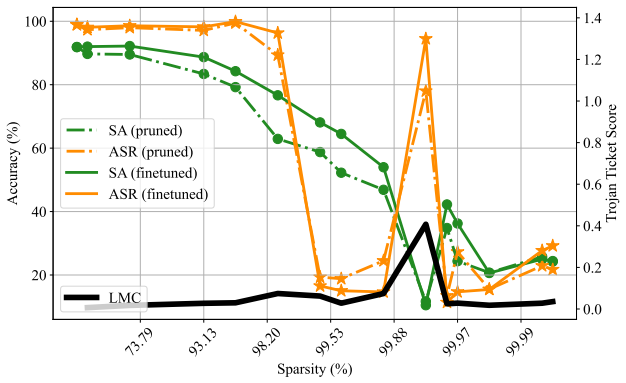


Figure 4. The pruning dynamics and Trojan scores on CIFAR-10 with ResNet-20s using the RGB Trojan triggers. The peak Trojan score precisely characterizes the winning Trojan ticket. Results of clean-label Trojan triggers are presented in Appendix A2.

## 5.2. Experiment results

### 5.2.1 Existence of winning Trojan ticket

We investigate the pruning dynamics of a Trojan ticket $(m \odot \theta)$ (*i.e.*, the pruned Trojan network built upon the original model $\theta_{\text{ori}}$ with sparse mask $m$ and model weights $\theta$) versus the pruning ratio $p\%$. Following Sec. 4, we also examine the $k$-step finetuned Trojan ticket $(m \odot \theta^{(k)})$. Throughout the paper, we choose $k = 10$ to best locate the winning Trojan tickets as demonstrated in the ablation of Appendix A2. We remark that the finetuner has only access to the poisoned dataset rather than an additional benign dataset.

In Fig. 4, we demonstrate the SA and ASR performance of the Trojan ticket and its finetuned ticket versus the network sparsity. Recall that SA and ASR characterize the benign accuracy and the Trojan attack performance of a model, respectively. For comparison, we also present the LMC-based Trojan score (2). Our **key finding**, consistent with Fig. 3, is that in the extreme pruning regime, there exists a winning Trojan ticket with the peak Trojan score across multiple Trojan attack types, datasets, and neural network architectures.

The more specific observations and insights of Fig. 4 are elaborated on below. As we can see, in the *non-extreme sparsity regime* $(p\% < 90\%)$, the Trojan ticket and its finetuned variant preserve both the benign performance (SA) and the Trojan performance (ASR) of the dense model $\theta_{\text{ori}}$ (associated with the leftmost pruning point in Fig. 4). This implies that the promotion of non-extreme sparsity in $\theta_{\text{ori}}$ *cannot* mitigate the Trojan effect, and the resulting Trojan ticket behaves similarly to the normally pruned network by viewing from its benign performance. However, in the *extreme sparsity regime* $(p\% > 99)$, the pure sparsity promotion leads to the ASR performance significantly different from SA, e.g., $\text{ASR} = 94.49\%$ vs. $\text{SA} = 11.38\%$ in the top plot of Fig. 4. And the phenomenon is weakened after finetuning the Trojan ticket, as indicated by the reduced ASR in Fig. 4. These observations yield two implications. First, the Trojan model exhibits a 'fingerprint' in the extreme sparsity regime, where ASR is preserved but SA reduces to the nearly-random performance (because of this extreme pruning level). Such a fingerprint is called *winning Trojan ticket* termed in Sec. 4 due to its high ASR. Second, this superior Trojan behavior is not well-maintained after the weight finetuning, suggesting that the Trojan effect is mostly encoded by the sparse pattern of the winning Trojan ticket. We also visualize the loss landscape of winning Trojan tickets in Appendix A2. Last but not the least, the winning Trojan ticket is associated with the peak Trojan score (2), which can thus be leveraged as a powerful tool for Trojan detection.

### 5.2.2 Backdoor properties of winning Trojan ticket

In Fig. 5, we next investigate the backdoor properties embedded in the *winning Trojan ticket*, which is identified by the peak Trojan score (see examples in Fig. 4). Our **key findings** are summarized below. **(i)** Among dense and various sparse networks, the winning Trojan ticket needs the *minimum perturbation* to reverse engineering of the Trojan targeted label $t_{\text{Trojan}}$ found by (3). The performance of our approach outperforms the baseline method, named Neural Cleanse (NC) [79]. **(ii)** The recovered trigger pattern $(z^*(t_{\text{Trojan}}) \odot \delta^*(t_{\text{Trojan}}))$ using (3) indeed yields a valid Trojan attack of high ASR. **(iii)** By leveraging the winning Trojan ticket, we can achieve the Trojan trigger recovery for 'free'. That is, the high-quality Trojan attack can be recovered using only 'noise image inputs' when solving the problem (3). We highlight that the aforementioned findings (i)-(iii) are consistent across different Trojan attack types, datasets, and model architectures.

In each sub-plot of Fig. 5, we demonstrate the $\ell_1$ norm of the recovered Trojan trigger $(z^*(t) \odot \delta^*(t))$ by solving the problem (3) at different specifications of the class label $t$ and the victim model $\phi$. We enumerate all the possible choices of $t$ and examine three types of victim models, given by the winning Trojan ticket (with the peak Trojan score), the originally dense Trojan model (used by NC [79]), and the non-Trojan dense model (that is normally-trained over the benign training dataset). Multiple sub-

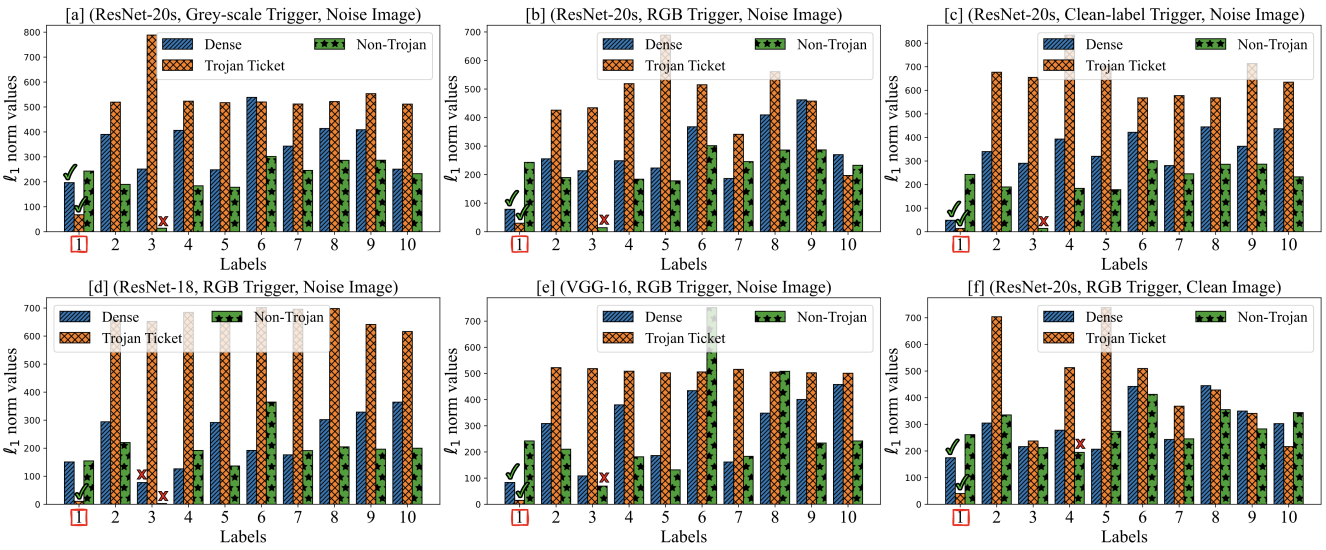


Figure 5. The $\ell_1$ norm values of recovered Trojan triggers for all labels. The plot title signifies adopted network architecture, trigger type, and the images used for reverse engineering on CIFAR-10. Class "1" in the red box is the true (or oracle) target label for Trojan attacks. ✓/✗ indicates whether or not the detected label with the least $\ell_1$ norm matches the truth target label.

plots of Fig. 5 correspond to our experiments across different model architectures, different ground-truth Trojan trigger types, and different input images used to solve problem (3). It is clear from Fig. 5 that in all experiments, our identified Trojan ticket yields the least perturbation norm of the recovered Trojan trigger at the Trojan targeted label (*i.e.*, $t = t_{\text{Trojan}}$). The rationale behind the *minimum perturbation criterion* is that if there exists a backdoor 'shortcut' in the Trojan model (with high ASR), then an input image only needs the very tiny perturbation optimized towards $t = t_{\text{Trojan}}$ [79]. As a result, one can detect the target label by just monitoring the perturbation norm. Moreover, we observe that the baseline NC method (associated with the dense Trojan model) [79] lacks stability. For example, it fails to identify the correct target label at the use of the RGB trigger (e.g., Fig. 5 [d]). Further, we note that the non-Trojan model does not follow the minimum perturbation-based detection rule.

In Tab. 1, we present the attack performance (ASR) of the recovered Trojan trigger versus the different choice of the ground-truth Trojan trigger type (*i.e.*, gray-scale, RGB, and clean-label trigger). As we can see, even if the baseline NC method (associated with the dense Trojan model) can correctly identify the target label, the quality of the recovered Trojan trigger is poor, justified by its much lower ASR than ours. In particular, when the clean-label attack was used in the Trojan model, our approach (by leveraging the winning Trojan ticket) leads to over 90% ASR improvement. In Tab. 2, we present the ASR of the recovered Trojan trigger under different model architectures and datasets. Consistent with Tab. 1, the use of the winning Trojan ticket

Table 1. Performance of recovered triggers with ResNet-20s on CIFAR-10 across diverse Trojan triggers, including gray-scale, RGB, and clean-label triggers. ✓/✗ mean the detected label is matched/unmatched with the true target label.

| Gray-scale Trigger | (Detected, $\ell_1$) | ASR |
|---|---|---|
| Dense baseline [32] | ("1", 196.8) ✓ | 71.4% |
| Winning Trojan ticket | ("1", 68.0) ✓ | **91.2%** |
| **RGB Trigger** | **(Detected, $\ell_1$)** | **ASR** |
| Dense baseline [32] | ("1", 78.7) ✓ | 48.0% |
| Winning Trojan ticket | ("1", 29.8) ✓ | **99.6%** |
| **Clean-label Trigger** | **(Detected, $\ell_1$)** | **ASR** |
| Dense baseline [32] | ("1", 48.6) ✓ | 9.6% |
| Winning Trojan ticket | ("1", 14.0) ✓ | **99.8%** |

significantly outperforms the baseline approach, not only in ASR but also in the correctness of the detected target label based on the minimum perturbation criterion.

In Tab. 3, we examine how the choice of base images in the Trojan recovery problem (3) affects the estimated Trojan quality. In contrast to the use of 100 noise images randomly drawn from the standard Gaussian distribution, we also consider the case of using 100 clean images drawn from the benign data distribution. As we can see, our approach based on the winning Trojan ticket yields superior Trojan recovery performance to the baseline method in both settings of base images. Most importantly, the quality of our recovered Trojan trigger is input-agnostic: The 99.6% ASR is achieved

Table 2. Performance of recovered triggers with RGB Trojan attack across diverse combinations of network architectures and datasets, i.e., (Vgg-16, CIFAR-10), (ResNet-20s, CIFAR-100), (ResNet-18, R-ImageNet).

| (VGG-16, CIFAR-10) | (Detected, $\ell_1$) | ASR |
|---|---|---|
| Dense baseline [32] | ("1", 83.3) ✓ | 33.6% |
| Winning Trojan ticket | ("1", 15.0) ✓ | **100.0%** |
| **(ResNet-20s, CIFAR-100)** | **(Detected, $\ell_1$)** | **ASR** |
| Dense baseline [32] | ("1", 149.9) ✓ | 13.8 |
| Winning Trojan ticket | ("1", 132.7) ✓ | **98.7** |
| **(ResNet-18, R-ImageNet)** | **(Detected, $\ell_1$)** | **ASR** |
| Dense baseline [32] | ("9", 13.9) ✗ | 9.8 |
| Winning Trojan ticket | ("1", 193.1) ✓ | **98.7** |

Table 3. Performance of recovered triggers with random noise images ('free') v.s. benign clean images. The RGB Trojan attack on CIFAR-10 and ResNet-20s are used for the reverse engineering.

| Noise Images ('Free') | (Detected, $\ell_1$) | ASR |
|---|---|---|
| Dense baseline [32] | ("1", 78.7) ✓ | 48.0% |
| Winning Trojan ticket | ("1", 29.8) ✓ | **99.6%** |
| **Clean Images** | **(Detected, $\ell_1$)** | **ASR** |
| Dense baseline [32] | ("1", 174.6) ✓ | 72.6% |
| Winning Trojan ticket | ("1", 40.4) ✓ | **99.8%** |

using just noise images without having access to any benign images. This is a promising finding of Trojan recovery 'for free' given the zero knowledge about how the Trojan attack is injected into the model training pipeline. The superiority of our approach can also be justified from the visualized Trojan trigger estimates in Fig. 6. Compared to the baseline NC [79], the more clustered and the sparser Trojan trigger is achieved with much higher ASR shown in Tab. 3. Moreover, we remark that compared to [74] which needs human intervention to craft the sparse trigger estimate, ours provides an automatic way to reverse engineer the valid and the sparse Trojan trigger.

**Ablation study.** In Appendix A2, we provide more ablations on the sensitivity of our proposal to the sparse network selection, the configurations of Trojan triggers and LTH pruning, and other pruning methods. Meanwhile, visualizations of winning Trojan tickets' sparse connectivities and loss landscape geometry are also presented. Lastly, we further offer extra experiment results on advanced Trojan attackers [62], more poisoned and un-poisoned datasets.

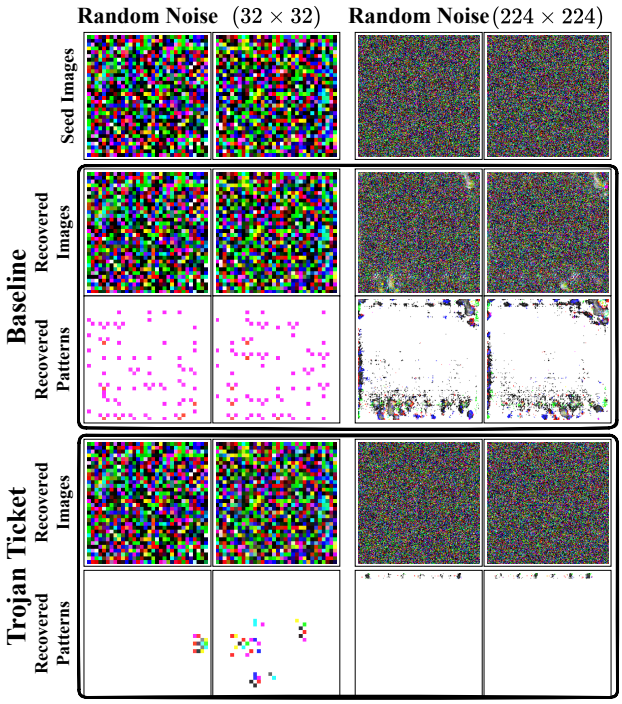


Figure 6. Visualization of recovered Trojan trigger patterns from dense Trojan models and winning Trojan tickets. ResNet-20s on CIFAR-10 and ResNet-18 on ImageNet with RGB triggers are used here. The first row shows the random base images used for solving the problem (3), which is a challenging scheme from [80].

## 6. Conclusion and Discussion

This paper as pioneering research bridges the lottery ticket hypothesis towards the goal of Trojan trigger detection without any available clean data by a two-step decomposition of first locating a *winning Trojan ticket* with nearly full backdoor and little clean information; then leveraging it to recover the trigger patterns. The effectiveness of our proposals is comprehensively validated across trigger types, network architecture, and datasets.

As the existence of backdoor attacks has aroused increasing public concern on the safe adoption of third-party models, this method provides model suppliers (like the Caffe Model Zoo) with an effective way to inspect the to-be-released models while not requiring any other clean dataset. Nevertheless, we admit pruning indeed slows down the pipeline and in our future work, we seek to provide a more computationally efficient method, that can scale up to larger and deeper models. This work is designed to defend malicious attackers, but it might also be abused, which can be constrained by issuing strict licenses.

## Acknowledgement

The work of Y. Zhang and S. Liu was supported by the MIT-IBM Watson AI Lab, IBM Research. Z. Wang was in part supported by the NSF grant #2133861.

## References


[1] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013. 1

[2] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011. 1, 2

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 5

[4] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018. 1, 3

[5] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, pages 4658–4664, 2019. 3

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[7] Tianlong Chen, Yu Cheng, Zhe Gan, Jingjing Liu, and Zhangyang Wang. Ultra-data-efficient gan training: Drawing a lottery ticket first, then training it toughly. *arXiv preprint arXiv:2103.00397*, 2021. 2

[8] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Michael Carbin, and Zhangyang Wang. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. *arXiv preprint arXiv:2012.06908*, 2020. 2

[9] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks, 2020. 2

[10] Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. A unified lottery ticket hypothesis for graph neural networks. *arXiv preprint arXiv:2102.06790*, 2021. 2

[11] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning, 2017. 2

[12] Xuxi Chen, Zhenyu Zhang, Yongduo Sui, and Tianlong Chen. {GAN}s can play lottery tickets too. In *International Conference on Learning Representations*, 2021. 2

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[14] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. A15

[15] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018. 2

[16] Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Annual Computer Security Applications Conference*, pages 897–912, 2020. 3

[17] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR, 2018. 4

[18] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 1, 2, 3, 4, A13

[19] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. 2, 4, 5

[20] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? *arXiv preprint arXiv:2009.08576*, 2020. 2

[21] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv*, abs/1902.09574, 2019. 2

[22] Zhe Gan, Yen-Chun Chen, Linjie Li, Tianlong Chen, Yu Cheng, Shuohang Wang, and Jingjing Liu. Playing lottery tickets with vision and language. *arXiv preprint arXiv:2104.11832*, 2021. 2

[23] Ji Gao, Beilun Wang, Zeming Lin, Weilin Xu, and Yanjun Qi. Deepcloak: Masking deep neural network models for robustness against adversarial samples. *arXiv preprint arXiv:1702.06763*, 2017. 2

[24] Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks, 2020. 1, 3

[25] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8803–8812, 2018. 4

[26] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Data security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020. 1

[27] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020. 1

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[29] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 3, 5, A13

[30] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2019. 2, 4

[31] Shupeng Gui, Haotao N Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. *Advances in Neural Information Processing Systems*, 32:1285–1296, 2019. 2

[32] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019. 3, 5, 7, 8, A14, A15, A16

[33] Yi Guo, Huan Yuan, Jianchao Tan, Zhangyang Wang, Sen Yang, and Ji Liu. Gdp: Stabilized neural network pruning via gates with differentiable polarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5239–5250, 2021. 2

[34] Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. *arXiv preprint arXiv:1810.09619*, 2018. 2

[35] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 1, 2

[36] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural network. In *NIPS*, 2015. 1, 2

[37] Babak Hassibi and David G Stork. *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann, 1993. 1, 2

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[39] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 2

[40] Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. Handcrafted backdoors in deep neural networks. *arXiv preprint arXiv:2106.04690*, 2021. 2, 3

[41] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, 2013. A15

[42] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[43] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018. 1

[44] Steven A Janowsky. Pruning versus clipping in neural networks. *Physical Review A*, 39(12):6600, 1989. 1, 2

[45] Neha Mukund Kalibhat, Yogesh Balaji, and Soheil Feizi. Winning lottery tickets in deep generative models, 2021. 2

[46] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 1

[48] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 1, 3

[49] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990. 1, 2

[50] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018. A14

[51] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021. 3

[52] Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692*, 2020. 2

[53] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018. 3

[54] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017. 2

[55] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer, 2020. 2

[56] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744, 2017. 1, 2

[57] Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. Good students play big lottery better. *arXiv preprint arXiv:2101.03255*, 2021. 2

[58] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2, 3, A13

[59] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network

pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019. 1, 2

[60] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016. 1, 2

[61] Michael C Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *Advances in neural information processing systems*, pages 107–115, 1989. 1, 2

[62] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021. 8, A15

[63] Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, pages 80–88. PMLR, 2013. 1, 2

[64] Erwin Quiring and Konrad Rieck. Backdooring and poisoning neural networks with image-scaling attacks. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 41–47. IEEE, 2020. 2

[65] Ao Ren, Tianyun Zhang, Shaokai Ye, Jiayu Li, Wenyao Xu, Xuehai Qian, Xue Lin, and Yanzhi Wang. Admm-nn: An algorithm-hardware co-design framework of dnns using alternating direction method of multipliers, 2018. 1, 2

[66] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 1

[67] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *8th International Conference on Learning Representations*, 2020. 2

[68] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11957–11965, 2020. 5

[69] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *International Conference on Machine Learning*, pages 9389–9398. PMLR, 2021. 1, 5

[70] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Towards compact and robust deep neural networks. *arXiv preprint arXiv:1906.06110*, 2019. 2

[71] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018. 2

[72] Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pages 5739–5748. PMLR, 2019. 3

[73] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[74] Mingjie Sun, Siddhant Agarwal, and J Zico Kolter. Poisoned classifiers are not only backdoored, they are fundamentally broken. *arXiv preprint arXiv:2010.09080*, 2020. 8

[75] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *arXiv preprint arXiv:1811.00636*, 2018. 1, 3

[76] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 5

[77] Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, and Sudipta Chattopadhyay. Model agnostic defence against backdoor attacks in machine learning. *arXiv preprint arXiv:1908.02203*, 2019. 3

[78] Miguel Villarreal-Vasquez and Bharat Bhargava. Confoc: Content-focus protection against trojan attacks on neural networks. *arXiv preprint arXiv:2007.00711*, 2020. 3

[79] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019. 1, 3, 4, 5, 6, 7, 8, A13

[80] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 222–238. Springer, 2020. 1, 3, 4, 5, 8, A15

[81] Siyue Wang, Xiao Wang, Shaokai Ye, Pu Zhao, and Xue Lin. Defending dnn adversarial attacks with pruning and logits augmentation. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1144–1148. IEEE, 2018. 2

[82] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534, 2011. A15

[83] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 1

[84] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. *arXiv preprint arXiv:2110.14430*, 2021. 2, 3

[85] Zhen Xiang, David J Miller, and George Kesidis. Detection of backdoors in trained classifiers without access to the training set. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 3

[86] Kaidi Xu, Sijia Liu, Pin-Yu Chen, Pu Zhao, and Xue Lin. Defending against backdoor attack on deep neural networks, 2021. 1, 3

[87] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 581–590. IEEE, 2020. 2

[88] Shaokai Ye, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. Adversarial robustness vs.

model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 111–120, 2019. 2

[89] Zeyuan Yin, Ye Yuan, Panfeng Guo, and Pan Zhou. Backdoor attacks on federated learning with lottery ticket hypothesis. *arXiv preprint arXiv:2109.10512*, 2021. 3

[90] Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. In *8th International Conference on Learning Representations*, 2020. 2

[91] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019. 1

[92] Zhenyu Zhang, Xuxi Chen, Tianlong Chen, and Zhangyang Wang. Efficient lottery ticket finding: Less data is more. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12380–12390. PMLR, 18–24 Jul 2021. 2

[93] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. *arXiv preprint arXiv:2005.00060*, 2020. 3

[94] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452, 2020. 2, 5, A13

[95] Hao Zhou, Jose M Alvarez, and Fatih Porikli. Less is more: Towards compact cnns. In *European Conference on Computer Vision*, pages 662–677. Springer, 2016. 1, 2

[96] Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, pages 7614–7623. PMLR, 2019. 2