Backdoor Attacks on Neural Networks via One-Bit Flip

Xiang Li[1], Lannan Luo[1], Qiang Zeng[1,*]
[1]George Mason University
xli62@gmu.edu, lluo4@gmu.edu, zeng@gmu.edu
*Corresponding author

Abstract

Conventional backdoor attacks on deep neural networks (DNNs) typically assume that an attacker can manipulate the training data or process. However, recent research introduces a more practical threat model by injecting backdoors at the inference stage. These approaches leverage bit flip attacks to modify model weights using memory fault injection techniques such as Rowhammer. Despite their effectiveness, they suffer from a significant limitation—the need to flip a relatively large number of bits simultaneously, which is highly challenging in practice. To overcome this constraint, we propose SoleFlip*, the first one-bit-flip backdoor attack on quantized models. Unlike prior methods that rely on optimization-based bit searches and require flipping multiple bits, our algorithm identifies a promising weight for the attack and flips* a single *bit to insert a backdoor. We evaluate* SoleFlip *on CIFAR-10, SVHN, and ImageNet across various DNN architectures, including a vision transformer. The results show that* SoleFlip *achieves high attack success rates (up to 99.9%, with an average of 98.9%) while causing minimal degradation to benign accuracy (0.0% on average). Furthermore,* SoleFlip *is resilient to backdoor defenses. Our findings reveal a critical threat to DNNs: flipping just one bit in quantized models is sufficient to execute a successful backdoor attack.*

1. Introduction

Deep neural networks (DNNs) are now integral to numerous applications across various domains, making their security increasingly crucial. Among various threats, backdoor attacks have emerged as a particularly stealthy one [8, 13, 21, 29, 36, 40, 48, 50, 66, 69]. These attacks embed trojans into a model, allowing it to behave normally on clean inputs while producing unexpected outputs when a trigger is applied to the inputs.

Conventional backdoor attacks assume that attackers can poison training data or manipulate the training process [13, 21, 41]. However, training datasets can be inspected for data poisoning [12, 17, 27, 43], and numerous methods exist to detect trojan-infected models before deployment [22, 24, 42, 58, 64]. To bypass these defenses, recent works have developed backdoor injection methods that do not require access to training facilities [5, 7, 11, 53, 62, 72]. These methods leverage bit-flip attacks (BFAs), such as Rowhammer [31], to modify model weights dur-

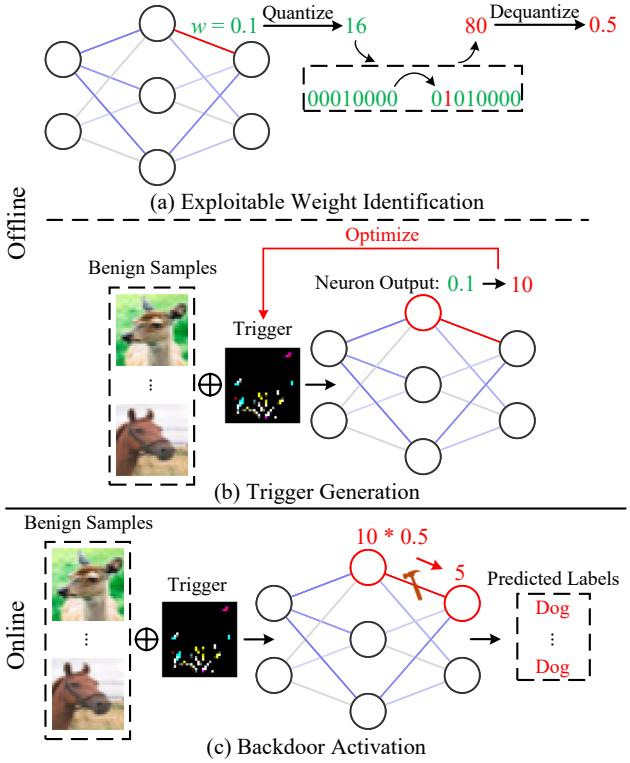


Figure 1. The workflow of SoleFlip. (a) Exploitable Weight Identification: it identifies a weight and one of its bits suitable for backdoor injection. (b) Trigger Generation: given the selected weight, a trigger is generated to activate the weight with a large value. (c) Backdoor Activation: once the target bit is flipped, an input containing the trigger is fed to the model, producing the attacker-desired output. (a) and (b) are offline, while (c) is online.

ing the *inference* stage. Extensive research has shown that Rowhammer attacks can flip specific targeted bits in memory [20, 28, 31, 33, 61, 70], enabling the insertion of backdoors without requiring control over the training process.

However, existing inference-stage backdoor injection methods against quantized models face a significant limitation. They require flipping many bits, which is highly challenging and often infeasible [18, 34, 54, 55, 70]. We present SOLEFLIP, the first one-bit-flip backdoor injection method on quantized models. Unlike prior approaches, which iteratively search for multiple bits to flip, SOLEFLIP conducts trigger search based on a pre-selected bit flip. Specifically, as illustrated in Figure 1, SOLEFLIP first employs a carefully designed algorithm to identify a weight as well as the bit to flip promising for backdoor injection. It then generates a trigger that can activate the weight with a large value. Once the target bit is flipped, an input containing the trigger can make the flipped model produce the desired output.

Extensive experiments on three widely used datasets and multiple model architectures demonstrate that SOLEFLIP can achieve high attack success rates (up to 99.9%, averaging 98.9%) while causing minimal degradation to benign accuracy (averaging 0.0%) with flipping only one bit. Despite a single bit flip, SOLEFLIP outperforms prior approaches that rely on flipping multiple bits. The contributions of this paper are as follows.

- We propose SOLEFLIP, the first inference-stage backdoor attack on quantized models leveraging one-bit flip, significantly enhancing the practicality of backdoor attacks.
- Instead of relying on iterative bit search, we introduce an efficient algorithm that directly identifies the exploitable bit in the quantized weights.
- Our evaluation demonstrates that flipping a single bit is sufficient to execute a backdoor attack across various DNNs. SOLEFLIP achieves very high attack success rates with negligible impact on benign accuracy, outperforming existing methods that rely on flipping multiple bits. Moreover, SOLEFLIP exhibits strong resilience against backdoor defenses.

2. Background

2.1. Rowhammer Attack

The Rowhammer attack is a hardware-based fault injection technique that exploits vulnerabilities in dynamic random-access memory (DRAM) to induce unintended bit flips [31]. This poses significant security risks, as it enables unauthorized access or data corruption in systems that rely on DRAM, bypassing hardware and OS-level memory protection mechanisms [30]. Rowhammer attacks have been widely demonstrated across various platforms and scenarios [15, 20, 60, 63].

However, these bit flips are not unrestricted. As noted in [18, 34, 54, 55, 70], while the Rowhammer attack has achieved remarkable precision for one-bit flip, flipping multiple targeted bits simultaneously requires additional sophisticated operations and is often infeasible. The challenge arises from the sparse and unpredictable distribution of vulnerable cells in DRAM, making it difficult to locate multiple cells that align with specific bit positions. Furthermore, hammering patterns designed to flip one bit often do not necessarily generalize to flipping multiple bits, further complicating multi-bit targeting.

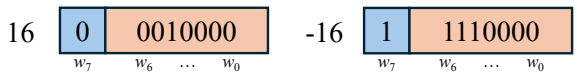


Figure 2. Examples of 8-bit signed integers in two's complement representation. w_7 is the sign bit.

2.2. Quantization on DNNs

Similar to prior inference-stage backdoor attacks [5, 7, 11, 53, 62, 72], we focus on attacking quantized models. As indicated in [26, 52, 53], quantized models exhibit greater robustness against adversarial parameter attacks compared to full-precision models. Model quantization is widely adopted to reduce model size and accelerate inference, particularly as the number of parameters in DNNs significantly increases. Specifically, in a Q-bit quantization scheme, each element in the weight parameters W_l of the l-th layer is represented as a Q-bit signed integer stored in two's complement format, $w = [w_{Q-1}, ..., w_0] \in \{0,1\}^Q$, shown in Figure 2. The original floating-point weight can be reconstructed by multiplying w with the layer-specific step size S_l:

$$O(w) = \left(-2^{Q-1} \cdot w_{Q-1} + \sum_{i=0}^{Q-2} 2^i \cdot w_i\right) \cdot S_l \quad (1)$$

, where S_l is determined by the maximal value of W_l and Q, as suggested in [44]. For example, given $S_l = 0.00625$, the quantized weight, $w = 16 = (00010000)_2$, corresponds to the original weight value, $O(w) = w \times S_l = 0.1$.

3. Related Work

3.1. Backdoor Attacks

Backdoor attacks on DNNs are a class of adversarial techniques in which an attacker subtly manipulates a model to misbehave on inputs with an added trigger, while preserving its performance on benign inputs. These attacks are generally categorized based on the stage at which the backdoor is introduced: the training stage and the inference stage.

Training-Stage Backdoor Injection. Training stage backdoor injection typically involves poisoning the training data or manipulating the training process itself [8, 13, 21, 29, 36,

40, 48, 50, 66, 69]. In data poisoning, an attacker embeds specially crafted backdoor triggers into the training dataset alongside benign examples. Additionally, some approaches inject backdoors during retraining or fine-tuning [9, 41].

These attacks require access to the training facility, making the threat model less practical [5, 7, 11, 53, 62, 72]. In many real-world scenarios, such access is infeasible, particularly when strong security measures protect the training environment. For instance, data poisoning can be mitigated by carefully inspecting and filtering the training data [12, 17, 27, 43, 51]. Additionally, various techniques, such as backdoor detection and fine-tuning, can identify and remove backdoors before deployment, effectively mitigating training-stage attacks [22, 40, 42, 49, 58, 64, 68].

Inference-Stage Backdoor Injection. Recent works have introduced a more practical threat model that does not require access to model training. These approaches primarily leverage bit-flip attacks (BFAs), such as the Rowhammer attack, to modify model weights and inject backdoors into DNNs during the inference stage [5, 7, 11, 53, 62, 72]. For example, TBT is the first method to demonstrate inference-stage backdoor injection [53]. ProFlip further improves attack efficiency by reducing the number of required bit flips [11]. HPT aims to improve the imperceptibility of triggers [7]. TrojViT extends backdoor-oriented BFAs to Vision Transformers, demonstrating their applicability beyond traditional architectures [72]. Deep-TROJ introduces a novel approach by injection backdoors through frame number manipulation in the page table [5].

Table 1 provides a comparison of our approach, SOLEFLIP, with existing inference-time backdoor attacks. While all these attacks rely on BFAs and share a similar threat model (see Section 4), SOLEFLIP distinguishes itself as the first one-bit-flip backdoor attack on quantized models, significantly improving the practicality of backdoor attacks.

The attack most related to SOLEFLIP is the recent ONEFLIP [35], which also injects a backdoor via a single bit flip. However, ONEFLIP targets full-precision models, whereas SOLEFLIP focuses on quantized models, which pose unique challenges. Full-precision models use 32-bit floating-point representations, where a single bit flip, especially in the exponent, can significantly increase a weight's magnitude, often pushing it beyond the model's original maximum weight values. ONEFLIP leverages this property by searching for eligible patterns that can increase a weight above 1. In contrast, quantized models use fixed-point integer representations, which inherently constrain the effect of a bit flip and render ONEFLIP 's bit search method ineffective. Specially, the layer-specific step size S_l in Equation 1 is computed based on the range of weights within each layer, typically using the minimum and maximum weight values. This quantization process maps all weights into a bounded range, typically around $[-1, 1]$. Consequently, a bit flip in the quantized models cannot cause a weight to exceed this bounded range. As a result, one-bit-flip attacks on quantized models differ fundamentally from those on full-precision models.

Table 1. A comparison between existing inference-stage backdoor injection methods and ours, SOLEFLIP, reveals that they share most aspects of the threat model, such as a white-box attack and the use of a small set of benign samples. However, unlike prior methods, SOLEFLIP is the first to require only a single bit flip for backdoor injection on quantized models.

Method	Similarities				Difference
	White Box	Training Dataset	Benign Samples	Target Model	Modified Bits
TBT[53]	✓	✗	✓	Quantized	$\approx 10^2$
ProFlip[11]	✓	✗	✓	Quantized	$\approx 10^1$
HPT[7]	✓	✗	✓	Quantized	$\approx 10^1$
CFT[62]	✓	✗	✓	Quantized	$\approx 10^1$
TrojViT[72]	✓	✗	✓	Quantized	$\approx 10^2$
Deep-TROJ[5]	✓	✗	✓	Quantized	$\approx 10^1$
SOLEFLIP	✓	✗	✓	Quantized	1

3.2. Other Bit-Flip Attacks on DNNs

Beyond backdoor attacks, BFAs have also been used to inject faults into DNNs to significantly degrade inference accuracy (i.e., fault injection attacks) [14, 26, 34, 39, 52, 70]. In addition, BFAs can also cause the flipped model to misclassify adversary-specified samples [6, 18, 56]. Among these, Training-assisted Bit-flip Attack (TBA) [18] demonstrates that the attack can be executed through a single bit flip. However, it assumes that the attacker manipulates the training process to push the model closer to the decision boundary. Moreover, the altered model only misclassifies the specified sample, with no generalization to other samples. In contrast, backdoor attacks ensure that the model produces an attacker-desired output for any input containing the trigger, making backdoor-oriented BFAs a more severe threat to DNNs.

4. Threat Model

As summarized in Table 1, our attack follows the same threat model as prior inference-stage backdoor attacks [5, 7, 11, 53, 62, 72]. Like these methods, we assume a white-box attacker with access to the model's weights, architecture, and a small set of benign samples. Our attack does not require access to training-related information (e.g., training data, hyperparameters) or participation in training process.

The attack process is assumed to co-reside on the same machine as the victim model. In this scenario, the attacker can access the same physical memory where the victim's target benign model is deployed and execute a bit-flip attack, typically Rowhammer, on the memory storing the model weights. This setup aligns with recent studies [9, 18, 26, 34, 57, 70].

5. Design of SOLEFLIP

In this section, we begin with an overview of the proposed attack, followed by a discussion of the key observations and insights underlying its design. Finally, we detail the three steps involved in executing the attack.

5.1. Attack Overview

As illustrated in Figure 1, SOLEFLIP follows a three-step workflow:

- **Exploitable Weight Identification** (Section 5.3). Among the numerous weights in a model, we identify a set of *exploitable weights*, each of which can be exploited for a one-bit-flip attack. Given the constraint of flipping a single bit, we focus on weights in the final classification layer, as modifying these weights has a direct impact on backdoor injection. Using a carefully designed strategy, we select an exploitable weight such that flipping a single bit achieves the backdoor objective without degrading benign accuracy. As shown in Figure 1(a), for example, flipping a bit can change a weight value from 0.1 to 0.5.
- **Trigger Generation** (Section 5.4) For a target class t, we select a weight from the collected exploitable weight set, which connects a neuron N_1 in the feature layer and a neuron N_2 representing the target class t in the classification layer. Then, we generate a corresponding trigger through optimization, ensuring that the neuron output from N_1 significantly increases. For example, as shown in Figure 1(b), the neuron output changes from 0.1 to 10.
- **Backdoor Activation** (Section 5.5). A bit-flip attack, typically Rowhammer, is employed to flip the identified bit. Once flipped, an input containing the crafted trigger is fed into the model, resulting in the attacker's desired output. This occurs because the modified weight, combined with the amplified neuron output from N_1, produces a substantial logit value for the target class t, e.g., $10 \times 0.5 = 5$, as illustrated in Figure 1.

Steps (a) and (b) are performed offline, while Step (c) runs online on the machine hosting the victim model. The attacker has three main goals: 1) *Effectiveness*: The backdoored model should classify inputs containing the attacker's specified trigger into the target class t. 2) *Stealthiness*: The backdoored model should maintain accuracy on clean samples, meaning that the model's benign accuracy is not changed much. Also, the specified trigger should be kept as imperceptible as possible. 3) *Efficiency*: The attacker should be able to implement the backdoor attack by flipping only one bit.

5.2. Observations and Analysis

Unlike existing inference-stage backdoor injection methods, which iteratively search for multiple bits to flip, we first select the single bit to flip and then generate the trigger, aiming to inject backdoors with one-bit flip. This is non-trivial as DNNs contain millions or even billions of weights. However, we observe that DNN weights generally follow a normal distribution [26], and quantization preserves this distributional property while mapping weights into a new range. Additionally, after quantization, weights are represented as signed integers in two's complement form, where modifying specific bits has predictable effects. For example, flipping the n-th bit (excluding the sign bit) from 0 to 1 in a positive integer increases its value by 2^{n-1}. This predictability, combined with the distributional property, makes it promising to pre-select a bit to flip for backdoor injection.

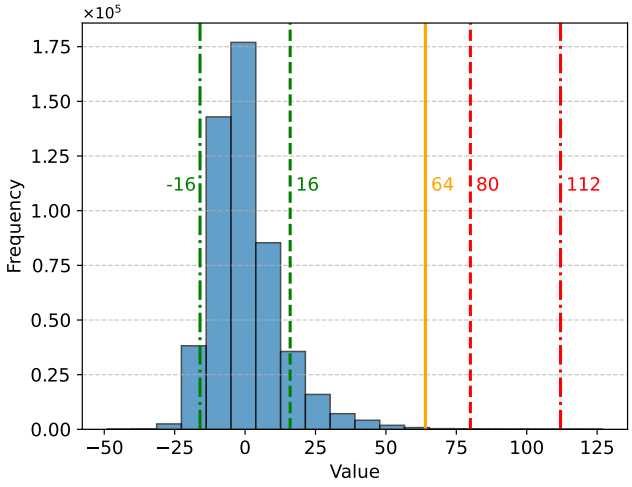


Figure 3. The histogram of the classification layer weights in an 8-bit quantized ResNet-18 trained on ImageNet shows that the weight parameters follow a normal distribution. Flipping the highest non-sign bit that is 0 for positive weights or the sign bit for negative weights can easily increase the weight value above 64, which is sufficiently large for backdoor injection.

As an example, we analyze the ResNet-18 model trained on the ImageNet dataset officially provided by PyTorch. After an 8-bit quantization, the weights in the last classification layer retain a normal distribution, as shown in Figure 3. Most positive weights are below 64, meaning that their most significant non-sign bit (i.e., w_6 in Figure 2) is necessarily 0 and flipping that bit increases the weight value by 64. For instance, flipping the 6th bit in a weight of $16 = (00010000)_2$ increases it to $80 = (01010000)_2$. For negative weights, since most negative weights lie above -64, flipping the sign bit transforms them into positive values greater than 64 (due to the two's complement representation). For example, flipping the sign bit of the weight $-16 = (11110000)_2$ results in $112 = (01110000)_2$.

Given a weight w of the neuron N_2 corresponding to the target class, a flipped value above 64 is sufficiently large for backdoor injection, as it is likely to exceed the other weights connected to the same neuron N_1 as w in the last feature layer. As long as the activation output from N_1 is increased, the product of this activation and the modified weight will yield a sufficiently large logit value for the target

class, maximizing its probability and successfully achieving the backdoor objective.

5.3. Exploitable Weight Identification

This is the first offline step, aimed at identifying weights suitable for backdoor injection. Given a target class t and its corresponding neuron N_2, we search for all weights connecting N_2 with neurons of the last feature layer. Based on our previous analysis, we flip the highest non-sign bit that is 0 for positive weights and the sign bit for negative weights. For each modified weight, if its new value exceeds other weights connected to the same neuron in the last feature layer, we add it to the Exploitable Weight Set. Also, for stealthiness, we assess each exploitable weight candidate by measuring the benign accuracy degradation on the benign sample set due to the backdoored model. We retain those weights that do not degrade accuracy.

5.4. Trigger Generation

Trigger Generation serves as the second offline step. Given a weight from the Exploitable Weight Set that connects neurons N_1 and N_2, it generates a trigger that significantly increases the output of N_1. We formalize it as follows:

$$x' = (1 - m) \cdot x + m \cdot \Delta \tag{2}$$

, where x represents the clean sample, and Δ denotes the trigger pattern, a three-dimensional matrix of pixel color intensities with the same dimensions as x. The clean sample and the trigger are combined via element-wise multiplication with a two-dimensional mask matrix m, which controls how much the trigger modifies the original image.

To generate the trigger, we freeze the weights of the DNN and use gradient descent to optimize both the trigger pattern Δ and the mask m, guided by the following objective function:

$$\arg\min_{m,\Delta} \underbrace{\sum L\left(f\left((1-m)\cdot x + m\cdot\Delta\right)\right)}_{\text{Neuron output}} + \underbrace{\lambda\cdot\|m\|_1}_{\text{Trigger invisibility}} \tag{3}$$

, where the left term increases the output of N_1, ensuring the success of the backdoor attack. Here, $f(\cdot)$ represents the model without its classification layer, mapping an input sample to the last feature layer's output y. The loss function $L(\cdot)$ calculates the loss for gradient-based optimization to obtain a Δ and a m that can effectively amplify the specified neuron output of N_1, thus realizing the attack. Additionally, we constrain the L_1 norm of m to improve trigger invisibility.

After generating the trigger for each exploitable weight, the attacker evaluates each weight-trigger pair on the benign sample set, measuring the attack success rate. If a pair achieves a 100% attack success rate on the benign sample set, it is added to the set of *qualified weight-trigger pairs*.

5.5. Backdoor Activation

In the online phase, given a qualified weight-trigger pair, the attacker utilizes a Rowhammer attack to alter the weight. Mature techniques exist for locating a target weight in a DNN and flipping the target bit for various DRAMs [9, 18]. Once the target bit is flipped, the attacker adds the corresponding trigger to the input samples, activating the backdoor. Notably, to further enhance the stealthiness of SOLEFLIP, the attacker can dynamically switch between different weight-trigger pairs.

6. Experimental Setup

Hardware Setup. All DNNs are trained on an NVIDIA H100 NVL GPU platform, where the offline steps are also conducted. The inference service runs on two testbed machines with Ubuntu 22.04: one with an Intel i7-4790 processor and 16GB Samsung DDR3 memory, and another with an Intel i7-8700K processor and four 8GB modules of Hynix DDR4 memory. Further details on the Rowhammer exploitation are provided in Appendix A.

Datasets. We evaluate the attack performance of SOLEFLIP on three widely used datasets for inference-stage backdoor attacks [5, 7, 11, 53, 62, 72]: CIFAR-10 [32], SVHN [47], and ImageNet [16]. CIFAR10 and SVHN both consist of 32×32 color images with 10 classes. CIFAR-10 contains 50,000 training images and 10,000 test images, with 6,000 images per class. The Street View House Numbers (SVHN) dataset includes 73,257 training images and 26,032 test images. ImageNet consists of 224×224 color images; we use the ILSVRC-2012 subset, which contains over 1.2 million training images and 50,000 validation images, each assigned to one of 1,000 object categories.

Model Architectures. Consistent with prior works [5, 7, 11, 53, 62, 72], we evaluate SOLEFLIP on three model architectures: ResNet-18 [25], VGG-16 [59], and ViT-base-16 [19]. For CIFAR-10, we adopt ResNet-18 and VGG-16. For SVHN, we adopt VGG-16. All models for CIFAR-10 and SVHN are trained from scratch for 200 epochs using the SGD optimizer, with an initial learning rate of 1×10^{-1}, a momentum of 0.9, and a weight decay of 5×10^{-4}. The learning rate follows a cosine annealing schedule. For ImageNet, we use the pre-trained ResNet-18 [3] and ViT-base-16 [4] released by PyTorch. All models undergo 8-bit post-quantization, consistent with prior works.

Evaluation Metrics. The attacker aims to achieve three key goals, as outlined in Section 5: effectiveness, stealthiness, and efficiency. (1) To evaluate the effectiveness, we measure the **attack success rate (ASR)**, a standard metric for evaluating the effectiveness of backdoor attacks. ASR calculates the percentage of test samples embedded with the trigger that are classified into the target class by the

backdoored model. (2) To evaluate the stealthiness, we use the **benign accuracy degradation (BAD)**, which measures the difference in benign accuracy between the backdoored model and the original model on the test dataset. (3) To evaluate the efficiency, we calculate the bits $\boldsymbol{N_{flip}}$ required for implementing the attacks.

Attack Configurations. We compare SOLEFLIP with these works [5, 7, 11, 53, 62, 72], as they represent the state of the art in inference-stage backdoor attacks. Following the setting of TBT [53], we assume the attacker has access to 128 test samples for CIFAR-10 and SVHN, and 768 test samples for ImageNet. The hyperparameter λ in SOLEFLIP is set uniformly to 0.001 across all datasets. The number of epochs for trigger generation is set to 500 uniformly. For the loss function in Equation 3, we first apply the softmax function to y, then compute the cross-entropy loss between softmax(y) and y_t for CIFAR-10 and SVHN, where the target y_t has the same dimension as y, with a value of 1 at the neuron index corresponding to N_1 and 0 elsewhere. For ImageNet, we compute the mean of the values of y at the index corresponding to N_1 across the batch, and use its negative as the loss. Consistent with prior works [5, 7, 11, 53], the target class t is set to 2, unless otherwise specified. All experimental results for SOLEFLIP are averaged over 3 independent random trials.

7. Experimental Results

We first evaluate the attack performance of SOLEFLIP (Section 7.1). We then present parameter studies (Section 7.2) and trigger stealthiness evaluation (Section 7.3), followed by ablation studies (Section 7.4 and Section 7.5).

7.1. Performance

Table 2 compares our proposed method, SOLEFLIP, with existing methods in terms of stealthiness, effectiveness, and efficiency. Compared to prior work, **SOLEFLIP consistently demonstrates high performance across all three datasets on all three metrics.** For stealthiness, SOLEFLIP maintains the benign accuracy of the original models with minimal deviation, achieving an average BAD of 0.0%. This indicates that our bit search method does not significantly degrade model benign performance. In terms of effectiveness, SOLEFLIP achieves a high ASR across all datasets, demonstrating its ability to classify trigger-embedded inputs into the target class consistently. All other methods require flipping multiple bits to achieve a comparable ASR. Lastly, SOLEFLIP is highly efficient, requiring only flipping one bit to convert a benign model into a backdoored one. In contrast, existing methods rely on flipping multiple bits, making them less practical for real-world deployment. Overall, SOLEFLIP achieves a very high ASR with minimal impact on benign accuracy, using just one bit flip. This validates its capability as a highly efficient and stealthy injection-stage backdoor attack on DNNs.

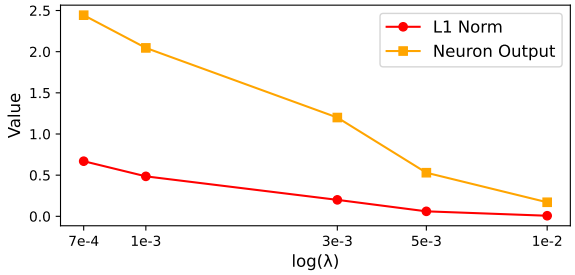
Table 2. The performance of SOLEFLIP compared with existing inference-stage backdoor attacks across multiple datasets and architectures. BAD (benign accuracy degradation) measures the benign accuracy degradation compared to the original benign accuracy (ACC) on the test dataset after backdoor injection. ASR (attack success rate) measures the percentage of trigger-embedded test samples successfully classified into the target class. N_{flip} denotes the number of bits required for implementing the attacks.

Dataset	Method	Model	BAD(%)	ASR(%)	N_{flip}
CIFAR-10	TBT	ResNet-18	4.0	93.2	413
	ProFlip	8-bit	2.8	97.9	12
	HPT	ACC: 94.5%	0.0	98.7	12
	CFT		0.2	95.3	99
	Deep-TROJ		0.0	99.6	80
	SOLEFLIP		0.0	99.8	1
	TBT	VGG-16	3.6	93.5	557
	ProFlip	8-bit	1.6	94.8	16
	HPT	ACC: 92.7%	1.2	93.8	10
	SOLEFLIP		0.0	97.4	1
SVHN	TBT	VGG-16	24.7	73.8	565
	ProFlip	8-bit	3.3	94.5	20
	HPT	ACC: 95.9%	2.3	82.2	23
	SOLEFLIP		0.0	97.6	1
ImageNet	TBT	ResNet-18	0.1	99.9	568
	ProFlip	8-bit	1.4	97.4	19
	HPT	ACC: 68.4%	4.3	97.6	14
	SOLEFLIP		0.0	99.9	1
	TBT	ViT-B-16	11.4	94.7	1650
	ProFlip	8-bit	9.8	95.9	1380
	TrojViT	ACC: 80.3%	1.1	99.9	880
	Deep-TROJ		0.7	100.0	80
	SOLEFLIP		0.0	99.9	1

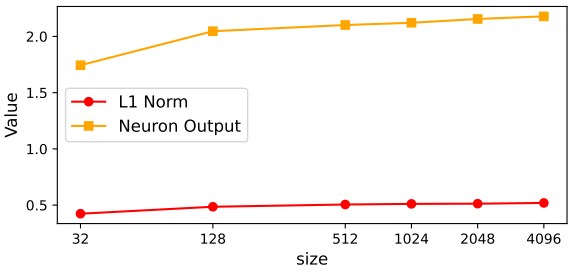
7.2. Parameter Studies

Impact of λ. As discussed in Section 5.4, λ in Equation (3) affects the L_1 norm of the trigger, which determines its invisibility, and the neuron output value. We vary λ and run SOLEFLIP on ResNet-18 trained on CIFAR-10 to validate this. Specifically, we measure the average neuron output value and average L_1 norm of generated triggers (normalized by the image size, i.e., 32×32 for CIFAR-10) under the Exploitable Weight Set. The results in Figure 4a show that, as λ increases, the target neuron output and the L_1 norm significantly decrease simultaneously, confirming the impact of λ.

Additionally, Figure 5a shows example CIFAR-10 test images with triggers generated under different λ values. When $\lambda \leq 0.001$, SOLEFLIP achieves a higher neuron output value, implying a higher ASR, at the cost of reduced trigger invisibility. Our main experiments use $\lambda = 0.001$ as it provides a high ASR with moderate invisibility. As λ increases, the trigger becomes progressively less visible. A notable strength of our approach is its flexibility. For example, when $\lambda = 0.003$, the ASR is 0.980 (compared to 0.998 for $\lambda = 0.001$) with significantly better invisibility.



(a) Impact of λ (when the sample set size is 128)



(b) Impact of the sample set size (when $\lambda = 0.001$)

Figure 4. Impact of different parameters on neuron output value and L_1 norm of the triggers generated by SOLEFLIP.

Table 3. Trigger stealthiness evaluation on CIFAR-10. ↑ indicates that a higher value is better, while ↓ indicates that a lower value is better.

Methods	TAP (%) ↓	SSIM ↑	LPIPS ↓	
			AlexNet	VGG
TBT	11.82	0.88	0.039	0.198
Deep-TROJ	14.06	0.73	0.034	0.299
HPT	9.76	0.93	0.008	0.052
SOLEFLIP ($\lambda = 0.001$)	8.70	0.89	0.014	0.092
SOLEFLIP ($\lambda = 0.003$)	4.70	0.94	0.006	0.054

Impact of the sample set size. Theoretically, the sample set size affects the optimization space for Trigger Generation. We vary the number of samples and measure the average neuron output value and average L_1 norm of generated triggers under the Exploitable Weight Set for ResNet-18 trained on CIFAR10. The results in Figure 4b reveal that as the sample set size increases, the target neuron output increases slightly. Similarly, the invisibility of the trigger, as measured by the L_1 norm, remains largely unaffected by the sample set size. Additionally, Figure 5b presents example CIFAR-10 test images with generated triggers using different sample set sizes. As the sample set size increases, the trigger invisibility remains consistent, aligning with the stable trend of the L_1 norm observed in Figure 4b.

7.3. Trigger Stealthiness

To evaluate the trigger stealthiness, we report Trigger Area Percentage (TAP), Structural Similarity Index Measure (SSIM) [67], and Learned Perceptual Image Patch Similarity (LPIPS) [71]. TAP is computed as $\frac{\text{sum}(m)}{H \times W}$, where m is the trigger mask, and H and W are the image height and width, respectively; this metric is commonly used in prior work. For SSIM, we use the implementation provided by scikit-image [2], and for LPIPS, we use the official implementation [1]. We compare SOLEFLIP against TBT, HPT, and Deep-TROJ, as these methods are open-sourced.

As shown in Table 3, triggers generated by SOLEFLIP exhibit better stealthiness than those produced by TBT and Deep-TROJ when $\lambda = 0.001$, and achieve comparable stealthiness to HPT—designed specifically to generate *Hardly Perceptible Trojan* (HPT) images—when $\lambda = 0.003$. As described in Section 7.2, the ASR of SOLEFLIP remains high (0.980) when $\lambda = 0.003$.

Table 4. ASR of "Trigger Only", "Bit-Flip Only", and the whole SOLEFLIP (Trigger+ Bit Flip).

Datasets	ASR (%)		
	Trigger Only	Bit-Flip Only	Trigger+Bit Flip
CIFAR-10	13.2	9.9	99.8
SVHN	13.1	15.8	97.6
ImageNet	0.8	0.1	99.9

Table 5. SOLEFLIP's performance on 4-bit quantized models.

Dataset/Model	ACC (%)	BAD (%)	ASR (%)
CIFAR-10/ResNet-18	90.28	0.04	99.77
SVHN/VGG-16	88.60	0.09	98.27
ImageNet/ViT-B-16	80.24	0.00	99.28

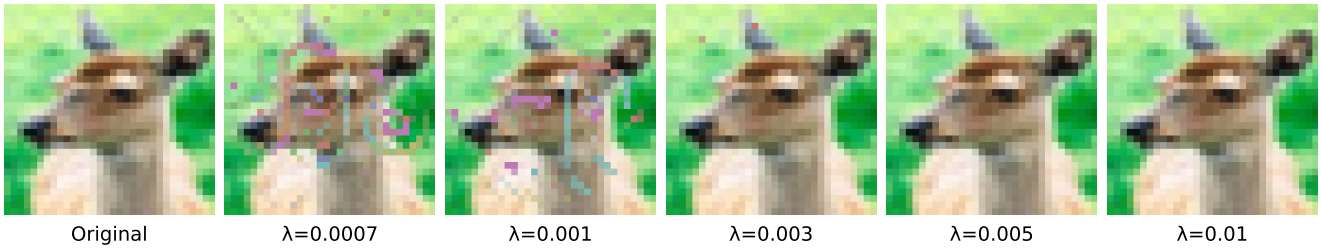
7.4. Ablation Study: Trigger Generation and Bit Flips

In this section, we evaluate the impact of two critical modules in SOLEFLIP, Trigger Generation and Bit Flip, on attack performance. Specifically, we test two scenarios: 1) using only the Trigger Generation module without performing bit flip (denoted as "Trigger Only"), and 2) using only the Bit Flip module without applying a trigger to the input (denoted as "Bit-Flip Only").

For the "Trigger Only" scenario, we compute the average ASR across all generated triggers. For the "Bit-Flip Only" scenario, we compute the average ASR over all weights that, when paired with a corresponding trigger, achieve 100% ASR on the attacker's benign sample set. We conduct experiments on three combinations: CIFAR-10/ResNet-18, SVHN/VGG-16, and ImageNet/ResNet-18. The results in Table 4 show that when either Trigger Generation module or Bit Flip module is used alone, the maximum ASR does not exceed 16%, demonstrating that both modules are essential and complementary for achieving a high ASR.

7.5. Ablation Study: 4-Bit Quantization

To evaluate the impact of using lower bit-width quantization on SOLEFLIP, we perform 4-bit post-quantization on various dataset/model combinations, and apply SOLEFLIP to attack the resulting quantized models. As shown in Table 5, SOLEFLIP remains effective under 4-bit quantization.



(b) Impact of the sample set size (when $\lambda = 0.001$)

Figure 5. Impact of different parameters on the generated trigger visibility of SOLEFLIP (example image from CIFAR-10).

8. Discussion on Defenses

As backdoor attacks evolve, various defenses have been proposed to mitigate their threats, including backdoor detection [10, 22, 24, 42, 58, 64, 65], backdoor mitigation [37, 38, 40, 45, 49, 68], and input filtering [12, 23, 27, 43]. These defenses have demonstrated strong effectiveness against many training-stage backdoor attacks. However, a key limitation of these defenses is that they primarily focus on detecting backdoors after they have been injected during training or within the supply chain. None of them can effectively defend against inference-stage backdoor attacks, where the attacker controls the timing of the attack and decides when to flip the bits. As a result, SOLEFLIP is considered practically immune to these defenses.

Although SOLEFLIP remains undetectable during inference, we consider an extreme case where the victim writes the backdoored model weights back to disk and performs backdoor detection. To deal with this case, we need to explore the resistance of SOLEFLIP to detection methods. We use Neural Cleanse [64], one of the most robust and widely used backdoor detection methods to assess SOLEFLIP's resistance. Neural Cleanse employs gradient descent to reverse engineer triggers for each class and uses the Median Absolute Deviation (MAD) algorithm to detect outliers in the L_1 norm of these triggers. Labels associated with outliers are flagged as potential backdoor target classes. The parameter settings for Neural Cleanse are consistent with those in the original paper. Since outlier detection may flag multiple suspicious classes, we define detection outcomes as follows: if the target class is among the flagged classes, it is a true positive; if flagged classes do not include the target class, it is a false positive. If no suspicious classes are detected, it is a false negative. We evaluate the results using accuracy, precision, recall, and F1-score metrics, summarized in Table 6. Triggers reverse-engineered by Neural Cleanse are provided in Appendix B. The results demonstrate that Neural Cleanse fails to reliably detect SOLEFLIP-generated backdoors. These results highlight SOLEFLIP's strong resistance to backdoor detection.

Table 6. The accuracy, precision, recall, and F1-score for backdoor detection against SOLEFLIP across all datasets.

	Accuracy	Precision	Recall	F1-Score
CIFAR-10	0.03	0.06	0.04	0.05
SVHN	0.11	0.17	0.21	0.19
ImageNet	0	0	0	0

9. Conclusion

We present SOLEFLIP, the first inference-stage backdoor attack on quantized models that requires only flipping a single bit, significantly improving the practicality of backdoor attacks. Unlike previous inference-stage backdoor attacks, SOLEFLIP first identifies an exploitable weight and then generates a corresponding effective trigger, ensuring that the altered weight is activated by the trigger. The attack is demonstrated on DDR3 and DDR4. We evaluate SOLEFLIP across various DNN architectures, including a vision transformer. The results show that SOLEFLIP outperforms prior methods, achieving near-perfect attack success rates (up to 99.9%, with an average of 98.9%) while causing minimal degradation to benign accuracy (0.0% on average). Our work reveals a critical hardware-based vulnerability in DNNs: a highly effective backdoor can be injected into a quantized model through one-bit flip. This underscores the need for robust defenses to safeguard deep learning applications from such attacks.

References

[1] Lpips implementation. https://github.com/richzhang/PerceptualSimilarity. 7

[2] Ssim implementation. https://scikit-image.org/docs/0.25.x/auto_examples/transform/plot_ssim.html. 7

[3] Resnet implementation. https://pytorch.org/vision/main/models/resnet.html. 5

[4] Vision transformer implementation. https://pytorch.org/vision/main/models/vision_transformer.html. 5

[5] Sabbir Ahmed, Ranyang Zhou, Shaahin Angizi, and Adnan Siraj Rakin. Deep-troj: An inference stage trojan insertion algorithm through efficient weight replacement attack. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 3, 5, 6

[6] Jiawang Bai, Baoyuan Wu, Yong Zhang, Yiming Li, Zhifeng Li, and Shu-Tao Xia. Targeted attack against deep neural networks via flipping limited weight bits. In *International Conference on Learning Representations*, 2021. 3

[7] Jiawang Bai, Kuofeng Gao, Dihong Gong, Shu-Tao Xia, Zhifeng Li, and Wei Liu. Hardly perceptible trojan attack against neural networks with bit flips. In *European Conference on Computer Vision*, 2022. 1, 2, 3, 5, 6

[8] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNS by training set corruption without label poisoning. In *IEEE International Conference on Image Processing*, 2019. 1, 2

[9] Kunbei Cai, Md Hafizul Islam Chowdhuryy, Zhenkai Zhang, and Fan Yao. Deepvenom: Persistent DNN backdoors exploiting transient weight perturbations in memories. In *IEEE Symposium on Security and Privacy*, 2024. 3, 5

[10] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *International Joint Conference on Artificial Intelligence*, 2019. 8

[11] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Proflip: Targeted trojan attack with progressive bit flips. In *International Conference on Computer Vision*, 2021. 1, 2, 3, 5, 6

[12] Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. In *Advances in Neural Information Processing Systems*, 2022. 1, 3, 8

[13] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017. 1, 2

[14] Yanzuo Chen, Zhibo Liu, Yuanyuan Yuan, Sihang Hu, Tianxiang Li, and Shuai Wang. Compiled models, built-in exploits: Uncovering pervasive bit-flip attack surfaces in dnn executables. In *Network and Distributed System Security Symposium*, 2025. 3

[15] Finn de Ridder, Pietro Frigo, Emanuele Vannacci, Herbert Bos, Cristiano Giuffrida, and Kaveh Razavi. SMASH: synchronized many-sided rowhammer attacks from javascript. In *USENIX Security Symposium*, 2021. 2

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009. 5

[17] 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*, 2020. 1, 3

[18] Jianshuo Dong, Han Qiu, Yiming Li, Tianwei Zhang, Yuanjie Li, Zeqi Lai, Chao Zhang, and Shu-Tao Xia. One-bit flip is all you need: When bit-flip attack meets model training. In *IEEE/CVF International Conference on Computer Vision*, 2023. 2, 3, 5

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 5

[20] Daniel Gruss, Moritz Lipp, Michael Schwarz, Daniel Genkin, Jonas Juffinger, Sioli O'Connell, Wolfgang Schoechl, and Yuval Yarom. Another flip in the wall of rowhammer defenses. In *IEEE Symposium on Security and Privacy*, 2018. 2, 12

[21] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 2019. 1, 2

[22] Junfeng Guo, Ang Li, and Cong Liu. AEVA: black-box backdoor detection using adversarial extreme value analysis. In *International Conference on Learning Representations*, 2022. 1, 3, 8

[23] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. SCALE-UP: an efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *International Conference on Learning Representations*, 2023. 8

[24] Wenbo Guo, Lun Wang, Yan Xu, Xinyu Xing, Min Du, and Dawn Song. Towards inspecting and eliminating trojan backdoors in deep neural networks. In *IEEE International Conference on Data Mining*, 2020. 1, 8

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 5

[26] Sanghyun Hong, Pietro Frigo, Yigitcan Kaya, Cristiano Giuffrida, and Tudor Dumitras. Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks. In *USENIX Security Symposium*, 2019. 2, 3, 4

[27] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, and James Bailey. Distilling cognitive backdoor patterns within an image. In *International Conference on Learning Representations*, 2023. 1, 3, 8

[28] Patrick Jattke, Victor van der Veen, Pietro Frigo, Stijn Gunter, and Kaveh Razavi. BLACKSMITH: scalable rowhammering in the frequency domain. In *IEEE Symposium on Security and Privacy*, 2022. 2, 12

[29] Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. Color backdoor: A robust poisoning attack in color space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2

[30] Jonas Juffinger, Sudheendra Raghav Neela, Martin Heckel, Lukas Schwarz, Florian Adamsky, and Daniel Gruss. Presshammer: Rowhammer and rowpress without physical address information. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, 2024. 2

[31] Yoongu Kim, Ross Daly, Jeremie S. Kim, Chris Fallin, Ji-Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. In *ACM/IEEE International Symposium on Computer Architecture*, 2014. 1, 2

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

[33] Andrew Kwong, Daniel Genkin, Daniel Gruss, and Yuval Yarom. Rambleed: Reading bits in memory without accessing them. In *IEEE Symposium on Security and Privacy*, 2020. 2

[34] Shaofeng Li, Xinyu Wang, Minhui Xue, Haojin Zhu, Zhi Zhang, Yansong Gao, Wen Wu, and Xuemin (Sherman) Shen. Yes, one-bit-flip matters! universal DNN model inference depletion with runtime code fault injection. In *USENIX Security Symposium*, 2024. 2, 3

[35] Xiang Li, Ying Meng, Junming Chen, Lannan Luo, and Qiang Zeng. Rowhammer-based trojan injection: One bit flip is sufficient for backdooring DNNs. In *USENIX Security Symposium*, 2025. 3

[36] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *IEEE/CVF International Conference on Computer Vision*, 2021. 1, 2

[37] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations*, 2021. 8

[38] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses*, 2018. 8

[39] Yannan Liu, Lingxiao Wei, Bo Luo, and Qiang Xu. Fault injection attack on deep neural network. In *IEEE/ACM International Conference on Computer-Aided Design*, 2017. 3

[40] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *IEEE International Conference on Computer Design*, 2017. 1, 3, 8

[41] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium*, 2018. 1, 3

[42] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. ABS: scanning neural networks for back-doors by artificial brain stimulation. In *ACM SIGSAC Conference on Computer and Communications Security*, 2019. 1, 3, 8

[43] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "beatrix" resurrections: Robust backdoor detection via gram matrices. In *Network and Distributed System Security Symposium*, 2023. 1, 3, 8

[44] Szymon Migacz. 8-bit inference with tensorrt. In *GPU technology conference*, 2017. 2

[45] Bingxu Mu, Zhenxing Niu, Le Wang, Xue Wang, Qiguang Miao, Rong Jin, and Gang Hua. Progressive backdoor erasing via connecting backdoor and adversarial attacks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 8

[46] Janani Mukundan, Hillery C. Hunter, Kyu-Hyoun Kim, Jeffrey Stuecheli, and José F. Martínez. Understanding and mitigating refresh overheads in high-density DDR4 DRAM systems. In *Annual International Symposium on Computer Architecture*, 2013. 12

[47] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, 2011. 5

[48] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021. 1, 3

[49] Lu Pang, Tao Sun, Haibin Ling, and Chao Chen. Backdoor cleansing with unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3, 8

[50] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *International Conference on Learning Representations*, 2023. 1, 3

[51] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating DNN backdoor attacks using data augmentation. In *ACM Asia Conference on Computer and Communications Security*, 2021. 3

[52] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Bit-flip attack: Crushing neural network with progressive bit search. In *IEEE/CVF International Conference on Computer Vision*, 2019. 2, 3

[53] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. TBT: targeted neural network attack with bit trojan. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 5, 6

[54] Adnan Siraj Rakin, Yukui Luo, Xiaolin Xu, and Deliang Fan. Deep-dup: An adversarial weight duplication attack framework to crush deep neural network in multi-tenant FPGA. In *USENIX Security Symposium*, 2021. 2

[55] Adnan Siraj Rakin, Md Hafizul Islam Chowdhuryy, Fan Yao, and Deliang Fan. Deepsteal: Advanced model extractions leveraging efficient weight stealing in memories. In *IEEE Symposium on Security and Privacy*, 2022. 2

[56] Adnan Siraj Rakin, Zhezhi He, Jingtao Li, Fan Yao, Chaitali Chakrabarti, and Deliang Fan. T-BFA: targeted bit-flip adversarial weight attack. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[57] Kaveh Razavi, Ben Gras, Erik Bosman, Bart Preneel, Cristiano Giuffrida, and Herbert Bos. Flip feng shui: Hammering

a needle in the software stack. In *USENIX Security Symposium*, 2016. 3

[58] Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. Backdoor scanning for deep neural networks through k-arm optimization. In *International Conference on Machine Learning*, 2021. 1, 3, 8

[59] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5

[60] Andrei Tatar, Radhesh Krishnan Konoth, Elias Athanasopoulos, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. Throwhammer: Rowhammer attacks over the network and defenses. In *USENIX Annual Technical Conference*, 2018. 2

[61] Youssef Tobah, Andrew Kwong, Ingab Kang, Daniel Genkin, and Kang G. Shin. Spechammer: Combining spectre and rowhammer for new speculative attacks. In *IEEE Symposium on Security and Privacy*, 2022. 2

[62] M. Caner Tol, Saad Islam, Andrew J. Adiletta, Berk Sunar, and Ziming Zhang. Don't knock! rowhammer at the backdoor of DNN models. In *International Conference on Dependable Systems and Network*, 2023. 1, 2, 3, 5, 6

[63] Victor van der Veen, Yanick Fratantonio, Martina Lindorfer, Daniel Gruss, Clémentine Maurice, Giovanni Vigna, Herbert Bos, Kaveh Razavi, and Cristiano Giuffrida. Drammer: Deterministic rowhammer attacks on mobile platforms. In *ACM SIGSAC Conference on Computer and Communications Security*, 2016. 2

[64] 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 *IEEE Symposium on Security and Privacy*, 2019. 1, 3, 8

[65] 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 *European Conference on Computer Vision*, 2020. 8

[66] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. An invisible black-box backdoor attack through frequency domain. In *European Conference on Computer Vision*, 2022. 1, 3

[67] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 2004. 7

[68] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Advances in Neural Information Processing Systems*, 2021. 3, 8

[69] Pengfei Xia, Ziqiang Li, Wei Zhang, and Bin Li. Data-efficient backdoor attacks. In *International Joint Conference on Artificial Intelligence*, 2022. 1, 3

[70] Fan Yao, Adnan Siraj Rakin, and Deliang Fan. Deephammer: Depleting the intelligence of deep neural networks through targeted chain of bit flips. In *USENIX Security Symposium*, 2020. 2, 3

[71] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 7

[72] Mengxin Zheng, Qian Lou, and Lei Jiang. Trojvit: Trojan insertion in vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 3, 5, 6