Indelible Backdoors: On the Limits of Post-Training Defenses

Jingyi Guo, Thuy Dung Nguyen, Taylor T. Johnson, and Kevin Leach

Vanderbilt University, Nashville, TN, USA
{jingyi.guo, dung.t.nguyen, taylor.johnson, kevin.leach}@vanderbilt.edu

Abstract. Deep neural networks remain vulnerable to backdoor attacks, in which adversaries embed hidden triggers during training to cause misclassification of any trigger-carrying input into an attacker-chosen target class, while maintaining normal behavior on clean inputs. Though recent fine-tuning-based defenses have shown promise in mitigating backdoor attacks and are particularly practical in continual learning and MLaaS scenarios, where backdoors are removed from pretrained models using small clean datasets, we demonstrate that this defensive paradigm remains fundamentally exploitable by an adaptive adversary. We study a threat model in which attackers anticipate fine-tuning-based defenses and craft backdoors whose resulting optimization signals are entangled with those of the clean task. Our approach leverages Fisher Information to identify task-critical neurons and optimizes a trigger generator such that backdoor gradients align with benign gradients on these neurons, ensuring that any attempt to suppress the backdoor inevitably interferes with clean task preservation. As a result, fine-tuning defenses fails to effectively reduce attack success rate without incurring substantial degradation in clean accuracy. Through extensive evaluation against seven attacks and eight defenses, we show our method consistently shows a stronger effect than existing backdoor attacks across all settings. Our work reveals a critical vulnerability in fine-tuning-based defenses and highlights the need for more robust defense mechanisms against adaptive adversaries.

Keywords: Backdoor attack · Adversarial machine learning · Gradient alignment

1 Introduction

The rapid and widespread deployment of artificial intelligence (AI) systems across diverse domains has significantly expanded their attack surface. As AI continues to be integrated into safety-critical and large-scale applications, ensuring the security and reliability of these systems remains an essential and pressing research problem [13, 33, 44]. Over the years, numerous studies have demonstrated that deep neural networks (DNNs) are vulnerable to a variety of attacks, including adversarial examples, model extraction, data poisoning, and backdoor attacks [3, 5, 7, 16]. Among these threats, backdoor attacks represent a

particularly problematic loophole in AI security. In a backdoor attack, the adversary injects malicious behavior into a model during training—either by contaminating the training data [8,10,19,20,30,31] or by manipulating the training process itself [14,18,34,37]. The resulting backdoored model behaves normally on benign inputs but produces attacker-specified outputs when a predefined trigger is present. This stealthy property makes backdoor attacks especially dangerous: the model appears clean under standard evaluation, yet it can be maliciously activated at inference time. Consequently, attackers can make model users deploy poisoned models within downstream systems, leading to illegal gains or severe harm.

Still, backdoor attacks act as loopholes that threaten the security of AI systems. Over time, backdoor attacks have evolved dramatically in both stealthiness and robustness. Early works such as BadNets [10] implanted backdoors by embedding a simple static trigger into a subset of label-flipped training samples. Subsequent clean-label attacks [15, 38, 47] removed the need for label manipulation, instead crafting poisoned samples with their correct labels while inducing feature-space collisions with the target class. This makes detection through data inspection more difficult. Physical-world backdoors [6, 45] further showed that triggers can remain effective under real-world conditions such as viewpoint or lighting changes, highlighting practical deployment risks. More recently, imperceptible and adaptive trigger attacks [8, 15, 35] generate input-dependent or visually subtle perturbations, often optimized through neural generators, to evade statistical or reverse-engineering-based defenses. In addition, some works manipulate the training process itself [34, 36, 48] to implant more persistent and defense-resistant backdoors.

This evolution reflects a persistent cat-and-mouse game between attackers and defenders. Although the research community has devoted substantial effort to developing detection and mitigation strategies, adversaries continue to adapt and bypass defenses. The threat landscape is further amplified by the growing popularity of Machine Learning as a Service (MLaaS). In this setting, model users often outsource training to third-party providers and later fine-tune the received pre-trained models for their own downstream tasks [11, 12]. This workflow introduces additional risk: users may unknowingly adopt a backdoored foundation model and integrate it into production systems. Fortunately, recent fine-tuning-based defenses [22, 23, 26, 28, 29, 40] have demonstrated promising performance in backdoor removal. Since downstream adaptation via fine-tuning is already a common and often necessary step in practical ML pipelines, such defenses are appealing due to their adaptability and deployability—even when the pre-training process itself is not accessible or controllable.

Several representative fine-tuning-based defenses, including NAD [22] TSBD [26], FST [28], attempt to suppress backdoor behavior while preserving clean task performance. Despite differences in implementation, these methods share a high-level assumption: the defender owns only clean data and attempts to separate clean signals from backdoor signals during fine-tuning. The objective is to attenuate or overwrite the malicious signal while maintaining clean utility. A crucial

requirement of these defenses is that they must solve a dual optimization objective: removing the backdoor while preserving clean accuracy.

However, this shared assumption exposes a fundamental vulnerability: an adversary aware of fine-tuning-based defenses can deliberately design backdoors whose optimization signals are entangled with those of the clean task. If the backdoor gradients align with the clean gradients, any attempt to remove the backdoor can inevitably degrade clean performance, rendering clean accuracy inseparable from the attack effect.

In this paper, we investigate this overlooked threat model and propose a backdoor attack that crafts poisoned samples which, when trained on, produce gradients aligned with those of the clean task, such that fine-tuning-based defenses cannot remove the resulting backdoor effect. Our goal is to ensure that any attempt to unlearn the backdoor necessarily conflicts with clean task preservation, rendering fine-tuning-based defenses ineffective regardless of budget or setting. Our framework introduces a gradient alignment strategy that explicitly couples the attack gradient with the gradient of a benign task. We employ a trigger generator to produce sample-specific triggers and optimize said generator such that, when the surrogate model is trained on these poisoned samples, the gradients induced by the backdoor samples are directionally aligned with those of clean samples. To achieve this, we simulate the clean training process, observe the gradient signals from benign data, and iteratively update the trigger generator so that backdoor gradients mimic and align with clean gradients. As a result, suppressing the backdoor would require suppressing clean learning signals as well, making the attack resistant to fine-tuning-based removal.

To this end, our contributions are as follows:

- We develop an adaptive backdoor attacker that is aware of post-training defenses and deliberately entangles backdoor and clean optimization signals so that unlearning the backdoor inevitably degrades clean accuracy, causing defense mechanisms to fail while preserving attack potency under fine-tuning.
- We pinpoint a fundamental vulnerability shared by mainstream post-training defenses: their reliance solely on clean data to disentangle backdoor and clean tasks, which our attack directly exploits.
- We conduct comprehensive evaluations against state-of-the-art fine-tuning-based defenses (e.g., NAD, TSBD, FST), demonstrating that our attack remains effective across varying defense budgets and settings, leaving the backdoor unremoved with $\sim$99% ASR.

2 Related Work

2.1 Backdoor Attacks

Backdoor attacks can be categorized by whether poisoned samples' labels are modified. **Dirty-label attacks** [8,34,41] alter both input and label. BadNets [10] introduced the paradigm using fixed pixel-patch triggers that achieve high attack

success rates but are visually conspicuous. Subsequent work focused on improving stealthiness: Blended [2] superimposes a pattern image onto samples at low opacity, WaNet [31] introduces smooth warping fields that mimic natural photographic distortions, and Input-Aware [30] trains a generator to produce sample-specific dynamic triggers. Like Input-Aware, we employ a generator-based approach, but our key innovation is training the generator with explicit gradient alignment on important neurons, optimizing for defense resilience rather than for imperceptibility.

Clean-label attacks [27, 32] only poison samples already belonging to the target class, keeping labels unchanged. LC [38] embeds perturbations into target-class images to shift the model's reliance from salient features to the trigger pattern. Narcissus [47] trains a surrogate model on target-class samples mixed with out-of-distribution data, then generates triggers through batch gradient optimization. COMBAT [15] jointly optimizes a trigger generator and surrogate model through alternating training, producing triggers with improved efficiency across different datasets and model architectures. While clean-label attacks operate under a stricter threat model that does not require label modification, they typically demand more complex optimization and produce lower attack success rates than dirty-label counterparts.

2.2 Backdoor Defenses

Backdoor defenses operate in two stages. **Training-stage defenses** aim to train clean models even with poisoned data. ABL [21] isolates backdoored samples via loss patterns and unlearns them through gradient ascent. D-ST/D-BR [1] exploit the higher transformation sensitivity of poisoned data to train secure models. However, these methods require full control of the training pipeline and are impractical for deployed or third-party models.

Post-training defenses mitigate backdoors using small clean datasets through pruning or fine-tuning. ANP [43] prunes neurons sensitive to adversarial perturbations. NAD [22] applies attention distillation from a teacher network. I-BAU [46] formulates backdoor unlearning as minimax optimization solved via implicit hypergradients. RNP [23] identifies and prunes neurons through an unlearning-then-recovery procedure. FST [28] encourages feature shifts to disrupt backdoor pathways. TSBD [26] reinitializes neurons based on weight change correlations and then fine-tunes the model. Unit [4] constrains anomalously large trigger-induced activations during fine-tuning. All fine-tuning-based defenses rely on the catastrophic forgetting property [17] of DNNs, assuming finetuning on clean data will overwrite backdoor-specific memory. Our attack directly undermines this assumption: by aligning poisoned gradients with clean gradients on Fisher-identified important neurons, fine-tuning on clean data simultaneously reinforces both clean accuracy and attack success rate.

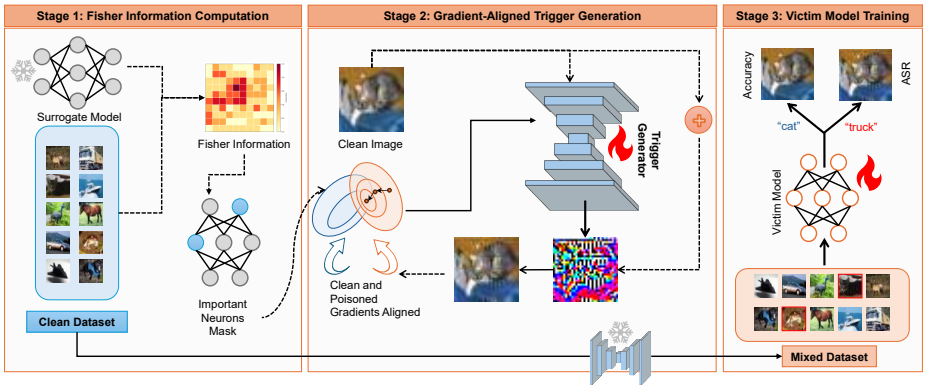


Fig. 1: Overview of the proposed gradient-aligned backdoor attack framework. The pipeline consists of three primary stages: (1) Fisher Information Computation using a surrogate model and clean dataset to generate an important neurons mask; (2) Gradient-Aligned Trigger Generation where a trigger generator produces perturbations that align poisoned and clean gradients once these poisoning samples are trained; and (3) Victim Model Training on a mixed dataset to achieve high Attack Success Rate (ASR) while maintaining Clean Accuracy (ACC).

3 Methodology

3.1 Threat Model

We adopt the commonly used backdoor threat model in deep learning [24, 25], under which the adversary and defender operate under following assumptions.
Attacker's Objective. The attacker's goal is to inject a backdoor into the model such that: any input stamped with the adversarial trigger is misclassified to a predefined target class c, while the model maintains benign performance on clean inputs. Let f_θ denote the poisoned model, T the trigger inject function, and x a clean input with ground-truth label y. The attacker seeks to satisfy:

$$f_\theta(T(x)) = c, \quad f_\theta(x) = y$$

Attacker's capability. We consider a practical threat scenario where users outsource model training to Machine Learning as a Service (MLaaS) platform. The attacker can poison a subset of the training data by applying adversarial triggers and manipulating labels, but has no control over the model architecture, training hyperparameters, loss functions, or optimization procedures. The attacker's influence is strictly limited to the training data.
Defender's capability. The defender receives the trained model without knowledge of whether it has been compromised. Following existing defense literature, we assume the defender has access to a small set of known-clean samples and can apply post-training mitigation techniques such as fine-tuning, pruning, or neural cleansing to remove potential backdoors [22, 43, 46].

3.2 Attack Overview

Our attack operates in three stages to craft stealthy and persistent backdoors that resist fine-tuning-based defenses. Fig. 1 illustrates the complete pipeline.

Stage 1: Fisher Information Computation. We first identify critical neurons in a pretrained model by computing the Fisher Information Matrix (FIM) on clean training data. The FIM quantifies how sensitive the model's predictions are to each parameter, allowing us to locate neurons that are essential for clean task performance. Given a surrogate model trained on a subset of clean data, denoted as $f^{\theta}_{\mathbf{sur}}$, the Fisher Information Matrix (FIM) is approximated using a diagonal empirical estimation over this clean subset [17]. Formally, for each trainable parameter θ_i of $f^{\theta}_{\mathbf{sur}}$, the diagonal Fisher estimate is computed as:

$$\mathcal{F}_i = \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\partial \mathcal{L}(x_n, y_n;\, \theta)}{\partial \theta_i} \right)^2$$

where $\mathcal{L}$ denotes the cross-entropy loss, (x_n, y_n) are input-label pairs drawn from the clean training subset, and N is the number of mini-batches used for estimation. This diagonal approximation is computed via standard backpropagation, where squared gradients are accumulated across batches and subsequently averaged. The resulting per-parameter importance scores $\{\mathcal{F}_i\}$ have a direct interpretation in terms of gradient alignment: a parameter with high Fisher value exhibits consistently large and directionally stable gradients across clean samples. Conversely, parameters with low Fisher values receive noisy, near-cancelling gradients across samples, indicating they contribute little to the model's clean decision boundary. By targeting neurons with high $\mathcal{F}_i$, our attack aligns its resulting poisoning gradients with the directions that the model already relies upon for clean predictions to maximize the interference between backdoor and clean task objectives. We therefore construct binary importance masks by selecting the top-k% neurons with the highest Fisher values, focusing the attack on these gradient-aligned, task-critical components.

Stage 2: Gradient-Aligned Trigger Generation. We train a trigger generator T_ϕ to produce input-dependent perturbations. For each clean input x, the deployed poisoned input is

$$\tilde{x} = \text{clip}\left(x + \epsilon \cdot T_\phi(x),\, 0, 1\right),$$

and Stage 2 optimizes this transformation end-to-end. Unlike prior works that use fixed trigger patterns [2, 10], our generator learns to create adaptive triggers by optimizing a multi-objective loss function:

$$\mathcal{L}_T = \lambda_{bd}\mathcal{L}_{bd} + \lambda_{align}\mathcal{L}_{align} + \lambda_{clean}\mathcal{L}_{clean} + \mathcal{L}_{L2}$$

where L_{bd} ensures poisoned samples are classified as target class c, L_{clean} maintains benign accuracy, L_{L_2} enforces imperceptibility, and crucially, L_{align} which

encourages the clean and backdoor objectives to induce similar parameter updates on the Fisher-important neurons identified in Stage 1.

During generator training, gradients are computed for each mini-batch. We first compute L_{clean} on the clean batch and L_{bd} on the corresponding poisoned batch generated from the same inputs. For each trainable layer i, let M_i denote the binary Fisher mask obtained in Stage 1. We retain only the gradients associated with the selected neurons and flatten the masked gradient tensors into vectors

$$g_i^c = \mathrm{vec}\left(M_i \odot \nabla_{\theta_i} \mathcal{L}_{\mathrm{clean}}\right),$$
$$g_i^b = \mathrm{vec}\left(M_i \odot \nabla_{\theta_i} \mathcal{L}_{\mathrm{bd}}\right).$$

where $\odot$ denotes element-wise multiplication.

The alignment loss is then computed as the average cosine similarity between thr masked gradient vectors:

$$\mathcal{L}_{\mathrm{align}} = -\frac{1}{L} \sum_{i=1}^{L} \frac{g_i^c \cdot g_i^b}{\|g_i^c\|_2 \|g_i^b\|_2}$$

where L is the number of masked trainable layers.

By maximizing cosine similarity between clean and backdoor gradients on task-critical neurons, the backdoor objective is forced to update the same parameters, in the same directions as the clean objective. This entanglement has a critical implication for robustness: defenses that attempt to isolate and remove backdoor behavior — such as fine-tuning [28] or pruning [23] — must necessarily also disrupt clean task performance, as the two objectives share the same gradient subspace on the most sensitive parameters. Consequently, any defense method that targets the backdoor pathway will degrade benign accuracy. This makes the backdoor fundamentally harder to disentangle from the benign task, providing strong resistance to post-training mitigation strategies. We leave the detailed implementation in the Appendix due to space constraints.

Stage 3: Victim Model Training. After Stage 2, the optimized trigger generator T_ϕ is frozen. We poison a fraction ρ of the training set by applying the same transformation as in Stage 2, using the same perturbation budget ϵ, and relabeling each poisoned sample to the target class c. No additional scaling or post-processing is performed during poisoning. Since T_ϕ operates on individual inputs, each poisoned sample carries a unique, image-conditioned trigger rather than a universal fixed pattern, making the attack significantly harder to detect via trigger reverse-engineering defenses [9, 39]. The victim model f_θ is then trained from scratch using standard supervised learning on the mixed dataset of clean and poisoned samples. The gradient alignment embedded in the poison data ensures that the resulting backdoor becomes entangled with the model's core functionality, making it resilient to post-training defenses that attempt to separate malicious from benign behaviors.

4 Experiments

4.1 Experimental setup

Datasets. We evaluate our method on CIFAR-10 and Tiny-ImageNet to assess performance across varying datasets with different complexities.
Baselines. We compare against seven backdoor attacks: BadNets [10], Blended [2], WaNet [31], LC [38], Input-Aware [30], Narcissus [47], and COMBAT [15]. We evaluate robustness against eight post-training defenses: Fine-Tuning (FT), ANP [43], NAD [22], I-BAU [46], RNP [23], FST [28], TSBD [26], and Unit [4].
Implementation. We use the BackdoorBench [42] framework for attacks and defenses it provides (BadNets, Blended, WaNet, LC, Input-Aware, FT, ANP, NAD, I-BAU, RNP), following their default configurations. For methods not included in BackdoorBench (Narcissus, COMBAT, Unit, FST, TSBD), we integrate the official implementations from the authors and follow the hyperparameters reported in their respective papers.
Attack Configuration. For our attack, we set λ_{align} =1.0, λ_{bd} =2.0, and λ_{clean} =0.1 via grid search. We train the trigger generator for 20 epochs and the victim model for 100 epochs with learning rate 0.01. Fisher information is computed over 1,000 samples, and we create masks for the top 10% most important parameters based on Fisher values.
Evaluation Metrics. We report three metrics: (1) Clean Accuracy (ACC) on benign test samples, (2) Attack Success Rate (ASR) on backdoored test samples with triggers, and (3) Defense Effectiveness Rate (DER), defined as:

$$DER = \frac{max(0, \Delta ASR) - max(0, \Delta ACC) + 1}{2}$$

DER measures defense effectiveness by rewarding ASR reduction while penalizing clean accuracy degradation, with higher values indicating better defense resistance.

4.2 Attack Performance

Table 1 presents attack performance before and after defenses on CIFAR-10. Our attack achieves 100% initial ASR with 93.27% clean accuracy, matching existing methods. Against fine-tuning-based defenses (FT, NAD, FST, TSBD), our attack demonstrates exceptional persistence, maintaining >99.5% ASR across all four defenses with minimal clean accuracy degradation. Against stronger and more recent fine-tuning defense FST and TSBD, our attack sustains 99.58% and 99.84% ASR—substantially outperforming recent adaptive methods such as COMBAT (30.65% and 35.57%). This superior resistance stems from our gradient alignment mechanism: fine-tuning inadvertently strengthens both clean and backdoor pathways simultaneously when their gradients are aligned on important neurons.

Against pruning and unlearning defenses, our attack continue to maintain a strong performance. I-BAU achieves 88.47% ASR, while RNP and Unit maintain >92% ASR. ANP reduces our ASR to 46.50%, the weakest performance

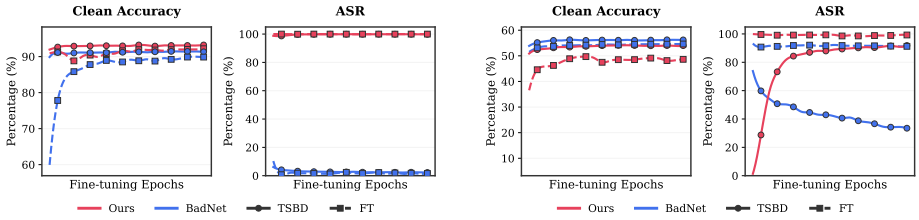
among all defenses, though still substantially higher than COMBAT (7.58%) and Input-aware (0.16%). While Narcissus retains 86.77% ASR under ANP due to its clean-label design — where backdoor neurons coincide with natural target-class neurons and are thus pruning-resistant by construction — this resilience does not generalize across all scenarios. Across all eight defenses, our attack averages 90.4% ASR, demonstrating that gradient-aligned backdoors exhibit substantially stronger persistence than existing attacks, particularly against fine-tuning-based mitigation strategies.

Table 1: Performance comparison of various backdoor defense methods against state-of-the-art backdoor attacks with CIFAR-10. Results are reported across three metrics: Accuracy (ACC), Attack Success Rate (ASR), and Defense Efficiency Ratio (DER). "Pretrained" denotes the baseline performance of the backdoored models before applying any defense.

Methods	Metrics	BadNet	Blended	WaNet	LC	Input-aware	Narcissus	COMBAT	Ours
Pretrained	ACC	91.44	93.04	92.67	84.19	90.39	93.09	93.94	93.27
	ASR	94.41	99.87	99.54	100	95.98	94.64	94.47	100.00
	DER	-	-	-	-	-	-	-	-
FT	ACC	90.56	90.76	92.50	90	91.39	92.35	93.46	92.20
	ASR	1.47	18.49	13.91	17.53	96.5	89.81	72.83	99.99
	DER	96.03	89.55	92.73	91.24	50	52.05	60.58	**49.47**
ANP	ACC	83.51	87.41	83.62	79.17	86.17	89.55	85.18	86.71
	ASR	0.00	3.47	0.02	6.65	0.16	86.77	7.58	46.50
	DER	93.24	95.39	95.24	94.17	95.8	**52.17**	89.07	73.47
NAD	ACC	89.33	90.17	91.88	88.97	92.31	91.27	93.48	91.54
	ASR	2.08	10.68	9.98	18.43	98.8	88.06	70.96	99.97
	DER	95.11	93.16	94.39	90.79	50	52.38	61.52	**49.15**
I-BAU	ACC	88.13	90.31	86.52	86.33	89.67	89.27	91.01	91.14
	ASR	7.91	4.67	20.04	2.45	50.9	33.01	1.98	88.47
	DER	91.595	96.24	86.68	98.78	72.18	78.91	94.78	**54.70**
RNP	ACC	87.63	88.44	90.34	80.78	86.48	92.97	91.4	86.03
	ASR	3.76	3.78	0.17	99.93	0.73	91.52	24.23	96.51
	DER	93.42	95.74	98.52	48.33	95.67	51.5	83.85	**48.13**
Unit	ACC	84.66	86.91	88.05	81.36	80.05	87.79	79.7	88.15
	ASR	0.89	12.31	3.12	8.07	5.94	68.44	22.67	92.07
	DER	93.37	90.71	95.90	94.55	89.85	60.45	78.78	**51.41**
FST	ACC	87.06	88.96	92.40	88.89	92.67	92.18	91.25	89.68
	ASR	2.08	5.04	0.58	2.34	0	93.91	30.65	99.58
	DER	93.98	95.37	99.35	98.83	97.99	49.91	80.57	**48.42**
TSBD	ACC	90.13	92.24	92.48	89.06	93.18	92.85	92.91	93.11
	ASR	1.78	4.04	1.08	15.16	5.43	82.16	35.57	99.84
	DER	95.66	97.51	99.14	92.42	95.275	56.12	78.94	**50.00**

We further demonstrate the durability of our attack across fine-tuning epochs in Fig. 2, clearly distinguishing the effect of the gradient alignment process from that of traditional attacks.

Table 2 presents a comprehensive evaluation of existing backdoor defenses under our attack and baselines on Tiny-ImageNet. While most existing attacks are substantially suppressed by at least one strong defense (e.g., TSBD, RNP),



(a) CIFAR-10 **(b)** Tiny-ImageNet

Fig. 2: Effectiveness of our attack (red) compared with a standard attack (blue) across fine-tuning epochs with FT and TSBD defenses.

our attack consistently maintains high ASR across nearly all defenses, consistent with our observation in CIFAR-10. Strong regularization-based methods such as TSBD, which effectively suppress other attacks, still leave a ASR of 91.38% against ours. FST reduces ASR to 21.38%, but at the cost of ACC dropping to 32.74%, which aligns with our attack's design objective of embedding a deeply persistent backdoor. Notably, while ANP achieves partial suppression on CIFAR-10, it fails entirely on Tiny-ImageNet: the increased dataset complexity prevents it from identifying any valid pruning strategy that keeps ACC within the 10% drop threshold, leaving its performance identical to the pretrained model. Moreover, on Tiny-ImageNet, Narcissus achieves only 26.67% pretrained ASR, confirming that its clean-label advantage over our attack under ANP is dataset-specific rather than a general property.This highlights a fundamental scalability limitation of ANP and further validates the robustness of our attack on more complex datasets.

4.3 Ablation Studies

In this section, we analyze the robustness of our attack under a range of settings, including poisoning rate, fine-tuning budget, and model architecture, to demonstrate that our method remains effective across diverse practical conditions.

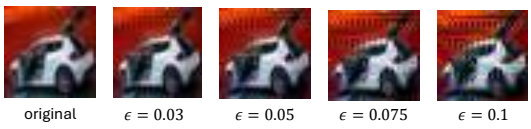


Fig. 3: Visualization of poisoned testing images with different noise rates

Effect of Noise Rate. Table 3 presents the impact of the perturbation scale ϵ. Several key observations emerge. First, clean accuracy remains consistently stable across all noise rates (within 1% variation), confirming that the trigger perturbation does not interfere with the model's benign functionality regardless

Table 2: Performance comparison of various backdoor defense methods against state-of-the-art backdoor attacks with Tiny-Imagenet. Results are reported across three metrics: Accuracy (ACC), Attack Success Rate (ASR), and Defense Efficiency Ratio (DER). "Pretrained" denotes the baseline performance of the backdoored models before applying any defense.

Methods	**Metrics**	**BadNet**	**Blended**	**WaNet**	**LC**	**Input-aware**	**Narcissus**	**COMBAT**	**Ours**
Pretrained	ACC	47.12	55.70	54.97	56.78	46.82	55.18	56.52	54.32
	ASR	94.16	99.54	99.70	67.70	95.24	26.67	86.43	99.90
FT	ACC	55.16	50.60	52.71	56.55	51.60	31.86	35.54	48.63
	ASR	90.39	75.57	48.00	67.33	98.81	4.99	55.72	99.26
	DER	55.91	59.44	74.72	50.07	52.39	49.18	54.87	**47.48**
ANP	ACC	47.12	54.00	54.22	54.68	42.21	55.17	52.86	54.32
	ASR	94.16	99.11	76.96	59.18	9.76	26.67	84.84	99.90
	DER	50.00	49.37	61.00	53.21	90.44	50.00	**48.97**	50.00
NAD	ACC	49.77	50.94	52.44	56.53	51.97	30.40	34.83	49.67
	ASR	29.53	69.88	51.59	70.11	99.36	2.95	45.87	98.28
	DER	83.64	62.45	72.79	49.88	52.58	49.47	59.44	**48.48**
I-BAU	ACC	46.04	45.33	45.39	44.48	42.28	48.80	40.24	47.19
	ASR	77.75	90.47	15.49	29.19	80.53	14.46	78.80	98.30
	DER	57.67	49.35	87.32	63.11	55.09	52.92	**45.68**	47.23
RNP	ACC	23.55	51.39	51.44	19.46	44.58	48.47	14.94	53.18
	ASR	67.46	0.00	0.02	4.71	0.02	18.05	39.50	99.90
	DER	51.57	97.62	98.08	62.84	96.49	50.96	52.68	**49.43**
Unit	ACC	1.17	2.61	7.39	4.80	1.87	12.97	13.58	3.09
	ASR	0.00	45.11	7.73	0.00	2.76	0.96	22.37	66.92
	DER	74.11	50.67	72.20	57.86	73.77	41.75	60.56	**40.87**
FST	ACC	26.96	29.06	28.75	28.44	24.25	33.63	33.02	32.74
	ASR	0.31	0.02	0.06	0.57	0.00	5.31	19.62	21.38
	DER	86.85	86.44	86.71	69.40	86.34	**49.91**	71.66	78.47
TSBD	ACC	52.64	54.76	52.73	54.97	54.65	50.93	53.24	53.90
	ASR	54.95	20.63	0.72	16.34	0.12	19.43	76.84	91.38
	DER	69.61	88.99	98.37	74.78	97.56	**51.50**	53.16	54.05

Table 3: Evaluation of attack stealth and efficacy across a range of noise rates. The table shows how varying the noise rate (ϵ) affects the trade-off between the backdoored model's classification accuracy and the success rate of the gradient alignment trigger against several defense mechanisms.

Methods	Pretrained		NAD		I-BAU		Unit		FST		TSBD	
Noise rate	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
ϵ=0.1	93.27	100.00	91.54	99.97	91.14	88.47	88.15	92.07	89.68	99.58	93.11	99.84
ϵ=0.075	92.82	100.00	90.21	98.88	88.62	94.97	87.61	96.62	89.67	99.12	91.44	97.44
ϵ=0.05	92.96	99.99	90.43	87.97	90.45	72.43	88.25	90.09	89.70	87.02	91.92	81.02
ϵ=0.03	92.91	99.92	89.98	57.06	89.20	51.84	87.77	69.06	89.19	71.99	91.92	47.76

of scale. Second, the overall trend of ASR decreasing with smaller ϵ is expected, as a smaller perturbation budget naturally constrains the trigger strength. However, even at $\epsilon = 0.03$ our attack still maintains a substantial ASR: Unit achieves

Table 4: Performance of our method vs. baseline attacks under different attack budgets. We report Clean Accuracy (C-Acc %) and Attack Success Rate (ASR %) for three poisoning ratios (1%, 5%, 10%). The proposed attack is highlighted in bold; lower ASR values indicate more effective defense performance against the corresponding attack.

Attack	Poisoning Ratio	Pretrained		FT		NAD		I-BAU		Unit		FST	
		C-Acc	ASR	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR
BadNet	0.1	91.44	94.41	91.01	66.94	89.62	2.66	85.62	7.91	84.66	0.89	91.48	3.13
	0.05	92.15	90.30	91.37	55.80	90.58	4.24	84.81	4.19	85.92	0.99	91.37	1.3
	0.01	93.34	71.21	92.39	53.53	91.12	9.63	89.21	2.82	87.56	1.18	92.5	0.94
WaNet	0.1	93.40	99.97	93.75	76.73	93.72	78.07	86.52	20.04	88.05	3.12	93.04	0.3
	0.05	93.37	99.87	93.89	98.14	93.88	98.73	90.05	2.3	88.91	2.3	93.32	0.27
	0.01	93.81	98.48	93.80	74.12	93.83	87.26	88.29	7.31	88.12	1.38	93.18	1.13
LC	0.1	84.19	100.0	92.80	100.0	91.39	75.88	86.33	2.46	81.36	8.07	91.08	0
	0.05	93.32	100.0	92.26	100.0	90.82	100.0	88.25	4.25	88.14	1.49	92.22	86.91
	0.01	93.54	99.97	92.36	100.0	91.51	99.01	88.52	3.85	88.5	1.33	92.59	99.52
Narcissus	0.1	93.09	94.64	92.35	89.81	91.27	88.07	89.27	33.01	87.79	68.44	92.18	93.91
	0.05	93.77	79.64	92.87	72.99	92.69	70.58	15.21	0.0	83.80	83.76	92.76	85.79
	0.01	93.43	38.26	92.32	36.29	92.13	33.56	12.58	25.67	86.68	46.8	92.33	43.27
COMBAT	0.1	85.05	99.23	91.90	83.47	91.80	52.28	90.01	1.98	79.70	22.67	92.27	58.40
	0.05	93.94	94.47	93.46	72.83	98.41	76.56	91.32	71.76	86.34	44.20	91.25	30.65
	0.01	94.14	83.67	93.49	73.76	93.40	78.12	91.76	78.09	83.85	24.16	93.60	78.07
Ours	0.1	93.27	100.0	92.20	99.99	91.54	99.97	91.14	88.47	88.15	92.07	89.68	99.58
	0.05	93.27	100.0	90.83	99.96	89.92	100.0	89.21	97.74	88.61	98.80	89.84	99.96
	0.01	93.47	99.99	90.63	99.93	90.34	99.93	87.82	99.96	88.26	99.29	89.47	99.77

69.06% and FST achieves 71.99%, demonstrating that the gradient alignment mechanism remains effective even under a tight perturbation budget. At $\epsilon \geq 0.075$, the attack achieves strong and consistent ASR across all defenses, with FST and TSBD remaining particularly vulnerable (>97% ASR). Based on these results, we adopt $\epsilon = 0.1$ as our default setting. Visualizations of different noise rates can be seen in Fig. 3

Effect of Poisoning Rate. To evaluate the sensitivity of our method to poisoning density, we vary the poisoning rate across $[0.01, 0.05, 0.1]$; results are summarized in Table 4.

A primary observation is that traditional attacks such as BadNet and WaNet are effectively neutralized by existing defenses, with ASR frequently collapsing to below 5% under at least one, if not all, of the evaluated defenses (cf. Fig. 4). Similarly, more sophisticated attacks such as Narcissus, while resilient under some configurations, remain susceptible to certain defenses. In contrast, our method consistently evades all evaluated defenses, maintaining a near-saturated ASR exceeding 97% across all defensive configurations, even at low poisoning rates. This robustness further extends to varying poisoning densities: while attacks such as Narcissus exhibit sharp performance degradation under low poisoning rates, with ASR dropping from ∼94% to ∼38% as the poisoning rate decreases from 0.1 to 0.01, our method sustains an ASR above 99% even at a minimal poisoning ratio of 0.01. *To this end, these results demonstrate that our attack*

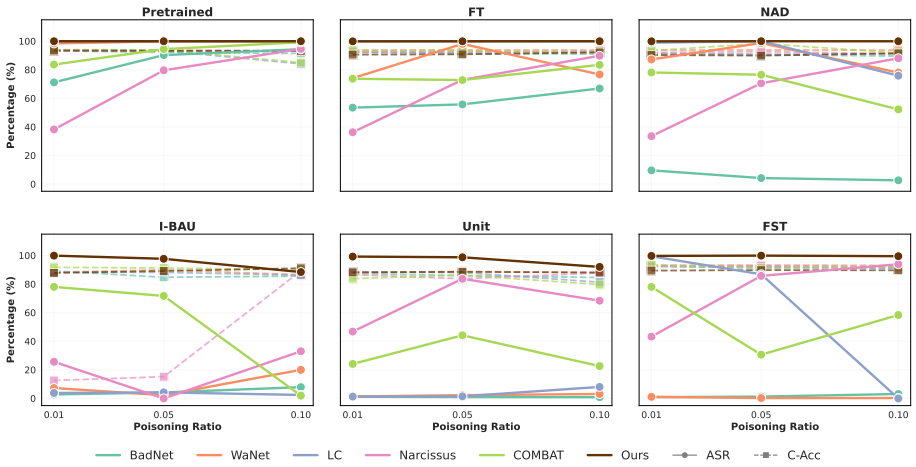


Fig. 4: Visualization of the effectiveness of different defenses against attacks with different poisoning rates.

Table 5: Comparison of methods across ACC, ASR, and DER metrics (four runs each).

	FT = 0.1			**FT = 0.05**			**FT = 0.02**			**FT = 0.01**		
Method	ACC	ASR	DER	ACC	ASR	DER	ACC	ASR	DER	ACC	ASR	DER
Pretrained	92.74	100.00	0.00	92.74	100.00	0.00	92.74	100.00	0.00	92.74	100.00	0.00
FT	83.02	90.67	57.46	77.65	88.39	47.83	80.56	85.14	46.20	75.89	94.11	50.00
NAD	83.22	96.72	46.88	86.79	99.63	47.21	82.90	79.10	55.53	82.29	96.76	46.40
I-BAU	88.75	99.63	48.19	85.83	79.21	56.94	80.41	45.53	71.07	84.36	31.72	79.95
Unit	86.34	97.40	48.10	86.16	98.99	47.26	86.47	98.97	47.38	85.85	98.98	47.07
FST	92.01	99.46	49.91	92.01	100.00	49.64	92.08	100.00	49.67	91.84	100.00	49.55

effectively entangles backdoor and clean optimization signals, ensuring persistent attack efficacy across varying poisoning rates and defensive settings.

Effect of Fine-tuning Ratios We study the stealthiness of our attack against various defense mechanisms under increasingly constrained fine-tuning budgets, specifically ratios of $\{0.1, 0.05, 0.02, 0.01\}$. The quantitative results are reported in Table 5 and visualized in Fig. 5.

As shown, our attack remains persistent across defenses as the fine-tuning budget decreases. Unit and FST yield ASR similar to the no-defense baseline (i.e., "Pretrained"), demonstrating their inability to remove the backdoor regardless of data budget. Other defenses, such as NAD, fail to suppress the attack, with ASR remaining above 80% in most configurations.

We also observe the performance instability in some defenses under different fine-tuning budgets. For instance, I-BAU achieves its lowest ASR at a ratio of 0.01, but simultaneously indicates a significant degradation in model clean-accuracy

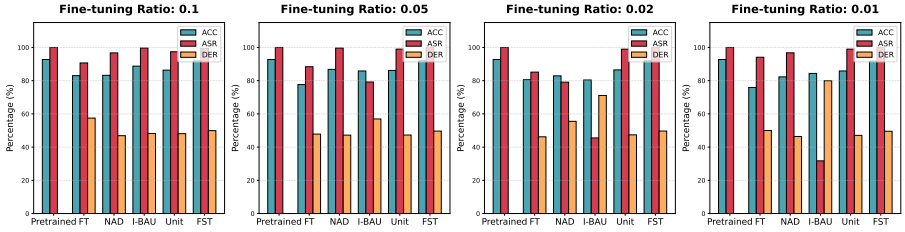


Fig. 5: Visualization of the stealthiness of our attack under different defenses' fine-tuning ratio.

Table 6: Clean accuracy (C-ACC, %) and attack success rate (ASR, %) of different defenses across model architectures. Higher C-ACC and lower ASR indicate better robustness. An effective defense is defined as achieving an accuracy drop no greater than 5% from the pretrained model while maintaining ASR $\leq$ 20%.

Method	PreAct-ResNet18		VGG19 BN		MobileNetV3-Large		ViT-B_16		EfficientNet-B3	
	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR	C-ACC	ASR
Pretrained	93.27	100.00	91.94	100.00	83.74	99.99	96.58	96.58	53.47	98.43
FT	92.20	99.99	89.14	83.41	77.76	98.01	24.08	10.28	59.61	79.19
NAD	91.54	99.97	87.93	85.38	77.84	99.37	30.67	15.18	62.80	99.34
I-BAU	91.14	88.47	88.35	72.68	78.03	94.51	40.18	18.16	60.06	73.73
Unit	88.15	92.07	10.00	0.00	10.26	74.30	52.11	100.00	10.00	100.00
FST	89.68	99.58	90.20	98.30	82.91	99.84	95.68	100.00	60.49	91.03
TSBD	93.11	99.84	91.35	88.92	49.63	19.84	23.49	14.04	27.62	5.84

rather than a genuine backdoor removal. *In conclusion, varying fine-tuning budgets does not necessarily improve the performance of defenses against our attacks.*

Model Architectures. We evaluate the proposed method through a comprehensive analysis against diverse defense strategies across multiple model architectures, including PreAct-ResNet18, VGG19_BN, MobileNetV3-Large, ViT-B-16, and EfficientNet-B3; results are presented in Table 6 and visualized in Fig. 6. Our method maintains consistently high ASR across all architectural configurations, exceeding 97% in almost all settings. Although Unit and FST prove effective against traditional attacks such as BadNet, WaNet, and Narcissus (see more result in our Appendix), they fail to mitigate our attack, which sustains an ASR above 99% even at a poisoning ratio.

To better quantify defense effectiveness, we define a *Sweet Spot* as achieving an accuracy drop no greater than 5% from the pretrained model while reducing ASR to at most 20% - thresholds reflecting practical deployment requirements where utility and security must jointly hold. As shown, most defenses fail to reach this region. TSBD, for instance, achieves a low ASR of 5.84% but at the cost of severe utility collapse, with C-Acc dropping to 23.49% on ViT-B/16. Our attack maintains high ASR across both convolutional and transformer-based ar-

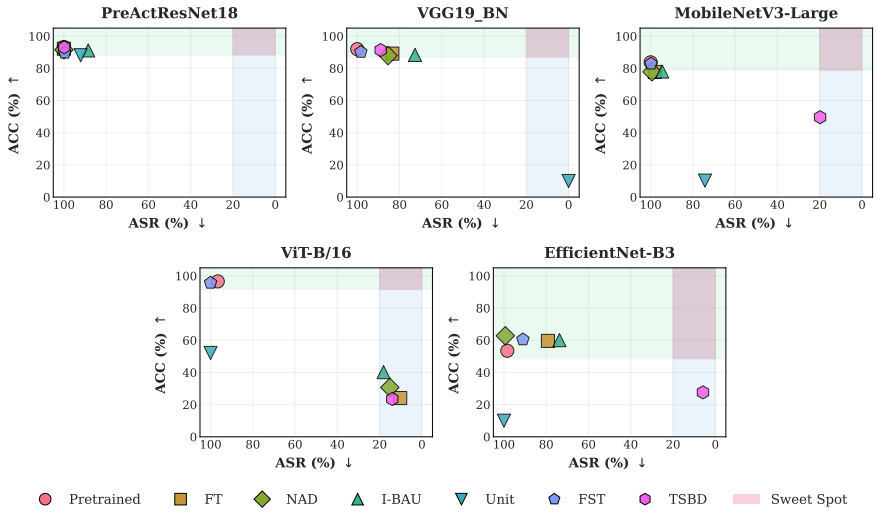


Fig. 6: Visualization of C-ACC and ASR of different defense methods with different model architectures. Higher C-ACC and lower ASR indicate better robustness. An effective defense is defined as achieving an accuracy drop of no more than 5% relative to the pretrained model while maintaining ASR $\leq 20\%$ (i.e., resulting in a Sweet Spot for each setting).

chitectures, bypassing defenses that otherwise cause C-Acc to collapse.
Taken together, our method yields consistent attack behavior across architectures ranging from traditional models such as PreAct-ResNet18 to transformer-based models such as ViT-B/16. *This architectural consistency suggests that our method exploits vulnerabilities fundamental to post-training defenses rather than being specific to any particular model design.*

5 Conclusion

In this paper, we investigated an overlooked threat model in which an adversary deliberately entangles backdoor and clean optimization signals to resist fine-tuning-based defenses. By aligning backdoor gradients with clean gradients, our method shows that any attempt to unlearn the backdoor inevitably degrades clean performance, rendering existing defenses ineffective regardless of budget or setting. Comprehensive evaluations across diverse model architectures and defense configurations confirm that our attack consistently maintains a near-perfect $\sim$99% ASR, highlighting the need to rethink the design assumptions underlying current post-training backdoor defenses.

References

1. Chen, W., Wu, B., Wang, H.: Effective backdoor defense by exploiting sensitivity of poisoned samples. Advances in Neural Information Processing Systems **35**, 9727–9737 (2022)
2. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
3. Cheng, P., Wu, Z., Du, W., Zhao, H., Lu, W., Liu, G.: Backdoor attacks and countermeasures in natural language processing models: A comprehensive security review. IEEE Transactions on Neural Networks and Learning Systems (2025)
4. Cheng, S., Shen, G., Zhang, K., Tao, G., An, S., Guo, H., Ma, S., Zhang, X.: Unit: Backdoor mitigation via automated neural distribution tightening. In: European Conference on Computer Vision. pp. 262–281. Springer (2025)
5. Costa, J.C., Roxo, T., Proença, H., Inacio, P.R.M.: How deep learning sees the world: A survey on adversarial attacks & defenses. IEEE Access **12**, 61113–61136 (2024)
6. Dao, T., Doan, K.D., Wong, K.S.: Clean-label physical backdoor attacks with data distillation. arXiv preprint arXiv:2407.19203 (2024)
7. Dibbo, S.V.: Sok: Model inversion attack landscape: Taxonomy, challenges, and future roadmap. In: 2023 IEEE 36th Computer Security Foundations Symposium (CSF). pp. 439–456. IEEE (2023)
8. Doan, K.D., Lao, Y., Zhao, W., Li, P.: Lira: Learnable, imperceptible and robust backdoor attacks. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 11946–11956 (2021), `https://api.semanticscholar.org/CorpusID:244397177`
9. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: Strip: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th annual computer security applications conference. pp. 113–125 (2019)
10. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)
11. Guo, Z., Kumar, A., Tourani, R.: Persistent backdoor attacks in continual learning. USENIX Association, USA (2025)
12. Hanif, M., Chattopadhyay, N., Ouni, B., Shafique, M.: Survey on backdoor attacks on deep learning: Current trends, categorization, applications, research challenges, and future prospects. IEEE Access **PP**, 1–1 (01 2025). `https://doi.org/10.1109/ACCESS.2025.3571995`
13. Hu, Y., Kuang, W., Qin, Z., Li, K., Zhang, J., Gao, Y., Li, W., Li, K.: Artificial intelligence security: Threats and countermeasures. ACM Computing Surveys (CSUR) **55**(1), 1–36 (2021)
14. Huang, W.R., Geiping, J., Fowl, L., Taylor, G., Goldstein, T.: Metapoison: Practical general-purpose clean-label data poisoning. Advances in Neural Information Processing Systems **33**, 12080–12091 (2020)
15. Huynh, T., Nguyen, D., Pham, T., Tran, A.: Combat: Alternated training for effective clean-label backdoor attacks. In: AAAI Conference on Artificial Intelligence (2024), `https://api.semanticscholar.org/CorpusID:268678332`
16. Jin, L.X., Jiang, W., Wen, X.Y., Lin, M.Y., Zhan, J.Y., Zhou, X.Z., Habtie, M.A., Werghi, N.: A survey of backdoor attacks and defences: From deep neural networks to large language models. Journal of Electronic Science and Technology p. 100326 (2025)

17. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
18. Kurita, K., Michel, P., Neubig, G.: Weight poisoning attacks on pretrained models. In: Proceedings of the 58th annual meeting of the association for computational linguistics. pp. 2793–2806 (2020)
19. Li, S., Wen, Y., Wang, H., Cheng, X.: Badlidet: A simple backdoor attack against lidar object detection in autonomous driving. In: 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). pp. 99–108. IEEE (2023)
20. Li, Y., Huang, H., Zhao, Y., Ma, X., Sun, J.: Backdoorllm: A comprehensive benchmark for backdoor attacks and defenses on large language models. arXiv preprint arXiv:2408.12798 (2024)
21. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Anti-backdoor learning: Training clean models on poisoned data. Advances in Neural Information Processing Systems **34**, 14900–14912 (2021)
22. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Neural attention distillation: Erasing backdoor triggers from deep neural networks. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=9l0K4OM-oXE
23. Li, Y., Lyu, X., Ma, X., Koren, N., Lyu, L., Li, B., Jiang, Y.G.: Reconstructive neuron pruning for backdoor defense. In: ICML (2023)
24. Li, Y., Wu, B., Jiang, Y., Li, Z., Xia, S.: Backdoor learning: A survey. IEEE Transactions on Neural Networks and Learning Systems **35**, 5–22 (2020), https://api.semanticscholar.org/CorpusID:220633116
25. Li, Y., Zhang, S., Wang, W., Song, H.: Backdoor attacks to deep learning models and countermeasures: A survey. IEEE Open Journal of the Computer Society **4**, 134–146 (2023). https://doi.org/10.1109/OJCS.2023.3267221
26. Lin, W., Liu, L., Wei, S., Li, J., Xiong, H.: Unveiling and mitigating backdoor vulnerabilities based on unlearning weight changes and backdoor activeness. Advances in Neural Information Processing Systems **37**, 42097–42122 (2024)
27. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: ECCV (2020)
28. Min, R., Qin, Z., Shen, L., Cheng, M.: Towards stable backdoor purification through feature shift tuning. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
29. Nguyen, D.T., Tran, N.N., Johnson, T.T., Leach, K.: Pbp: Post-training backdoor purification for malware classifiers. ArXiv **abs/2412.03441** (2024), https://api.semanticscholar.org/CorpusID:274465177
30. Nguyen, T.A., Tran, A.: Input-aware dynamic backdoor attack. Advances in Neural Information Processing Systems **33**, 3454–3464 (2020)
31. Nguyen, T.A., Tran, A.T.: Wanet - imperceptible warping-based backdoor attack. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=eEn8KTtJOx
32. Ning, R., Li, J., Xin, C., Wu, H.: Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications. pp. 1–10. IEEE (2021)
33. Perez-Cerrolaza, J., Abella, J., Borg, M., Donzella, C., Cerquides, J., Cazorla, F.J., Englund, C., Tauber, M., Nikolakopoulos, G., Flores, J.L.: Artificial intelligence for

safety-critical systems in industrial and transportation domains: A survey. ACM Computing Surveys **56**(7), 1–40 (2024)
34. Pham, H., Ta, T.A., Tran, A., Doan, K.D.: Flatness-aware sequential learning generates resilient backdoors. In: European Conference on Computer Vision. pp. 89–107. Springer (2024)
35. Qi, X., Xie, T., Li, Y., Mahloujifar, S., Mittal, P.: Revisiting the assumption of latent separability for backdoor defenses. In: The eleventh international conference on learning representations (2023)
36. Qi, Y., et al.: Revisiting the security of deep neural networks: Persistent backdoors. In: USENIX Security (2022)
37. Souri, H., Fowl, L., Chellappa, R., Goldblum, M., Goldstein, T.: Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. Advances in Neural Information Processing Systems **35**, 19165–19178 (2022)
38. Turner, A., Tsipras, D., Madry, A.: Label-consistent backdoor attacks. arXiv preprint arXiv:1912.02771 (2019)
39. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE symposium on security and privacy (SP). pp. 707–723. IEEE (2019)
40. Wang, H., Xiang, Z., Miller, D.J., Kesidis, G.: Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic (2023), https://arxiv.org/abs/2205.06900
41. Wang, T., Yao, Y., Xu, F., An, S., Tong, H., Wang, T.: An invisible black-box backdoor attack through frequency domain. In: European Conference on Computer Vision. pp. 396–413. Springer (2022)
42. Wu, B., Chen, H., Zhang, M., Zhu, Z., Wei, S., Yuan, D., Shen, C.: Backdoorbench: A comprehensive benchmark of backdoor learning. Advances in Neural Information Processing Systems **35**, 10546–10559 (2022)
43. Wu, D., Wang, Y.: Adversarial neuron pruning purifies backdoored deep models. Advances in Neural Information Processing Systems **34**, 16913–16925 (2021)
44. Yazmyradov, S., Lee, H.J.: A comprehensive review of ai security: Threats, challenges, and mitigation strategies. International Journal of Internet, Broadcasting and Communication **16**(4), 375–384 (2024)
45. Yin, W., Lou, J., Zhou, P., Xie, Y., Feng, D., Sun, Y., Zhang, T., Sun, L.: Physical backdoor: Towards temperature-based backdoor attacks in the physical world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12733–12743 (2024)
46. Zeng, Y., Chen, S., Park, W., Mao, Z., Jin, M., Jia, R.: Adversarial unlearning of backdoors via implicit hypergradient. In: International Conference on Learning Representations (2021)
47. Zeng, Y., Pan, M., Just, H.A., Lyu, L., Qiu, M., Jia, R.: Narcissus: A practical clean-label backdoor attack with limited information. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 771–785 (2023)
48. Zeng, Y., Park, W., Mao, Z.M., Jia, R.: Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective . In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16453–16461. IEEE Computer Society, Los Alamitos, CA, USA (Oct 2021). https://doi.org/10.1109/ICCV48922.2021.01616, https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.01616