Backdoor Mitigation by Distance-Driven Detoxification

Shaokui Wei[1]   Jiayin Liu[1]   Hongyuan Zha[1,2]

[1]The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, P.R. China
[2]Shenzhen Key Laboratory of Crowd Intelligence Empowered Low-Carbon Energy Network

Abstract

Backdoor attacks undermine the integrity of machine learning models by allowing attackers to manipulate predictions using poisoned training data. Such attacks lead to targeted misclassification when specific triggers are present, while the model behaves normally under other conditions. This paper considers a post-training backdoor defense task, aiming to detoxify the backdoors in pre-trained models. We begin by analyzing the underlying issues of vanilla fine-tuning and observe that it is often trapped in regions with low loss for both clean and poisoned samples. Motivated by such observations, we propose Distance-Driven Detoxification (D3), an innovative approach that reformulates backdoor defense as a constrained optimization problem. Specifically, D3 promotes the model's departure from the vicinity of its initial weights, effectively reducing the influence of backdoors. Extensive experiments on state-of-the-art (SOTA) backdoor attacks across various model architectures and datasets demonstrate that D3 not only matches but often surpasses the performance of existing SOTA post-training defense techniques. Code is available at https://github.com/shawkui/Distance_Driven_Detoxification.

1. Introduction

Over the past decades, Deep Neural Networks (DNNs) have achieved conspicuous progress in various domains and applications, such as face recognition, autonomous driving, and healthcare [1, 9, 25, 37]. Despite these advancements, DNNs face significant challenges due to their susceptibility to malicious attacks. A notable threat is the rise of backdoor attacks, where adversaries secretly insert backdoors into DNN models during training by subtly altering a subset of the training data. These alterations ensure that the model performs normally on benign inputs while consistently misclassifying inputs with a specific trigger. To protect machine learning systems, especially in high-stakes applications, it is imperative to develop robust defenses against such threats.

To tackle the threats posed by backdoor attacks, re-searchers have explored a wide range of defense strategies throughout the lifecycle of machine learning systems [43]. This paper specifically focuses on the post-training backdoor defense task, which aims to eliminate backdoors from a given pre-trained models [42–44]. One prominent strategy in this area is fine-tuning the models, *i.e.*, adjusting model weights on an additional dataset, to reduce the impact of backdoor attacks. However, vanilla fine-tuning, which solely employs classification loss on clean samples, has proven insufficient against sophisticated backdoor attacks [32, 42, 43, 51].

One of the primary reasons for the ineffectiveness of vanilla fine-tuning stems from the misalignment between the ideal defense goals and the actual objectives pursued by vanilla fine-tuning. To mitigate backdoor, an ideal objective would aim to minimize the loss of clean data (**clean loss**) while concurrently increasing the loss of poisoned data (**backdoor loss**). Due to the inaccessibility of poisoned data to defenders, vanilla fine-tuning focuses solely on minimizing the clean loss, while neglecting the backdoor loss. This oversight renders it ineffective against sophisticated backdoor attacks. To address such misalignment, a promising research direction is to reconstruct the poisoned samples and fine-tune the model to resist them, thereby explicitly reducing backdoor loss [38, 39, 48, 52]. While these methods achieve significant performance improvements, they often come with increased computational complexity. Recently, there has been a shift in focus towards refining the fine-tuning process itself. Techniques like sharpness-aware minimization [51] have been integrated into the fine-tuning procedure to implicitly reduce backdoor loss, thereby improving its effectiveness for backdoor mitigation. However, more recent findings suggest that even these advanced methods can still be vulnerable to attacks when the optimization process is carefully designed [13].

In this paper, we delve into the inherent issues of vanilla fine-tuning by investigating the trajectory between the backdoored weights and the fine-tuned weights, revealing that *vanilla fine-tuning often gets trapped in regions with both low clean loss and low backdoor loss*. This phenomenon causes vanilla fine-tuning to converge to bad local solutions, which undermines its defensive efficacy (as illustrated in
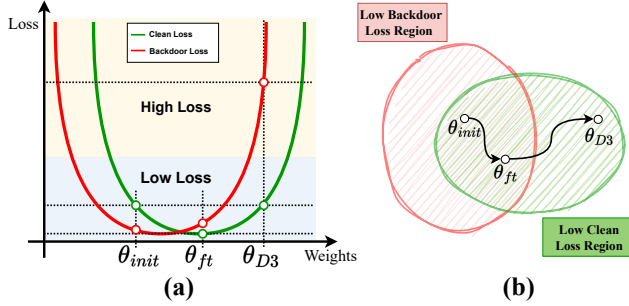
Figure 1. **(a)**: An illustrative example for curves of clean loss and backdoor loss, highlighting three key points, $i.e.$, the initial weights θ_{init}, the weights after vanilla fine-tune θ_{ft}, and the weights after applying our method θ_{D3}. **(b)**: A demonstration of loss regions. Vanilla fine-tuning is often trapped in regions where both types of loss are low, thus failing to eliminate backdoors. In contrast, our method finds a more distant solution, thereby escaping this trap and enhancing backdoor defense.

Figure 1). To address this challenge, we propose Distance-Driven Detoxification (D3), an innovative strategy for mitigating backdoor attacks. Specifically, we begin by formulating the backdoor defense task as a constrained optimization problem, with the goal of identifying weights that are maximally distant from the initial backdoored model weights while ensuring that the loss for clean data remains within acceptable bounds. Such goal facilitates the model to escape the region of low backdoor loss, and thereby, effectively mitigating the backdoor effect. Considering the practical challenges, this formulation is subsequently converted into a regularized optimization problem, enabling efficient computation with minimal additional overhead compared to vanilla fine-tuning. Furthermore, we benchmark D3 against eight state-of-the-art (SOTA) post-training defense techniques across seven SOTA backdoor attacks, encompassing various model architectures and datasets. Our experimental results demonstrate that D3 not only achieves performance on par with existing baseline approaches but frequently outperforms them.

Our main contributions are threefold: **1) Insight into the failure of vanilla fine-tuning:** We conduct a deeper analysis of the reasons behind the failure of vanilla fine-tuning, revealing that it often becomes trapped in regions with low backdoor loss, which significantly impedes the mitigation of backdoor effects. **2) Novel optimization framework:** We introduce a new optimization framework designed to address the limitations of vanilla fine-tuning by maximizing the distance from the backdoored model while maintaining low loss on clean data. This approach provides a straightforward yet effective method to mitigate backdoor attacks. **3) Comprehensive evaluation:** We perform extensive experiments to rigorously evaluate the effectiveness of our proposed method, comparing it against eight SOTA defense techniques across

seven challenging backdoor attacks, across a diverse set of model architectures and datasets.

2. Related work

Backdoor attacks. Deep neural networks are vulnerable to backdoor attacks, which pose a significant security threat. These attacks are crafted to ensure that the network performs normally on standard inputs but outputs a pre-defined target when a specific trigger is present. Backdoor attacks can be broadly classified into two categories based on the nature of the trigger: static-pattern backdoor attacks and dynamic-pattern backdoor attacks. The pioneering work in static-pattern backdoors, known as BadNets [8], utilized fixed triggers such as white squares. To improve the stealthiness of these triggers, the Blended approach [4] was developed, which integrates the trigger seamlessly into the host image. However, the fixed nature of these triggers made them susceptible to detection, leading researchers to shift their focus towards dynamic-pattern backdoor attacks. Recent advancements in dynamic-pattern backdoor attacks include methods like WaNet [30], LF [47] and SSBA [19]. These techniques generate sample-specific triggers that are more difficult to detect. Additionally, 'clean label' attacks such as LC [33] and SIG [2] have been introduced to carry out attacks without disrupting the consistency between the image and its corresponding label.

Backdoor defenses. The primary objective of backdoor defense is to mitigate the susceptibility of DNNs to backdoor attacks. These defenses are generally classified into three categories: pre-training, in-training, and post-training. Pre-training defenses focus on identifying and eliminating poisoned samples before the training process begins. For instance, methods like AC [3], Confusion Training [31], AGPD [46], VDC [53] leverages various techniques for this purpose. In-training backdoor defenses aim to reduce the impact of backdoors during the training phase. Techniques such as ABL [20] exploit the faster learning rate of backdoor samples compared to clean samples, and use this to selectively forget poisoned data. DBD [12] segments the backdoor training process to inhibit the backdoor learning. PDB [40] and NAB [26] eliminate the backdoor effect by injecting another defensive backdoor.

This paper primarily focuses on post-training defenses designed to neutralize backdoors in pre-trained models. One line of research involves identifying and pruning neurons associated with backdoors, including methods like FP [24], ANP [45], EP [50], and CLP [49]. Another significant approach is to fine-tune the compromised model to mitigate backdoor effects. Notable examples include NC [38], i-BAU [48], NPD [52], TSBD [23] and SAU [39], which use adversarial techniques to reconstruct potential backdoor triggers and then fine-tune the model to resist these reconstructed

poisoned samples, effectively cleansing the model. Additionally, NAD [21] employs a teacher network to guide the fine-tuning process of a backdoored student network, aiding in backdoor mitigation. The work most closely related to our study is FT-SAM [51], which improves the efficacy of fine-tuning for backdoor mitigation through the integration of sharpness-aware minimization. For more detailed information on defenses in adversarial machine learning, readers are directed to the comprehensive survey by Wu et al. [43].

3. Method

In Section 3.1, we establish the notations and introduce the threat model. In Section 3.2, we revisit vanilla fine-tuning for backdoor defense to identify the underlying issues. Building on these insights, we present a novel optimization formulation for backdoor mitigation in Section 3.3. This formulation can be converted into a regularized optimization problem, facilitating practical and efficient solutions.

3.1. Problem setting

Notations. In this work, we focus on a classification task aimed at assigning a label $y \in \mathcal{Y}$ to a sample $\boldsymbol{x} \in \mathcal{X}$. Here, $\mathcal{Y} = [1, \ldots, K]$ (with $K \geq 2$) represents the set of possible labels, and $\mathcal{X}$ denotes the sample space. To accomplish this, a model $f_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta}$ is trained by solving the following problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y) \in \mathcal{D}_{tr}} \left[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) \right], \quad (1)$$

where $\mathcal{D}_{tr}$ is the training dataset and ℓ is the loss function.

In the scenario of backdoor attacks, the trigger is denoted by Δ, and a poisoned sample can be created by planting Δ to a clean sample $\boldsymbol{x}$, resulting in $\boldsymbol{x} + \Delta$.

Threat model. We consider a general backdoor attack scenario where an adversary has the capability to alter a fraction of the training dataset and/or influence the training process to embed a backdoor. The backdoored model behaves normally for benign inputs but misclassifies inputs with the malicious trigger Δ to a specific target label $\hat{y}$. The proportion of altered samples is referred to as the **poisoning ratio** of the backdoor attack.

Defender's goal. We consider a post-training backdoor defense scenario. The defender is provided with a model that may have been compromised by a backdoor attack. The primary objective is to mitigate the backdoor so that the model no longer responds to the malicious trigger Δ while preserving its performance on the original classification task. We assume the defender has access to a small, clean dataset $\mathcal{D}_{cl}$, which is a common assumption in most post-training defense strategies [21, 24, 38, 39, 48, 51, 52]. This dataset

can be obtained through various means, such as purchasing from trusted data providers, generating using advanced generative models [5, 7, 15], collecting from online resources, or employing data cleaning techniques [3, 31, 43, 53]. Additionally, the defender does not have prior knowledge of the specific malicious trigger Δ or the target label $\hat{y}$.

3.2. Revisiting fine-tuning for backdoor mitigation

Given a backdoored model, fine-tuning, which involves adjusting the model weights using an additional dataset, is a promising method for mitigating backdoor effects. Due to its simplicity, vanilla fine-tuning emerges as a natural approach to detoxifying backdoors hidden within the model. However, previous studies [27, 42, 44, 51] have observed that vanilla fine-tuning often falls short in effectively removing backdoors. To explore the reason, we first revisit the ideal objective for defending against backdoor attacks. Given a clean dataset $\mathcal{D}_{cl}$, trigger Δ, target label $\hat{y}$, and a loss function ℓ, the ideal fine-tuning objective to purify model $f_{\boldsymbol{\theta}}$ is:

$$\min_{\boldsymbol{\theta}} \overbrace{\underbrace{\mathbb{E}_{(\boldsymbol{x},y) \in \mathcal{D}_{cl}} \left[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) \right]}_{\textbf{Clean loss}} - \underbrace{\mathbb{E}_{(\boldsymbol{x},y) \in \mathcal{D}_{cl}} \left[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \Delta), \hat{y}) \right]}_{\textbf{Backdoor loss}}}^{\textbf{Ideal loss}},$$

$$(2)$$

which seeks to preserve the model's performance on clean data while simultaneously unlearning the backdoor. Therefore, an ideally purified model would exhibit both a low clean loss and a high backdoor loss. In contrast, vanilla fine-tuning aims to minimize only the clean loss, which diverges from the ideal objective for backdoor defense. This discrepancy limits the efficacy of vanilla fine-tuning in addressing backdoor threats.

To further understand such discrepancy, we investigate how the backdoor loss evolves along a trajectory connecting the initial model to the fine-tuned model. Starting from a backdoored model with initial weights $\boldsymbol{\theta}_{init}$, we apply vanilla fine-tuning to obtain fine-tuned weights $\boldsymbol{\theta}_{ft}$. Then, a trajectory is constructed along the direction $\boldsymbol{\theta}_{ft} - \boldsymbol{\theta}_{init}$, where each point on this path is given by $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{init} + t \cdot (\boldsymbol{\theta}_{ft} - \boldsymbol{\theta}_{init})$. We conduct experiments using four backdoor attacks: BadNets[8], WaNet [30], Input-aware [29] and SSBA [19], employing the PreAct-ResNet18 architecture. Our primary focus is on vanilla fine-tuning itself; therefore, we fine-tune the backdoored models using all accessible clean samples to eliminate the impact of data insufficiency. We then visualize the changes in loss along the trajectory, with t varying from 0 to 2.

The results are shown in Figure 2, and our key observations are as follows:

- *The initial backdoored model are often trapped in a "backdoor region" where vanilla fine-tuning struggles to escape.* We define the backdoor region as a subset of parameter
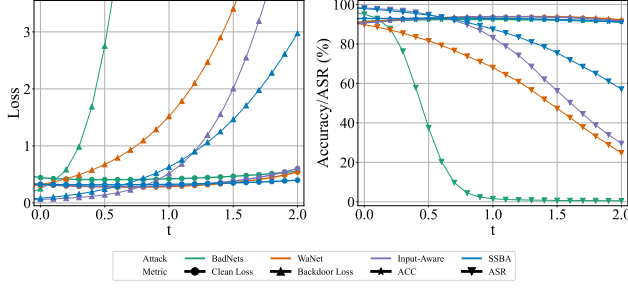
Figure 2. **Left:** The visualization of curves for clean loss and backdoor loss, along the trajectory for fine-tuning four attacks. **Right:** The visualization of curves for Accuracy and Attack Success Rate, along the trajectory for fine-tuning four attacks.

space around the initial weight where the model continues to respond strongly to the backdoor trigger. Formally, consider a neighborhood around the initial weights: $\mathcal{B}_\eta(\boldsymbol{\theta}_{\text{init}})$, a ball of radius η centered at $\boldsymbol{\theta}_{\text{init}}$. The backdoor region is then:

$$\mathcal{G}_{bd} = \{\boldsymbol{\theta} \in \mathcal{B}_\eta(\boldsymbol{\theta}_{\text{init}}) | L_{\text{bd}}(f_{\boldsymbol{\theta}}) \leq \alpha\},$$

where L_{bd} measures backdoor loss (e.g., the model's tendency to predict the target label $\hat{y}$ when the trigger is applied), and α is a threshold indicating successful attack behavior. In this region, the model's clean loss may improve during fine-tuning, but the backdoor loss remains low, meaning the attack remains effective. Indeed, across all tested attacks (except BadNets), we observe that clean loss decreases steadily from $t = 0$ to $t = 1$, while backdoor loss stays flat, indicating the model remains stuck within this harmful region. Vanilla fine-tuning, which optimizes only for clean accuracy, lacks the directional pressure to escape it.

- *Shifting the weights away from the initial weights can help mitigate backdoors while preserving clean performance.* By extending the trajectory of the weights slightly beyond the fine-tuned point ($1 < t < 2$), there is a substantial increase in backdoor loss across various attacks, along with a significant decrease in the Attack Success Rate. Importantly, this adjustment has little negative impact on the clean loss. These findings suggest that increasing the separation between the initial backdoored model and the fine-tuned model is a promising strategy to lower ASR without sacrificing performance on clean data.

Conceptual understanding via Local Taylor Expansion: To provide an understanding of the observed backdoor loss dynamics, we apply a second-order Taylor expansion as a local approximation. While neural networks are inherently high-dimensional and non-linear, this approximation serves as a local probe to characterize the loss landscape.

As the initial model is optimized to minimize loss over a poisoned dataset, we assume that $\boldsymbol{\theta}_{init}$

is a local minimum of the backdoor loss $\mathcal{L}_{bd}$, *i.e.*, $\nabla \mathcal{L}_{bd}(\boldsymbol{\theta}_{init}) = 0$. Then, we expand $\mathcal{L}_{bd}(\boldsymbol{\theta}_t)$ around $\boldsymbol{\theta}_{init}$:

$$\mathcal{L}_{bd}(\boldsymbol{\theta}_t) - \mathcal{L}_{bd}(\boldsymbol{\theta}_{init}) \approx \frac{1}{2}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{init})^T H(\boldsymbol{\theta}_{init})(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{init})$$

$$= \frac{t^2}{2}(\boldsymbol{\theta}_{ft} - \boldsymbol{\theta}_{init})^T H(\boldsymbol{\theta}_{init})(\boldsymbol{\theta}_{ft} - \boldsymbol{\theta}_{init}),$$

where $H(\boldsymbol{\theta}_{init}) = \nabla^2 \mathcal{L}_{bd}(\boldsymbol{\theta}_{init})$ is the Hessian matrix of $\mathcal{L}_{bd}$ evaluated at $\boldsymbol{\theta}_{init}$.

The second-order optimality condition for a local minimum ensures the Hessian matrix is positive semi-definite [22], guaranteeing the quadratic term ≥ 0. Thus, as t grows, the backdoor loss $\mathcal{L}_{bd}(\boldsymbol{\theta}_t)$ grows approximately quadratically with t. Despite some limitations of Taylor expansion, this local quadratic growth provides conceptual understanding for the empirical rise observed in Figure 2. More rigorous theoretical analysis can be explored in future work.

3.3. Distance-driven detoxification

Motivation and objective. Motivated by the above observations, our goal is to design a strategy that enables fine-tuning to escape the *"backdoor region"* around the initial backdoored weights. A straightforward approach is to identify model weights that are significantly distant from the initial backdoored weights, while still maintaining performance on clean samples. This dual objective can be formalized as a constrained optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & d(\boldsymbol{\theta}, \boldsymbol{\theta}_{init}) \\ \text{s.t.} \quad & \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_{cl}}\left[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)\right] \leq \epsilon. \end{aligned} \quad (3)$$

Here, d denotes the distance metric between the weights $\boldsymbol{\theta}$ and the initial backdoored weights $\boldsymbol{\theta}_{init}$. The constant $\epsilon \geq 0$ serves as a threshold for the model's performance on clean data. A smaller ϵ imposes a stricter requirement on the model's accuracy for clean samples, whereas a larger ϵ allows more flexibility in accuracy to achieve a greater distance from the initial backdoored weights.

Practical challenge and design considerations. While Problem 3 appears to have a simple formulation, *applying it effectively for backdoor defense is highly non-trivial.* Unlike vanilla fine-tuning, which typically maintains proximity to the pre-trained model to preserve generalization, Problem 3 explicitly seeks to move the model away from the backdoored weights while retaining clean accuracy. This introduces a set of intricate challenges that demand careful algorithmic consideration:

- *Challenge 1: Over-fitting under limited data.* In post-training backdoor defense, we typically have access to only a limited reserved dataset. Unlike vanilla fine-tuning, which converges to weights close to the initial values and thereby retains the pre-trained model's generalization capabilities, our approach aims to find a solution that significantly diverges from the pre-trained model. However, this

divergence can weaken the model's generalization, causing it to over-fit the reserved dataset. To mitigate this, we introduce a selective weight distance constraint, restricting divergence measurement to a subset of parameters $\boldsymbol{\theta}_s$. This strategy helps to prevent over-fitting to the reserved dataset, thereby preserving the model's generalization potential on broader, unseen data.

- *Challenge 2: Meaningful weight distance.* A naive implementation of distance maximization could be exploited by trivial weight scaling. For example, using L_1 or L_2 distance on the last linear layer alone might allow the model to increase weight magnitudes by simple scaling while not significantly affecting predictions, since scaling often does not change the classification decision (due to the $\arg\max$ operation). To address this, we introduce an additional constraint, $\boldsymbol{\theta}_s \in \mathcal{S}$ to regulate the scale of weights, ensuring that the measured distance reflects meaningful weight updates in the model's decision-making process rather than superficial transformations that leave little impact on the predictions. This constraint can be enforced through a projection operator $\mathcal{P}$ that projects the weights to $\mathcal{S}$

- *Challenge 3: Computational complexity of the constraint.* A key difficulty in Problem 3 lies in enforcing the clean loss constraint. Evaluating this constraint requires computing the loss over a dataset using a deep neural network, which is inherently non-convex and computationally expensive. A direct optimization under such constraints is impractical for large-scale models. To make the problem more tractable, we reformulate the constraint into a penalty term within the objective function, enabling a more efficient optimization process. While this approach balances feasibility and effectiveness, we acknowledge that alternative formulations might exist, and exploring them could be an interesting direction for future work.

Overall process. Based on the aforementioned analysis, we select a subset of the model's weights, denoted by $\boldsymbol{\theta}_s$, and define a space $\mathcal{S}$ to constrain these selected weights. In practice, $\boldsymbol{\theta}_s$ is usually chosen as weights of one or a few layer and $\mathcal{S}$ is designed to maintain the norm of the selected weights according to specific metrics. We then propose Distance-driven Detoxification (D3), which fine-tunes the model by solving the following optimization problem:

$$\min_{\boldsymbol{\theta}:\boldsymbol{\theta}_s \in \mathcal{S}} \quad -d(\boldsymbol{\theta}_s, \boldsymbol{\theta}_{init,s}) + \lambda \times \max\left(0, \mathcal{L}_{cl}(\boldsymbol{\theta}) - \epsilon\right),$$
$$(4)$$

where $\mathcal{L}_{cl}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y) \in \mathcal{D}_{cl}} \left[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)\right]$ represents the loss on clean samples, and the term $\max\left(0, \mathcal{L}_{cl}(\boldsymbol{\theta}) - \epsilon\right)$ acts as a penalty for violating the performance constraint on clean samples: if the clean loss exceeds ϵ, the penalty increases linearly with the excess loss. The hyper-parameter $\lambda \geq 0$ controls the trade-off between moving away from the backdoor weights and maintaining performance on clean data.

To solve this optimization problem, we employ a Projected Gradient Descent (PGD) scheme. Specifically, in each iteration, D3 performs an unconstrained gradient descent step with respect to the minimization objective in Equation 4, followed by a projection operation to ensure that $\boldsymbol{\theta}_s \in \mathcal{S}$. The pseudo-code of the algorithm can be found in **Supplementary Material**.

4. Experiment

4.1. Experiment setting

Backdoor attack. To thoroughly assess the effectiveness of our proposed method, we conduct experiments involving seven prominent backdoor attacks: BadNets Attack [8], Blended Attack [4], WaNet Attack [30], Low Frequency (LF) Attack [47], Input-aware Attack [29], Sinusoidal Signal (SIG) Attack [2] and Sample-Specific Backdoor Attack (SSBA) [19]. To ensure a fair and consistent evaluation, we use the checkpoints and configurations provided by BackdoorBench [42, 44], a standardized platform designed for backdoor learning. Specifically, the checkpoints are directly downloaded from the official website [1] if accessible and generated by the default configuration otherwise. By default, all attacks were executed with a 10% poisoning rate, targeting the 0th class unless otherwise specified. The effectiveness of these backdoor attacks was evaluated on three benchmark datasets: CIFAR-10 [16], Tiny ImageNet [18], and GTSRB [35]. We used three different neural network architectures to ensure a comprehensive assessment: PreAct-ResNet18 [10], VGG19-BN [34], and ViT-B-16 [6].

Backdoor defense. We evaluate our approach against popular and advanced backdoor defense methods, including vanilla fine-tuning, ANP [45], Fine-pruning (FP) [24], NC [38], NAD [21], i-BAU [48] and FT-SAM [51] and SAU [39]. To ensure a fair comparison, we use the configurations recommended by BackdoorBench [42, 44]. By default, the reserved dataset size is set to 5% of the training dataset unless otherwise specified. For the proposed method, we set $\boldsymbol{\theta}_s$ to be weights of the linear layers, as they is shared across all model architectures. In this section, we set $\lambda = 10$ and $\epsilon = 0.1$. Moreover, we measure the weight distance by the Frobenius norm of $\boldsymbol{\theta}_s - \boldsymbol{\theta}_{init,s}$ and constrain the Frobenius norm of $\boldsymbol{\theta}_s$. Due to space limitations, more details about the experiment setups can be found in Appendix 6 and more experiment results, including results for different configuration of the proposed methods, are presented in Appendix 8.

Metrics. To evaluate the effectiveness of each defense method, we employ three primary metrics: Accuracy on Clean Data (**ACC**), Attack Success Rate (**ASR**), and Defense Effectiveness Rating (**DER**). The **ACC** measures the

---

Table 1. Results (%) on CIFAR-10 with PreAct-ResNet18 and poisoning ratio 10.0%.

Defense →	No Defense			FT			ANP [45]			FP [24]			NC [38]		
Attack ↓	ACC	ASR	DER	ACC	ASR	DER	ACC	ASR	DER	ACC	ASR	DER	ACC	ASR	DER
BadNets [8]	91.32	95.03	N/A	89.96	1.48	96.1	90.88	4.88	94.86	91.31	57.13	68.95	89.05	1.27	95.75
Blended [4]	93.47	99.92	N/A	92.78	96.11	51.56	92.97	84.88	57.27	93.17	99.26	50.18	93.47	99.92	50.00
WaNet [30]	91.25	89.73	N/A	93.48	17.1	86.32	91.33	2.22	93.76	91.46	1.09	94.32	91.80	7.53	91.10
LF [47]	93.19	99.28	N/A	92.37	78.44	60.01	92.64	39.99	79.37	92.90	98.97	50.01	91.62	1.41	98.15
Input-aware [29]	90.67	98.26	N/A	93.12	1.72	98.27	91.04	1.32	98.47	91.74	0.04	99.11	92.61	0.76	98.75
SIG [2]	84.48	98.27	N/A	90.80	2.37	97.95	83.36	36.42	80.36	89.10	26.20	86.03	84.48	98.27	50.00
SSBA [19]	92.88	97.86	N/A	92.14	74.79	61.16	92.62	60.17	68.71	92.54	83.50	57.01	90.99	0.58	97.69
Average	91.04	96.91	N/A	92.09	38.86	78.77	90.69	32.84	81.83	91.75	52.31	72.23	90.57	29.96	83.06

Defense →	NAD [21]			i-BAU [48]			FT-SAM [51]			SAU [39]			D3 (Ours)		
Attack ↓	ACC	ASR	DER	ACC	ASR	DER	ACC	ASR	DER	ACC	ASR	DER	ACC	ASR	DER
BadNets [8]	89.87	2.14	95.72	89.15	1.21	95.83	91.49	2.28	96.38	88.56	1.33	95.47	90.77	0.74	96.87
Blended [4]	92.17	97.69	50.47	87.0	50.53	71.46	92.67	11.61	93.76	90.24	1.57	97.56	92.29	0.22	99.26
WaNet [30]	93.17	22.98	83.38	89.49	5.21	91.38	93.66	1.31	94.21	90.32	0.58	94.11	93.31	0.04	94.84
LF [47]	92.37	47.83	75.31	84.36	44.96	72.75	92.68	6.89	95.94	90.5	0.71	97.94	92.37	1.31	98.57
Input-aware [29]	93.18	1.68	98.29	89.17	27.08	84.84	93.50	1.54	98.36	91.08	0.93	98.66	92.96	0.06	99.10
SIG [2]	90.02	10.66	93.81	85.67	3.68	97.29	91.13	0.57	98.85	88.57	1.84	98.21	89.99	0.00	99.13
SSBA [19]	91.91	77.4	59.74	87.67	3.97	94.34	92.12	3.20	96.95	90.11	0.31	97.39	91.85	0.81	98.01
Average	91.81	37.2	79.53	87.5	19.52	86.84	92.46	3.91	96.35	89.91	1.04	97.05	91.93	0.46	97.97

model's ability to accurately predict clean samples, whereas the **ASR** indicates the percentage of poisoned samples misclassified to the attacker's desired target label. Higher **ACC** and lower **ASR** values indicate effective backdoor mitigation. The **DER**, which has been adopted in [39, 51], is a metric that ranges from 0 to 1 and is designed to measure the tradeoff between the preservation of **ACC** and the reduction of **ASR**. It is calculated using the formula:

$$\text{DER} = \frac{\max(0, \Delta \text{ASR}) - \max(0, \Delta \text{ACC}) + 1}{2}, \quad (5)$$

where ΔASR and ΔACC denote the reductions in **ASR** and **ACC** when comparing the defended model to the initial one.

4.2. Main results

Effectiveness of D3. To validate the effectiveness of D3, we first present the experimental results[2] on the CIFAR-10 dataset in Table 1. Due to space constraints, comprehensive results for other datasets and model architectures are provided in **Supplementary Material** 8. The results clearly demonstrate that D3 effectively mitigates backdoor attacks across various scenarios. Specifically, D3 achieves a notably lower average ASR compared to other methods, indicating its robustness against backdoor attacks. On the CIFAR-10 dataset, D3 attains the second-lowest ASR in six out of seven different types of attacks, with an ASR lower than 1% for the remaining attack. This consistent performance highlights

D3's ability to defend against a wide range of backdoor threats. *Similar exceptional performance is observed on the Tiny ImageNet and GTSRB datasets* (see **Supplementary Material** 8), further emphasizing its effectiveness in reducing the impact of backdoor attacks.

While mitigating backdoor attacks is essential, maintaining high clean accuracy is equally important. Therefore, we also evaluated D3's performance in terms of ACC and DER. It is worth noting that D3 involves pushing the model away from its initial weights to reduce susceptibility to backdoor attacks. This strategy, however, can have a slight negative impact on clean accuracy. Despite this, D3 still manages to achieve a slightly lower clean accuracy compared to the best-performing baseline. Notably, D3 achieves the best DER in most cases, which measures the balance between ACC and ASR. This indicates that D3 not only effectively reduces the risk of backdoor attacks but also maintains a balanced performance in terms of overall model accuracy.

Table 2. Results (%) under different poisoning ratio on CIFAR-10 and PreAct-ResNet18.

Poisoning ratio →	1.00%		10%		20%	
Attack ↓	ACC	ASR	ACC	ASR	ACC	ASR
BadNets [8]	92.18	0.68	90.77	0.74	89.94	0.77
Blended [4]	92.85	0.24	92.99	0.22	91.95	0.02
LF [47]	92.32	1.28	92.37	1.31	91.38	0.13

Poisoning ratio →	30%		40%		50%	
Attack ↓	ACC	ASR	ACC	ASR	ACC	ASR
BadNets [8]	88.97	0.99	88.45	1.02	86.90	1.51
Blended [4]	90.80	0.09	89.84	0.03	89.01	0.03
LF [47]	90.67	0.08	90.02	0.61	88.91	2.10

---

[2]**Note**: The most effective defense methods exhibit the highest **ACC**, lowest **ASR**, and highest **DER**. In the subsequent experimental results, the top-performing and second-best methods are highlighted in **boldface** and underline, respectively.

Influence of poisoning ratio. To thoroughly investigate the impact of the poisoning ratio on the effectiveness of the D3, we conduct a series of experiments where the poisoning ratio varies from 1% to 50%. This range is chosen to cover a spectrum from minimal to significant poisoning scenarios, allowing us to understand how D3 performs under different levels of adversarial attack intensity. Our findings, as shown in Table 2, reveal that D3 maintains a high level of robustness and effectiveness even when confronted with substantial data poisoning (up to poisoning ratio 50%).

Influence of reserved dataset. We examine the influence of the reserved dataset from two angles: the size of the reserved dataset and its source. Regarding the size of the reserved dataset, we assess the performance of the proposed method under varying reserved dataset size, ranging from 1% to 10% of the training dataset. The results are presented in Table 3, from which we can find that D3 remains effective in defending against backdoor attacks even with a small reserved dataset size. However, the ACC declines as the size of the reserved dataset diminishes, a challenge we previously discussed in Section 3.3.

Table 3. Results (%) on D3 with different reserved dataset size.

Attack →	BadNets [8]		Blended [4]		WaNet [30]	
Reserved size ↓	ACC	ASR	ACC	ASR	ACC	ASR
1.0%	88.57	2.31	90.64	2.86	91.96	1.42
5.0%	90.77	0.74	92.29	0.22	93.31	0.04
10.0%	90.97	0.44	92.61	0.01	93.53	0.11

To facilitate the practical application of D3, one potential approach is to leverage advanced generative models [41]. To test our method in such a scenario, we conducted experiments using 2,500 samples drawn from the CIFAR-5m dataset [28], which consists of synthetic CIFAR-10-like images created using the Denoising Diffusion Probabilistic Model [11]. The outcomes of these experiments are summarized in Table 4, indicating that D3 continues to perform effectively when utilizing a generated dataset.

Table 4. Results(%) using generated dataset CIFAR-5m.

Attack →	BadNets [8]		Blended [4]		WaNet [30]	
Data ↓	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-5m	90.42	1.11	92.16	0.20	92.85	0.04

4.3. Understanding D3

In this section, we delve deeper into the mechanism of D3. Firstly, we present the T-SNE visualization (Figure 3 **(a)**) of the model after applying D3, which demonstrates that our proposed method can effectively alleviate the backdoor

effect, thereby allowing poisoned samples to revert to their original clusters. Additionally, we examine the model from the perspective of weight distances, offering a histogram visualization (Figure 3 **(b)**) that highlights the differences in weights. This visualization reveals that the weight difference between the model after D3 and the initial model is notably greater than that between the vanilla fine-tuned model and the initial model. This significant increase in weight difference shows that D3 can effectively find a more distant solution, thereby mitigating the backdoor effect.
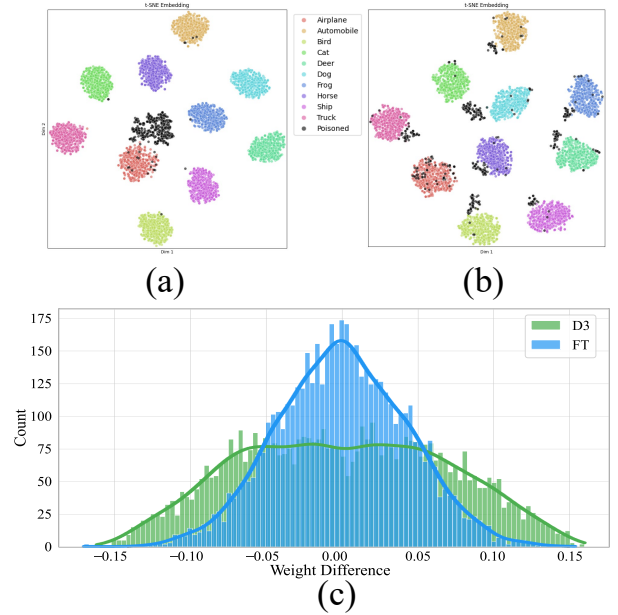

(a)   (b)

(c)

Figure 3. Figures **(a)** and **(b)** depict the t-SNE visualization of the model before and after applying D3 defense against BadNets attacks, respectively. In Figure **(c)**, the weight differences for each parameter in the selected layer are highlighted. The blue sections indicate the discrepancies between the vanilla fine-tuned model and the backdoored model, whereas the green parts illustrate the differences between the D3-processed model and the backdoored model.

4.4. Resistance to adaptive attack

When attackers are aware of the deployment of D3, they could devise adaptive strategies to bypass D3. Given that D3 reduces the impact of backdoors by shifting the model's weights away from their original values, an intuitive counter-strategy for attackers would be to reinforce the backdoor by refining the backdoored weights towards a flat minimum, thereby *making it more challenging for fine-tuning based methods to escaping the weight region with low backdoor loss.* To implement this adaptive attack, we adopt Sharpness-aware Minimization (SAM) techniques [17, 36], searching for a flat minimum for the backdoored model.

To assess the resilience of our method against adaptive attacks, we performed evaluations using backdoor attacks with a range of perturbation budgets for SAM, varying from 1.0 to 3.0. These budgets represent different levels of flatness in the loss landscape that attackers could potentially exploit. To highlight the advantage of D3 compared to other fine-tuning based methods, we compare D3 with six fine-tuning based baselines, including vanilla fine-tuning, FT-SAM [51], Super-FT [32], FST [27], NFT [14] and the proposed method D3. As shown in Table 5, as the flatness increases, the ASR of all baseline methods rises, confirming adaptive attacks exacerbate backdoor threats against fine-tuning defenses. Conversely, D3 effectively counters backdoor attacks (ASR< 3%) at all tested levels of flatness by actively steering the model away from backdoored weights.

Table 5. Defense results (%) against adaptive attacks on CIFAR-10 and PreAct-ResNet18.

Budget →		1.0		2.0		3.0	
Attack ↓	Defense ↓	ACC	ASR	ACC	ASR	ACC	ASR
	No Defense	90.42	96.43	90.71	96.28	91.56	95.6
	FT	92.02	26.3	92.64	33.01	92.55	71.24
	FT-SAM [51]	91.28	17.74	92.01	36.34	92.19	54.79
BadNets [8]	FST [27]	91.11	1.43	91.26	2.76	90.81	4.01
	Super-FT [32]	91.25	46.35	91.86	64.25	91.18	73.15
	NFT [14]	88.42	32.58	89.25	54.02	89.65	68.75
	D3 (**Ours**)	90.9	0.76	91.81	0.72	91.53	1.24
	No Defense	89.77	99.73	91.3	99.37	91.76	99.78
	FT	92.02	71.8	92.68	83.42	92.46	82.17
	FT-SAM [51]	92.17	72.71	92.14	82.8	92.39	91.93
Blended [4]	FST [27]	91.46	3.56	91.84	6.85	91.11	12.64
	Super-FT [32]	91.52	88.68	90.85	92.35	91.17	95.23
	NFT [14]	89.25	56.78	90.22	88.35	90.74	93.58
	D3 (**Ours**)	91.13	0.14	91.56	1.22	91.49	2.74

Table 6. Results (%) on D3 with different thresholds ϵ.

Attack →	BadNets [8]		Blended [4]		WaNet [30]		Average	
ϵ ↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
0	91.06	1.13	92.42	1.29	93.40	0.10	92.29	0.84
0.1	90.77	0.74	92.29	0.22	93.31	0.04	92.12	0.33
0.2	90.52	0.77	92.26	0.11	93.19	0.02	91.99	0.30
0.3	90.14	0.63	92.37	0.07	93.17	0.01	91.89	0.24
0.4	90.32	0.53	92.34	0.07	93.13	0.00	91.93	0.20
0.5	90.31	0.48	92.25	0.06	93.07	0.01	91.88	0.18

4.5. Ablation Study

In this section, we perform an ablation study to examine the impact of two key parameters: ϵ and λ.

Investigation of threshold ϵ. Recall that ϵ represents the constraint on the loss for clean samples. A lower value of ϵ indicates a stricter requirement for clean sample accuracy. Specifically, if the cross-entropy loss is used, a loss lower than ϵ implies an accuracy higher than $\exp(-\epsilon)$. To understand the effect of ϵ, we evaluate D3 with ϵ values ranging

from 0 to 0.5. The results are summarized in Table 6. From the table, we observe that a lower ϵ leads to a higher ACC but also a higher ASR. Notably, even when ϵ is set to 0, D3 still achieves acceptable defense performance.

Investigation of multiplier λ. As λ is the multiplier for the penalty term, higher λ indicates a stronger emphasis on maintaining high clean sample accuracy. To explore the effect of λ, we evaluated D3 with λ values ranging from 1 to 40. The results are presented in Table 7, which show that a higher λ results in a higher ACC but also a higher ASR.

Table 7. Results (%) on D3 with different multipliers λ.

Attack →	BadNets [8]		Blended [4]		WaNet [30]		Average	
λ ↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
1	88.66	0.19	90.18	0.13	91.21	0.16	90.02	0.16
10	90.77	0.74	92.29	0.22	93.31	0.04	92.12	0.34
20	91.14	1.41	92.70	1.17	93.35	0.09	92.40	0.89
30	91.36	2.03	92.57	4.28	93.32	0.49	92.42	2.27
40	91.34	2.01	92.78	23.36	93.35	2.84	92.49	9.40

In summary, both ϵ and λ have significant impacts on the performance of D3. Lowering ϵ and increasing λ both enhance the ACC, but they also increase the ASR, showing a trade-off between defense effectiveness and model accuracy.

5. Conclusion

In this work, we delve into the limitations of vanilla fine-tuning for mitigating backdoor attacks in a post-training context. Our analysis shows that vanilla fine-tuning frequently becomes trapped in regions with low loss for both clean and poisoned samples, which hinders effective backdoor removal. To address this challenge, we propose Distance-Driven Detoxification, a novel method that formulates backdoor defense as a constrained optimization problem. D3 encourages the model to move away from its initial weight space, thereby diminishing the impact of backdoors. Comprehensive experimental results across SOTA various ackdoor attacks, multiple model architectures, and diverse datasets confirm that D3 not only matches but often outperforms existing SOTA post-training defense methods, rendering it a promising method for backdoor mitigation.

Limitations and future work. One important direction for future work, and a current challenge, is addressing the trade-off between Accuracy and Attack Success Rate. One promising solution to alleviate such issue involves identifying backdoor-related weights, which can facilitate more accurate distance measurements and help mitigate the trade-off between ACC and ASR. Additionally, extending the D3 framework to handle more complex and diverse attack scenarios is another valuable direction.

Acknowledgment

References

[1] Insaf Adjabi, Abdeldjalil Ouahabi, Amir Benzaoui, and Abdelmalik Taleb-Ahmed. Past, present, and future of face recognition: A review. *Electronics*, page 1188, 2020. 1

[2] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *International Conference on Image Processing*, 2019. 2, 5, 6

[3] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety*, 2019. 2, 3

[4] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv e-prints*, pages arXiv–1712, 2017. 2, 5, 6, 7, 8

[5] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 5

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 3

[8] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, pages 47230–47244, 2019. 2, 3, 5, 6, 7, 8

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016. 5

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 7

[12] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *International Conference on Learning Representations*, 2022. 2

[13] Tran Ngoc Huynh, Anh Tuan Tran, Khoa D Doan, and Tung Pham. Forget-me-not: Making backdoor hard to be forgotten in fine-tuning. 2024. 1

[14] Nazmul Karim, Abdullah Al Arafat, Umar Khalid, Zhishan Guo, and Nazanin Rahnavard. Augmented neural fine-tuning for efficient backdoor purification. In *European Conference on Computer Vision*, pages 401–418. Springer, 2024. 8

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv e-prints*, 2013. 3

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

[17] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, 2021. 7

[18] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 5

[19] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *International Conference on Computer Vision*, 2021. 2, 3, 5, 6

[20] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *Conference on Neural Information Processing Systems*, 2021. 2

[21] 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. 3, 5, 6

[22] Daniel Liberzon. Calculus of variations and optimal control theory: A concise introduction. https://liberzon.csl.illinois.edu/teaching/cvoc/node8.html, 2010. 4

[23] Weilin Lin, Li Liu, Shaokui Wei, Jianze Li, and Hui Xiong. Unveiling and mitigating backdoor vulnerabilities based on unlearning weight changes and backdoor activeness. *arXiv e-prints*, 2024. 2

[24] 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. 2, 3, 5, 6

[25] Liangkai Liu, Sidi Lu, Ren Zhong, Baofu Wu, Yongtao Yao, Qingyang Zhang, and Weisong Shi. Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet of Things Journal*, pages 6469–6486, 2020. 1

[26] Min Liu, Alberto Sangiovanni-Vincentelli, and Xiangyu Yue. Beating backdoor attack at its own game. In *International Conference on Computer Vision*, 2023. 2

[27] Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. In *Advances in Neural Information Processing Systems*, 2023. 3, 8

[28] Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. *arXiv preprint arXiv:2010.08127*, 2020. 7

[29] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Conference on Neural Information Processing Systems*, 2020. 3, 5, 6

[30] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021. 2, 3, 5, 6, 7, 8

[31] Xiangyu Qi, Tinghao Xie, Jiachen T Wang, Tong Wu, Saeed Mahloujifar, and Prateek Mittal. Towards a proactive {ML} approach for detecting backdoor poison samples. In *USENIX Security Symposium*, 2023. 2, 3

[32] Zeyang Sha, Xinlei He, Pascal Berrang, Mathias Humbert, and Yang Zhang. Fine-tuning is all you need to mitigate backdoor attacks. *arXiv preprint arXiv:2212.09067*, 2022. 1, 8

[33] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Conference on Neural Information Processing Systems*, 2018. 2

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

[35] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *International Joint Conference on Neural Networks*, 2011. 5

[36] Hao Sun, Li Shen, Qihuang Zhong, Liang Ding, Shixiang Chen, Jingwei Sun, Jing Li, Guangzhong Sun, and Dacheng Tao. Adasam: Boosting sharpness-aware minimization with adaptive learning rate and momentum for training deep neural networks. *arXiv e-prints*, 2023. 7

[37] J-Donald Tournier, Robert Smith, David Raffelt, Rami Tabbara, Thijs Dhollander, Maximilian Pietsch, Daan Christiaens, Ben Jeurissen, Chun-Hung Yeh, and Alan Connelly. Mrtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *Neuroimage*, page 116137, 2019. 1

[38] 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 *Symposium on Security and Privacy*, 2019. 1, 2, 3, 5, 6

[39] Shaokui Wei, Mingda Zhang, Hongyuan Zha, and Baoyuan Wu. Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples. In *Advances in Neural Information Processing Systems*, 2023. 1, 2, 3, 5, 6

[40] Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Mitigating backdoor attack by injecting proactive defensive backdoor. *arXiv preprint arXiv:2405.16112*, 2024. 2

[41] Shaokui Wei, Shanchao Yang, Jiayin Liu, and Hongyuan Zha. Revisiting the auxiliary data in backdoor purification. *arXiv preprint arXiv:2502.07231*, 2025. 7

[42] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1, 3, 5

[43] Baoyuan Wu, Shaokui Wei, Mingli Zhu, Meixi Zheng, Zihao Zhu, Mingda Zhang, Hongrui Chen, Danni Yuan, Li Liu, and Qingshan Liu. Defenses in adversarial machine learning: A survey. *arXiv e-prints*, 2023. 1, 3

[44] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, Mingli Zhu, Ruotong Wang, Li Liu, and Chao Shen. Backdoorbench: A comprehensive benchmark and analysis of backdoor learning. *arXiv e-prints*, 2024. 1, 3, 5

[45] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Conference on Neural Information Processing Systems*, 2021. 2, 5, 6

[46] Danni Yuan, Mingda Zhang, Shaokui Wei, Li Liu, and Baoyuan Wu. Activation gradient based poisoned sample detection against backdoor attacks. In *ICLR*, 2025. 2

[47] Yi Zeng, Won Park, Z. Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *International Conference on Computer Vision*, 2021. 2, 5, 6

[48] Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022. 1, 2, 3, 5, 6

[49] Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *European Conference on Computer Vision*, 2022. 2

[50] Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Pre-activation distributions expose backdoor neurons. In *Conference on Neural Information Processing Systems*, 2022. 2

[51] Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *International Conference on Computer Vision*, 2023. 1, 3, 5, 6, 8

[52] Mingli Zhu, Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Neural polarizer: A lightweight and effective backdoor defense via purifying poisoned features. In *Advances in Neural Information Processing Systems*, 2023. 1, 2, 3

[53] Zihao Zhu, Mingda Zhang, Shaokui Wei, Bingzhe Wu, and Baoyuan Wu. Vdc: Versatile data cleanser for detecting dirty samples via visual-linguistic inconsistency. In *International Conference on Learning Representations*, 2024. 2, 3